Experiment – 2
Simulation and Real-time Implementation of a Switch-mode DC Converter

IT IS PREFERED that students ANSWER THE QUESTION/S BEFORE DOING THE LAB BECAUSE THAT provides THE BACKGROUND information needed for THIS LAB. (10% of the grade of the lab)

Q1: What are the power electronic devices which can be used in switch Mode DC converter and Why choose one over another(HINT: frequency limits, voltage limits, current limits etc?
Q2: Refering to the experiment # 2, what is the function of the triangle wave form "Repeating Sequence" Explain?
Q3: Refer to the Textbook (Chapter 5)
Show one way to reverse the direction of the rotation of the Compound DC Motor.

2.1 Introduction

In the previous experiment, a demonstration highlighting various components of the electric drives laboratory was performed. Real-time simulation file (*.mdl) and a Control-desk layout file (*.lay) were provided.

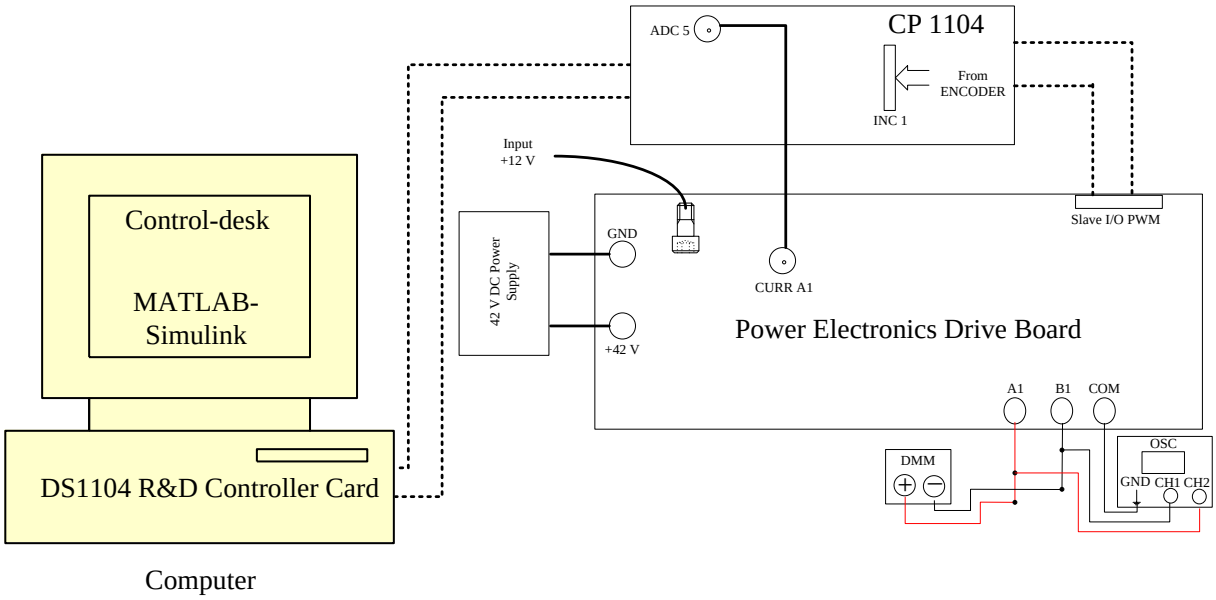

Figure 2.1: System Connections

In this experiment, a Simulink model (*.mdl) of a DC switch-mode power converter will be built. After verifying the simulation results with Simulink model, the model will be modified to control the output voltage of the converter in real-time. A control panel using dSPACE Control-desk will be designed (*.lay) that will serve as a user-interface to regulate the output voltage of the converter.

In section 2.2, theoretical background to implement DC switch-mode power converter in Simulink is briefed. Section 2.3 gives step-by-step instructions to simulate the converter in Simulink. In section 2.4, the Simulink model is modified for real-time implementation and step-by-step instructions to design the control panel using Control-desk are given.

2.2 Theoretical Background of DC Switch-mode converter

2.2.1 *Switching Power-Pole Building Block*

The switching power-pole building block has been explained in Section 1-6-1 of [1]. Depending on the position of the bi-positional switch, the output pole-voltage v_A is either V_{in} or 0. The output pole-voltage of the power-pole is a switching waveform whose value alternates between V_{in} and 0 depending on the pole switching function q_A. The average output voltage $\bar{v}_A$ of the power-pole can be controlled by controlling the pulse width of the pole switching function q_A.

$$\bar{v}_A = \frac{T_{up}}{T_s} V_{in} = d_A V_{in} \tag{1}$$

$$T_{up} \rightarrow \text{pulse-width of } q_A$$

$$T_s \rightarrow \text{switching time period}$$

2.2.2 *PWM of the Switching Power-Pole*

As seen in section 2.2.1, in order to control the average output voltage of the switching power-pole, the pulse width of the pole switching function q_A needs to be controlled. This is achieved using a technique called Pulse-Width Modulation (PWM). This technique is explained in section 12-2-1 of [1]. To obtain the switching function q_A, a control voltage $v_{cntrl,a}$ is compared with a triangular waveform v_{tri} of time period T_s. Switching signal $q_A = 1$ if $v_{cntrl,a} > v_{tri}$; 0 otherwise. As in [1],

$$v_{cntrl,a} = d_a \hat{V}_{tri} \tag{2}$$

Using equations (1) & (2) and assuming $\hat{V}_{tri} = 1V$,

$$v_{cntrl,a} = \frac{\bar{v}_{aN}}{V_d} \tag{3}$$

Where $\bar{v}_{aN} = \bar{v}_A$ = average pole-output voltage with respect to negative DC-bus voltage.

2.2.3 *Two-pole DC Converter*

The two-pole switch-mode DC converter utilizes two switching power-poles as described in the previous sections. The output voltage of the two-pole converter is the difference between the individual pole-voltages of the two switching power-poles. The average output voltage $\bar{v}_o = \bar{v}_{ab}$ can range from $-V_d$ to $+V_d$ depending on the individual average pole-voltages.

$$\bar{v}_o = \bar{v}_{ab} = \bar{v}_{aN} - \bar{v}_{bN} \tag{4}$$

To achieve both positive and negative values of $\bar{v}_o$, a common-mode voltage equal in magnitude to $V_d / 2$ is injected in the individual pole-voltages. The pole-voltages are then given by:

$$\bar{v}_{aN} = \frac{V_d}{2} + \frac{\bar{v}_o}{2} \tag{5}$$

$$\bar{v}_{bN} = \frac{V_d}{2} - \frac{\bar{v}_o}{2} \tag{6}$$

Solving equation (1) to (6),

$$d_a = v_{cntrl,a} = \frac{1}{2} + \frac{1}{2}\frac{\bar{v}_o}{V_d} \tag{7}$$

$$d_b = v_{cntrl,b} = \frac{1}{2} - \frac{1}{2}\frac{\bar{v}_o}{V_d} \tag{8}$$

The above equations will be implemented in Simulink.

2.3 Simulation of DC Switch-mode Converter

2.3.1 *Triangular waveform*

As explained in section 2.2.2, to modulate the pulse-width of the switching signal in a power converter, a control voltage has to be compared with a triangular waveform signal. This triangular waveform will be generated in Simulink, using the **Repeating Sequence block**.

- Create a new directory for the experiment (say *Expt02*).
- Start Matlab and set the path to this directory.
- Type "simulink" at the command prompt and create a new model from **File>New model**.
- Access the Simulink library by clicking **View > Library Browser.**
- In the Library Browser expand the **Simulink** tree and click on **Sources.** Drag and drop the **Repeating Sequence** block into your model.
- Simulink blocks usually have properties that can be modified by double-clicking on the blocks. Double click on the **Repeating Sequence** block and edit the properties as:
    - Time values: [0 0.5/fsw 1/fsw]
    - Output values: [0 1 0]
- Where "fsw" is the switching frequency (10 kHz) set as a global variable in the Matlab prompt. Type >>fsw = 10000
- Add a **Scope** to the model from **Simulink** $\rightarrow$ **Sinks.**
- Connect the output of **Repeating Sequence** block to the input of the **Scope.**
- The simulation model is now ready. However before running the simulation parameters need to

be changed. Go to **Simulation** menu and select **Configuration Parameters.** Set the parameters to the following values:

- Stop time : 0.002
- Fixed step size : 1e-6
- Solver Options: fixed step, ode1 (Euler)

- Run the simulation by clicking on the triangular button on the top. Double click on the scope after the simulation finishes. The result should look similar to the one shown in Fig. 2.2.

2.3.2 *Duty Ratio and Switching Function*

For a desired average output pole-voltage $\bar{v}_{aN}$, the control voltage $\bar{v}_{cnrl,a}$ is given by equation (3). Equation (3) is implemented in Simulink and the control voltage thus generated is compared with the triangular signal generated in the last part.

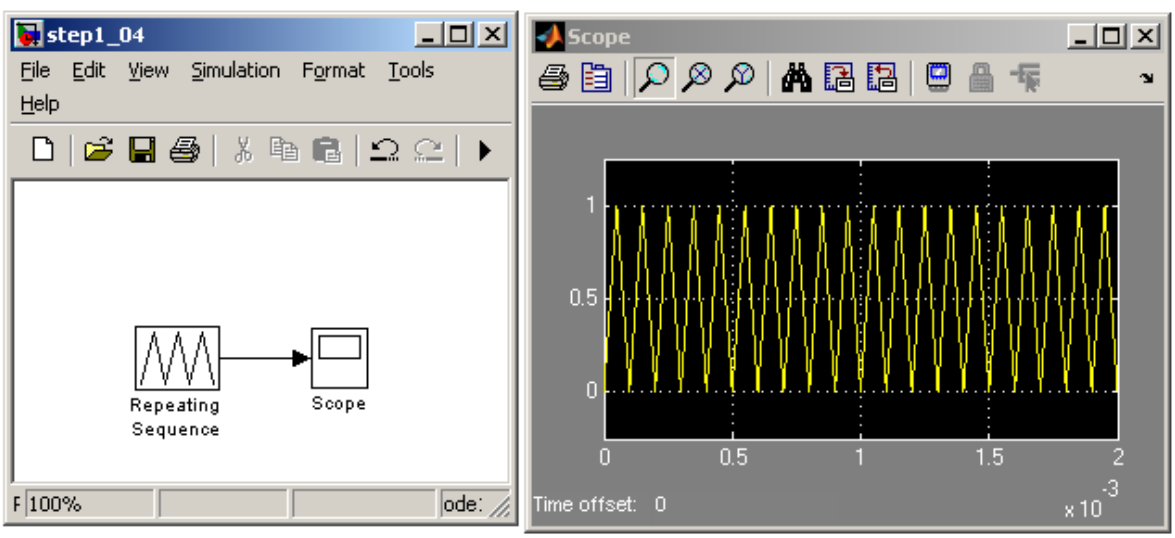

Figure 2.2: Triangular Waveform with 10 kHz frequency

The desired voltage $\bar{v}_{aN}$ is set by a **Constant** block with value one, and can be varied with a **Slider gain** from '0' to the maximum DC-bus voltage V_d (V_d = 42V in the model). The control voltage is generated by dividing $\bar{v}_{aN}$ by V_d. This is done by using a **Gain** block (of value $1/V_d$) at the output of the **Slider gain.**

Comparison of the triangular signal and the control voltage is done using a **Relay** block. The triangular signal is subtracted from the control voltage. The **Relay** block output is then set to '**1**', when the difference is positive and '**0**' when the difference is negative.

To create the model, follow the steps below:

- Open Simulink and create a new model.
- Copy and paste the model of triangular waveform generator from section 2.3.1 (Fig 2.2).
- Add these parts to the model
    - **Constant** block from **Simulink** $\rightarrow$ **Sources.**
    - **Slider Gain** from **Simulink** $\rightarrow$ **Math Operations.**
    - **Gain** from **Simulink** $\rightarrow$ **Math Operations.**
    - **Sum** from **Simulink** $\rightarrow$ **Math Operations.**
    - **Relay** from **Simulink** $\rightarrow$ **Discontinuities.**
- Now change the properties of these blocks as follows:
    - Change the **Slider Gain** limits as shown in Fig. 2.3.
    - Change the value of **Gain** to $1/V_d$ (V_d will be set to 42V from the command prompt later).
    - Change **Sum** block signs to $|-+$.

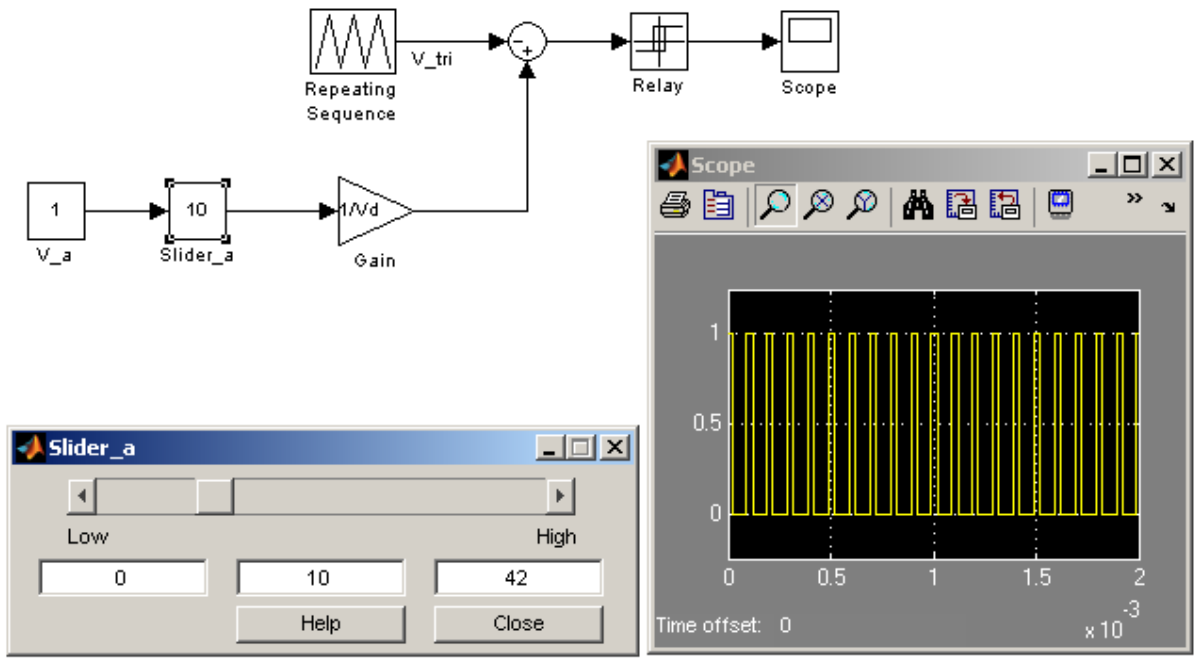

Figure 2.3: Switching Function generation for single pole converter

- Rename the blocks and connect them as shown in Fig. 2.3.
- In the Matlab prompt, type: fsw = 10000, Vd= 42.
- Set the simulation parameters as in section 2.3.1 and save the model.
- Run the simulation and save the waveform for the switching function. (Fig. 2.3)

2.3.3 *Two-pole Converter Model*

Equations (7) and (8) describe the control voltages of the two poles A & B depending on the desired output voltage $\overline{v}_o = \overline{v}_{ab}$. These equations will be implemented in Simulink (Fig. 2.4). Also, the switching power-poles will be modeled using a **Switch.** The **Relay** blocks provide the switching functions for the poles q_a and q_b. Depending on the value of the switching function, the **Switch** outputs the pole-voltage as follows:

For $q_a = 1$, switch output (Pole A) = $v_{aN} = V_d$

For $q_a = 0$, switch output (Pole A) = $v_{aN} = 0$

Create the Simulink model as shown in Fig. 2.5. The **Switch** block can be found in **Simulink**→ **Signal Routing.** Change the threshold voltage of the switch to 0.5 (Fig 2.4), or to any number greater than 0 but less than 1. Can you tell why?

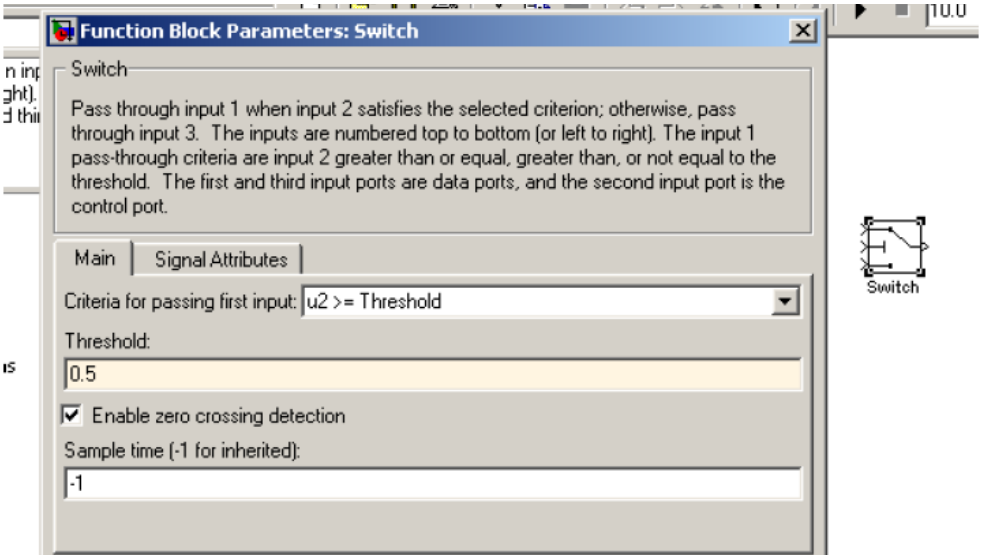

Figure 2.4: Settings for switch block

- Set the simulation parameters and values of fsw and Vd as in section 2.3.2. Run the simulation.

- Collect the following results:

  - Switching function q (t) for pole 'A' of the two-pole converter.

  - Simulation results of a two pole converter model for two different values of **V_ab**, one positive and one negative.

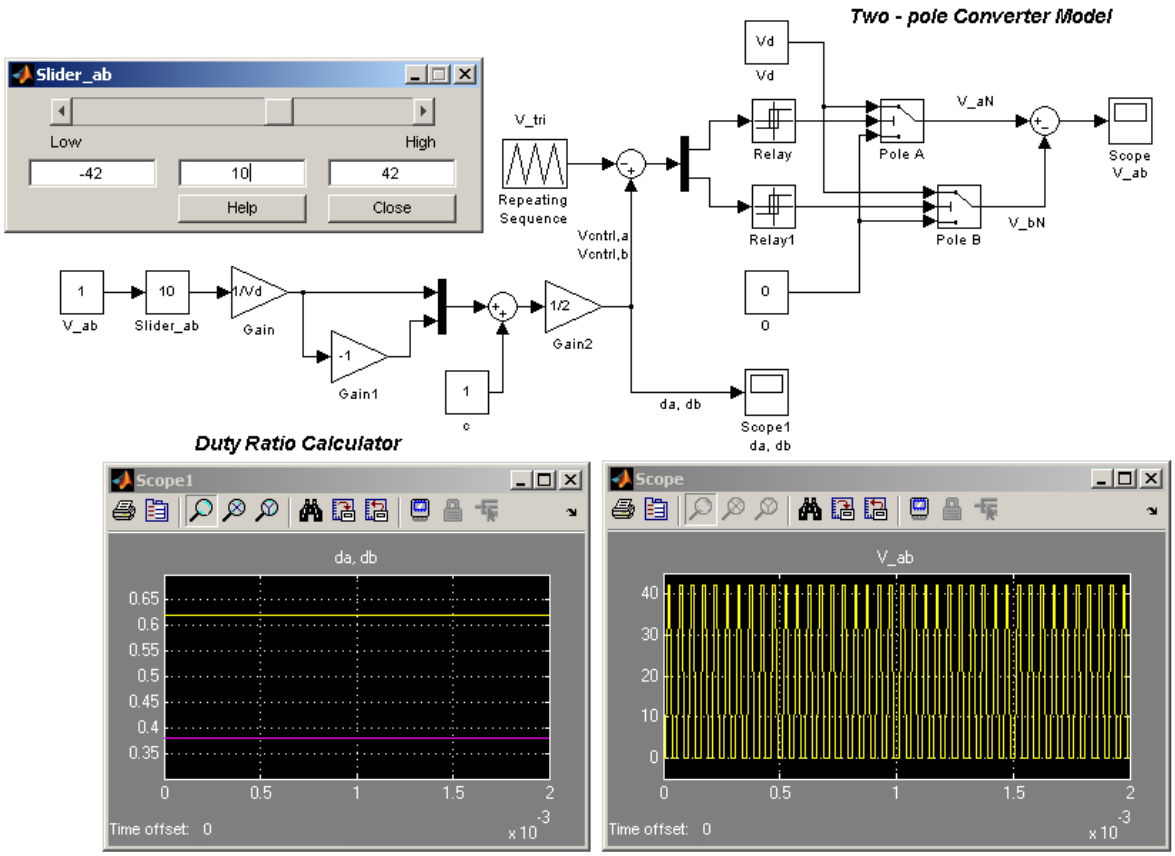

Figure 2.5: Two Pole Switch-Mode Converter Model in Simulink

2.4 Real-time Implementation of DC Switch-mode Converter

Having simulated the two-pole DC switch-mode converter, it will now be implemented in real-time on DS1104. This means that the converter will now be implemented in hardware and its output voltage amplitude will be controlled in real-time using an interface (made possible by the use of dSPACE Control-desk). As explained in experiment-1, real-time implementation involves exchange of signals between the dSPACE Control-desk interface, DS1104 and the Power-Electronics-Drives-Board. In this experiment, the output voltage reference will be set from the Control-desk interface. The duty ratios for the two poles will be calculated from this output voltage reference inside DS1104. PWM will be internally performed and the switching signals thus generated will be sent to the power electronics drives board through the CP1104 I/O interface. Make connections as shown in Fig. 2.1.

dSPACE provides a block called **DS1104SL_DSP_PWM3**, which embeds the triangular waveform generator and the comparator for all converter poles. The inputs for

DS1104SL_DSP_PWM3 are the duty-ratios for the poles. In Fig. 2.5, the lower part of the model is called the ***Duty Ratio Calculator.*** This part of the model will again be used in the real-time model to generate the pole duty ratios. The triangular wave generator and comparison using relays will be replaced by **DS1104SL_DSP_PWM3,** as these functions are internal to the block. Two legs of the drives board (refer appendix 'A') will replace the two poles (modeled using the **Switches** in Simulink).

- Create the real-time model as shown in Fig. 2.6. Use the ***Duty Ratio Calculator*** from section 2.3.3.

- For the **DS1104SL_DSP_PWM3** block, set the switching frequency as 10000 Hz and the dead-band to '0'. (**DS1104SL_DSP_PWM3** from dSPACE RTI1104 →Slave DSP)

- Make the following changes in the Configuration Parameters.
  - Simulation➔Configuration Parameters

    Change the stop-time to ***inf,*** fixed step size to 0.0001
  - Simulation➔Configuration Parameters➔Code Generation

    Set the 'system target file' to rti1104.tlc
  - Simulation> Configuration Parameters >Optimization>

    uncheck 'Block Reduction'

- Set Vd = 42V in the Matlab command prompt.

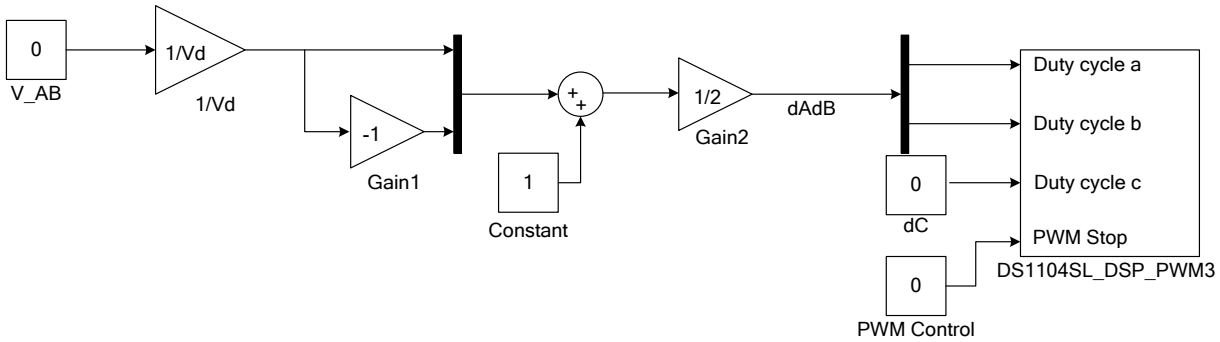

Figure 2.6: Control of a two-pole switch-mode converter in real-time

Once the real-time model is ready, it can be implemented on the DSP of DS1104 by building the model. As explained in experiment-1 building the model will broadly cause:

1. Compilation of C-code (generated by Simulink) and its hardware implementation on DS1104.

2. Generation of a variable file (with extension .sdf) that allows access to the variables and signals in the real-time Simulink model.

   - Build the Simulink model by pressing (CTRL+B). Observe the sequence of events in the Matlab command window.

   - Once the real-time model is successfully built, open Control Desk (icon on PC Desktop).

   - Using the File menu, create a **New Experiment** and save it in the same working root as the real-time Simulink model. Create a **New Layout** using the File menu again. Two new windows will appear in the Control Desk workspace. The one called **Layout1** will contain the instruments used for managing the experiment. The second window is a library, which will let us drag and drop the necessary controls for the experiment into the **Layout**. You can also open the existing **exp2.lay** file from Lab2_Summer2011 folder.

   - Now, select **File>Open Variable File.** Browse to the directory containing the real-time Simulink model. Open the .sdf file (e.g. For Simulink model named twopole.mdl, the variable file will be twopole.sdf).

   - After opening the variable file, notice that a new tab in the lower window called **Variable Manager** appears below the layout (Fig 2.7). The variables of the real-time Simulink model file are under the tree **Model Root.** Expand **Model Root,** observe the variables and relate them with the real-time Simulink model.

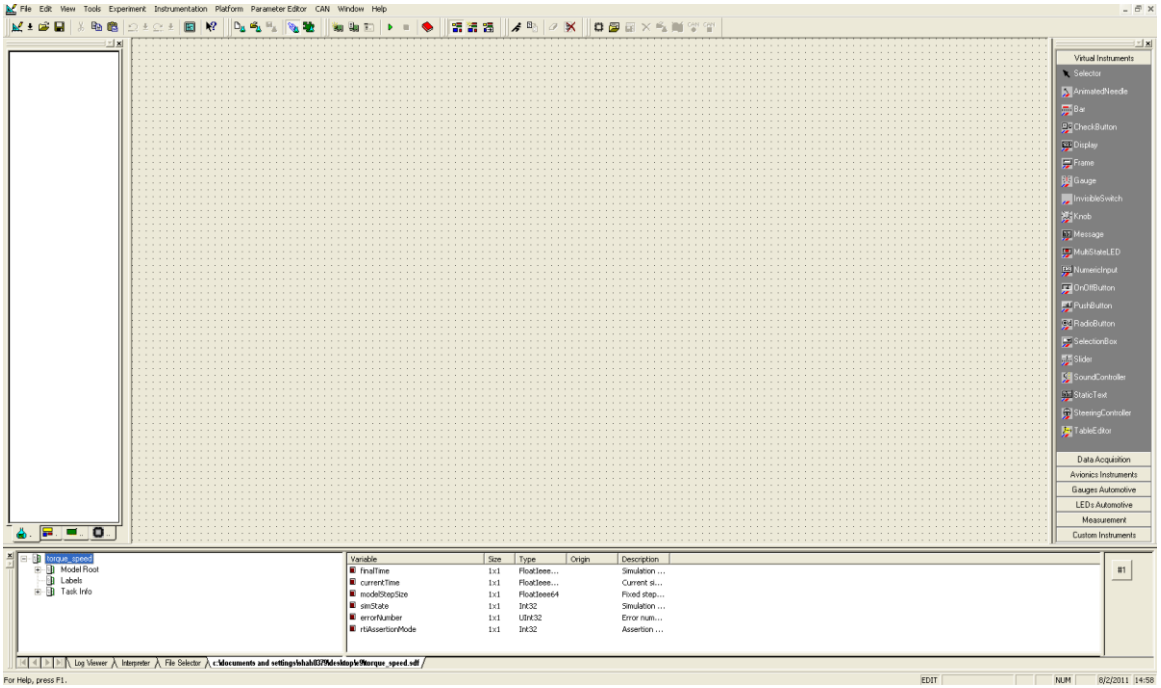

Figure 2.7: New Layout Window for Instrumentation and Control

- Now a user-interface that allows us to change input variables & system parameters (in real-time) and also observe signals will be created. The input variable in this experiment is the output voltage of the switch-mode converter. The duty ratios generated by the ***Duty Ratio Calculator*** will be the signals that will be observed in the layout. The actual pole voltages will be observed directly from the power electronics drives board using an oscilloscope.

- In order to change the reference output voltage and observe the duty ratios, suitable parts need to be added to the layout. These parts are available in the window to the right of the layout. The output voltage reference **V_AB** will be set using a **Slider** and a **Numerical Input.** Both these parts are found under **Virtual Instruments.** Click and draw these parts in the layout. The duty ratios will be observed in a **Plotter** available in **Data Acquisition.** Select **Plotter** and draw it in the layout.

- Now appropriate variables will be assigned to the parts. Under **Model Root,** locate **V_AB** and select it. It will have a parameter called **Value** (right side panel) which corresponds to the value of the **Constant** block **V_AB** in the real-time Simulink model. Drag and drop **V_AB/Value** into the **Slider** and also on **Numerical Input** one-by-one. Now, the value of

V_AB can be changed in real-time using these two parts. Similarly, to observe the duty ratios in real-time, assign the two outputs **(Out1** and **Out2)** of the **De-mux** (the one following the **Gain2** block) to the plotter. The experiment is now ready; it should look as shown in Fig 2.8. Start the experiment by clicking the Start button and select the animation mode (Fig 2.8)

- Turn the power supply ON and observe the pole-voltages on the oscilloscope. Vary the output voltage reference **(V_AB)** using the **Slider** or the **Numerical Input.** Observe the changing duty-ratios and pulse-widths of the pole-voltages.

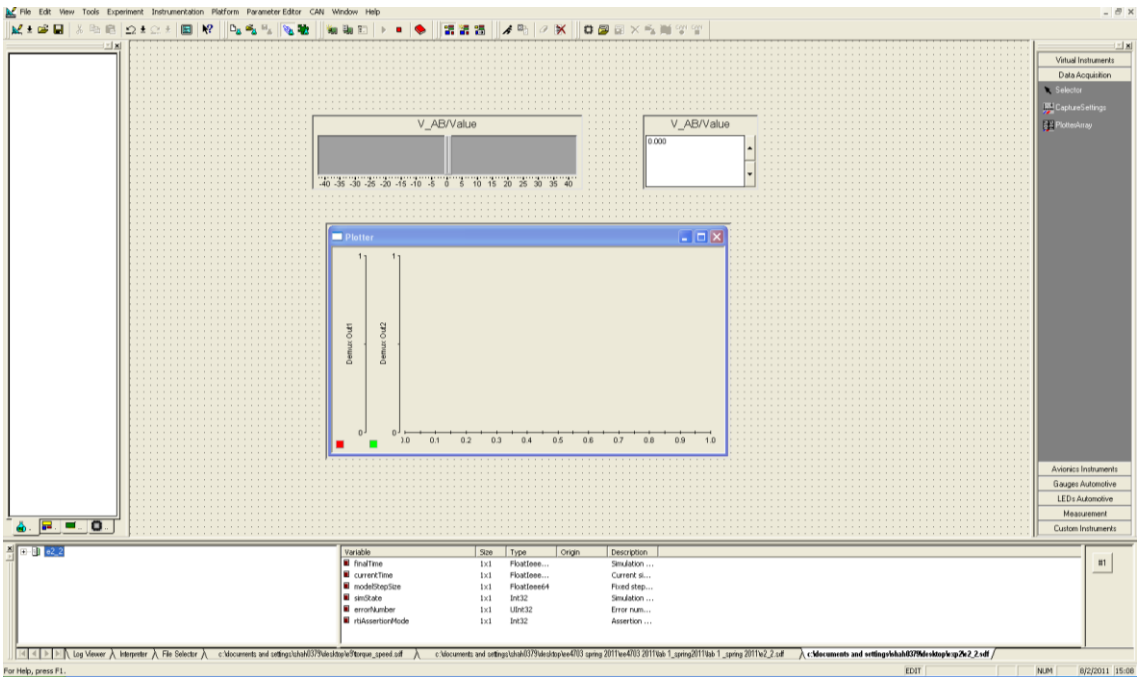

Figure 2.8: Control Desk layout for Switchmode DC Converter

2.5 Lab Report

- Include the following results in your report

  1) Section 2.3.2: Run the simulation and save the waveform for the switching function. (Fig. 2.3)
  2) Section 2.3.3:
     a) Duty ratios da and db for the two-pole converter.

     b) Simulation results of two-pole converter model for two different values of **V_ab**, one positive and one negative.

- Record the output voltage waveform on the oscilloscope for $\mathbf{V_{A1}}$ and $\mathbf{V_{B1}}$ w.r.t. COM (two probes will be used) and obtain by subtraction on the scope the values of $\mathbf{V_{A1B1}}$ set in section 2.4.

- Record the corresponding duty ratio waveforms for the above values.

- Measure the output voltage frequency and comment on the result obtained (**Hint**: relate the frequency set in the PWM block to the frequency of the voltage observed on the oscilloscope).