

# Lab 5

## Saving and Loading Data

### 5.1 Introduction

In this lab manual, we will learn how to save and load data in MATLAB and VisualDSP++ environments.

The saving of data to disk and loading of data from the disk becomes handy during a number of scenarios, e.g.

- When a lot of processes are going on and the output data that is generated has not enough space in the memory, so at some intermediate data can be stored and processed again
- When we require data at some other time
- When we want to export data from one machine to another
- When we want to use data in some other application on the same machine
- When we do not want to generate the same data again and again
- When we want the data generation and data manipulation sections separate

Keeping the above-mentioned scenarios in mind, it becomes necessary to get a grasp of saving and loading data.

#### 5.1.1 Saving and Loading Data in MATLAB

The MATLAB environment is very versatile in generating data and signals. The procedure to save and load the data is also very efficient and easy in MATLAB.

The commands to save an array or arrays are given by the keyword *save*. For example, to save a matrix *A* to a file called *filename* in ascii format we type the expression

*Save filename.dat A -ascii (5.1)*

---

To store all the workspace variable in a file *filename* in \*.mat format, we use the expression

$$\text{save}(\text{"filename"}) \quad (5.2)$$

To save some of the variable in MATLAB to a file *filename*, we use the format like:

$$\text{save}(\text{"filename"}, \text{var1}, \text{var2}) \quad (5.3)$$

The same procedure is applied when we save the data in ascii format.

$$\text{save}(\text{'filename'}, \text{var1}, \text{var2}, \text{'ASCII'}) \quad (5.4)$$

In order to load data from a file to the workspace, we use the command *load*. To load data from a file called *filename*, we use the command

$$\text{load}(\text{"filename"}) \quad (5.5)$$

If the extension of the file is other than \*.mat, MATLAB treats the file as an ascii file. To load specific variables from a \*.mat file only, use the following command.

$$\text{load}(\text{"filename"}, \text{var1}, \text{var2}) \quad (5.6)$$

The command

$$\text{val} = \text{load}(\dots) \quad (5.7)$$

loads the contents of the file in a variable *val*.

If the file is made in some other software and the data must be given to MATLAB for further processing, the commands *fopen()* is first used to open the file to MATLAB's environment. To read or write data to the file, we use the commands *fread()* and *fwrite()*. When the editing is done, we close the file using *fclose* function.

### 5.1.2 Saving and Loading Data in VisualDSP++

To load or save data from VisualDSP++ environment is the same as in C/C++. The commands like *fopen()*, *fclose()*, *fread()* and *fwrite()* are used here as well.

To open a file *filename* for reading in VisualDSP++, use the command *fopen*.

$$\text{fptr} = \text{fopen}(\text{"filename"}, \text{"rb"}); \quad (5.8)$$

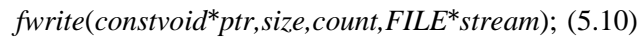
where *fptr* is a FILE\* type pointer. For writing data to a file, we replace *rb* by *wb*.

The command

$$\text{fread}(\text{void*ptr}, \text{size}, \text{count}, \text{FILE*stream}); \quad (5.9)$$

reads an array of *count* elements, each one with a size of *size* bytes, from the *stream* and stores them in the block of memory specified by *ptr*.

To Write an array of *count* elements, each one with a size of *size* bytes, from the block of memory pointed by *ptr* to the current position in the *stream*, we use the command



- 
4. Save sinusoids of frequency 20Hz and 30Hz in an \*.mat file. Retrieve the 20Hz sinusoidal only from the file.
  5. Generate an array of short in VisualDSP++ 4.5. Display the data in MATLAB.
  6. Import an integer array from MATLAB into VisualDSP++ 4.5 and perform its plotting.
  7. Import data from MATLAB into Simulink.

### **5.3 Hints**

Use the MATLAB's help during the procedure. The C function to generate random numbers is *rand()*.