Working with LabWindows CVI 7.0 and the Mascot Matrix Option

Jesus Ivan Gonzalez
REU Report – FTICR Group
July 30, 2004

Abstract

This report oversees the usage of LabWindows CVI 7.0 and how it corresponds in the Fourier Transform Ion Cyclotron Resonance Mass Spectrum (FT-ICR MS) group. It is directed to anyone who is in the interest of computer science, or who wants to explore how computer science is used in research facilities. This includes working with the C program language and working with the graphical user interface (GUI) and how it interacts with the Mascot Matrix website (http://icr-mascot.com). Also included with this report will be a light discussion of the physics and mathematics to give the general audience an introduction of how FT-ICR works.

Introduction – How Everything Fits Together

The purpose of FT-ICR provides the understanding of identifying complex compounds in a molecular level. It gives an insight of how molecules are constructed and how that information can be transformed into useful data that can help fight diseases, increase prevention, and improve medicine effectiveness.

As in any research project, the FT-ICR group follows a procedure to test their ion samples. This procedure includes inserting an ion sample into a series of vacuums to take away atomic pressure on the ion sample until the ion sample feels that is in a virtually zero pressure atmosphere while approaching a magnet. When the ion sample reaches the magnet, the presence of a magnetic field along with an electric field will force the ions to spin in a cyclotron pattern. Once the ion samples reach to a radius where it barely touches the outer electrodes, a current is produced. Then the current is connected to an RC circuit in which the mass-to-charge ratio, kinetic energy, frequency, magnetic strength, and other information can be found.

When the results are received, they are displayed in a sinusoidal format since the ions are in a constant cyclotron spin due to the presence of the magnetic and electric fields. FIGURE 1 shows an example of how results are gained in the real time domain. When converted into a Fourier Transform format, then results shown in FIGURE 2 are expected. In the Fourier Transform mode, the peaks represents a complex molecular structure within the ion sample.

FIGURE 1 – Results Attained From an Ion Sample in Sinusoidal Form

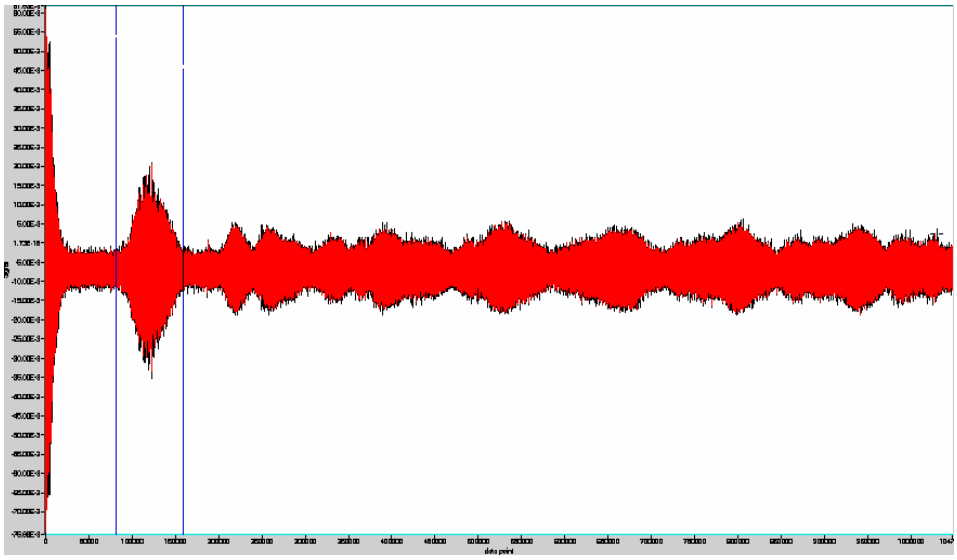
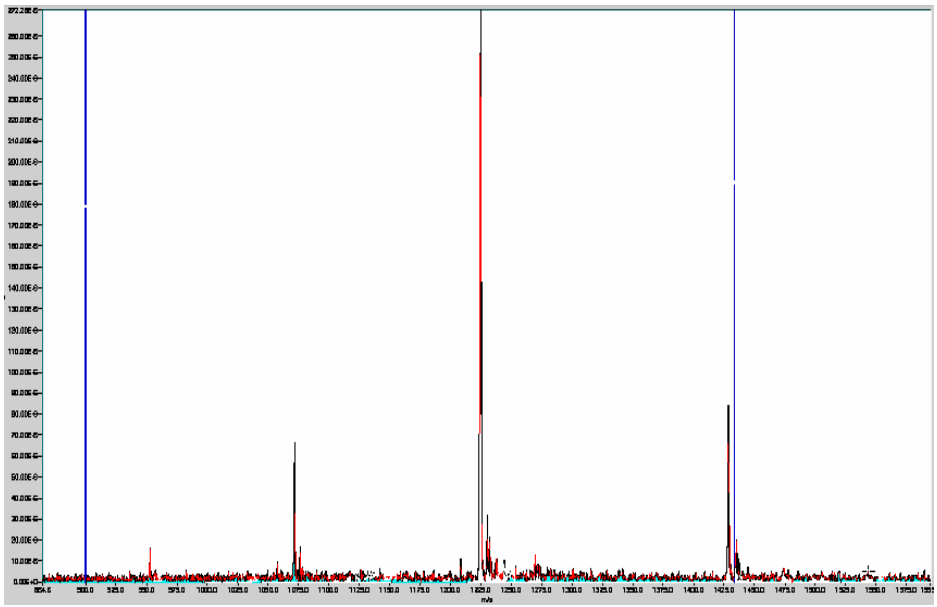


FIGURE 2 - The Fourier Transform of FIGURE 1



The computer science behind the FT-ICR group deals with how the information is handled and where the information can be processed. This is where the MASCOT option comes in. The MASCOT option will have the ability to create individual profiles, to be able to set the information provided to a specific setting to the user's liking, and be able to send the information to the MASCOT website in where possible results can be attained, if any.

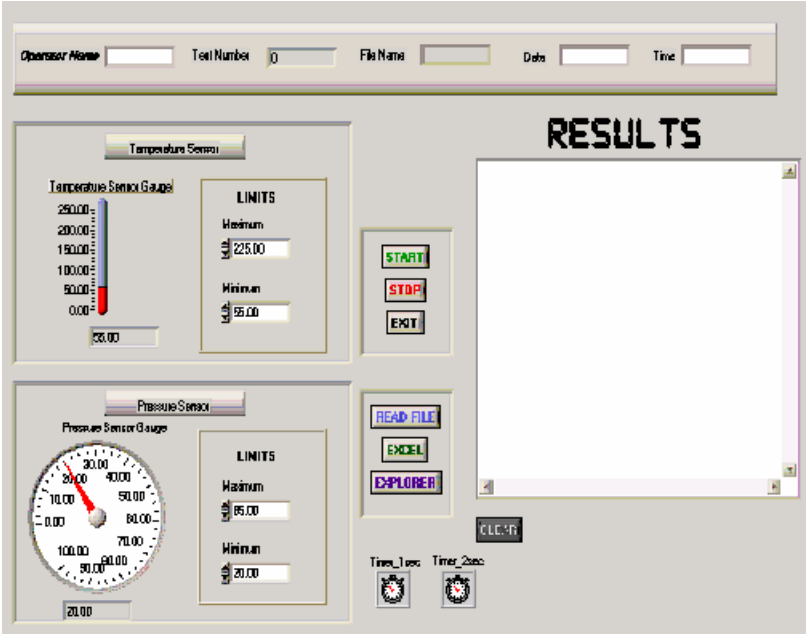
Learning LabWindows CVI 7.0

One if the first tasks to create the MASCOT option was to know the way how LabWindows CVI 7.0 works. One of the major benefits of CVI is that it gave the programmer the ability to deal with the Graphical User Interface (GUI). GUI allows the programmer to program with buttons, gauges, lists boxes, rings, etc. in which each item can be converted into skeleton code and be manipulated by the actions of the mouse pointer. This can be beneficial because the programmer has the ability to make a blank panel into a user friendly panel and decrease the chance of any programming errors to occur. Another benefit that CVI has is that the programming behind it is written in C language which is a common programming language used in the programming world. This was ideal for the MASCOT option.

To learn the basics of CVI, a series of sample GUI panels and C program coding were needed to be created. This hands-on experience was a valuable way to gain inside knowledge of how the rules work in the CVI world. One CVI example that stood

challenging was analyzing sample gauges of temperature and pressure and how their corresponding results are displayed. FIGURE 3 shows the GUI part of that example.

FIGURE 3 – Using CVI through GUI



Creating the GUI part of CVI was the easy part. The grunge of the work comes from the C programming. Like mentioned before, CVI uses skeleton code which is directed connected to the actions of the pointer of the mouse. This includes such items as the location of the pointer, changing the value on a gauge, and what mouse button occurred. An example of the C programming language from FIGURE 3 is displayed in FIGURE 4.

FIGURE 4 – A Small Portion of C Programming of FIGURE 3

```
int CVICALLBACK StartTestCB (int panel, int control, int event,
void *callbackData, int eventData1, int eventData2)
  switch (event)
    {
    case EVENT_COMMIT:
```

```
      TestRunNumber++;
      SetCtrlVal(TestHandle, TST_TEST_NUM, TestRunNumber);
      //Display the time on the panel
      SetCtrlVal(TestHandle, TST_TIME, TimeStr);
      response = ConfirmPopup("WRITE TO FILE", Do you want to write data to the file?");

      if (response) //
      WriteToFile();

      DefaultCtrl(TestHandle, TST_RESULTS);  //Clear Box
      Fmt(buf, "%s<Time   Temperature   Result    Pressure    Result");
      InsertTextBoxLine(TestHandle, TST_RESULTS, -1, buf);

      //Start the timers
      SetCtrlAttribute(TestHandle, TST_TIMER_1SEC, ATTR_ENABLED, 1);
      SetCtrlAttribute(TestHandle, TST_TIMER_2SEC, ATTR_ENABLED, 1);

      SetInputMode(TestHandle, TST_START, 0); //Disables the start button
      SetInputMode(TestHandle, TST_READFILE, 0);//Disables the read file button
      SetInputMode(TestHandle, TST_STOP, 1); //Activates the stop button

    case EVENT_RIGHT_CLICK:
      MessagePopUp("", "Starts the project running");
      break;
    }
  return 0;    } //StartTestCB
```

In FIGURE 4, the callback function on the top section of the C code determines the location and action of the pointer respected to the item being activated. This is the skeleton code in CVI that is rarely touched by the programmer. Below that is the switch feature in where it tells the computer what would happen if certain actions are taken place on that particular button, list box, etc. The first case statement tells the computer that if EVENT_COMMIT happens (which is the standard left-click on the mouse), then the following commands that is within the case statement will be activated. If another action

occurs, like EVENT_RIGHT_CLICK, then another set of actions occurs. So in FIGURE 4, when someone left-clicks on the START button, which its callback function is called StartTestCB, then CVI will realize the START button was activated by the use of the left-click button and will follow the exact instructions inside the case statement. If the START button was right-clicked, then CVI will notice that the START button was activated using the right-click button and will display a pop up message telling what the action of the START button.

Creating the MASCOT Option

Once gaining the basic familiarity of CVI, it would provide an easier task of creating the MASCOT option. Like mentioned before, the MASCOT option will provide users the ability to create profiles so information from the ion samples can be stored, be able to manipulate the required information given in the MASCOT option to the user's liking, and be able to send in the information to the MASCOT website for analysis. The first thing that seems the easiest task was to create the GUI part of the MASCOT option.

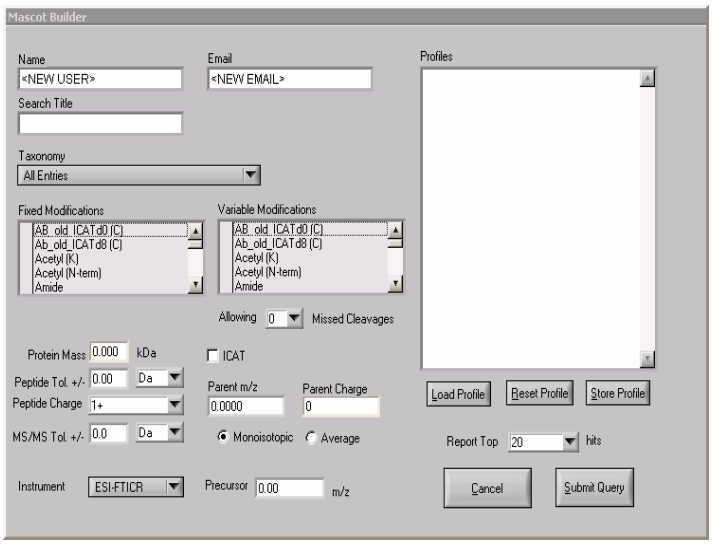
GUI Part of the MASCOT Option

The approach of creating the MASCOT option was similarly related to the temperature and pressure gauge example. This included creating a user-friendly panel while taking advantage of the skeleton code provided in CVI. One noticeable difference between the

temperature and pressure gauge example and the MASCOT option is that information needs to be transferred from the MASCOT option to the MASCOT website.

How the GUI was approached was to divide the information needed into three different areas. The first area of information that was needed was text boxes for the name of the user, the e-mail to receive the information from the MASCOT website, and the search title. The second area of information was the modifications of ion samples located in the lower left-handed section of FIGURE 5.

FIGURE 5 – The GUI Part of the MASCOT Option



This is where the user can determine what modifications they want to use in dealing with the ion sample such as the taxonomy, the variables used (if known), the protein mass, the peptide tolerance, and the instrument used. The final area of information consists of giving

the option to reset, load, & save to a given profile, displaying the name of the profile(s) in a list box, a SUBMIT QUERY button to submit information to the MASCOT website (or cancel), and the number of hits to receive in the e-mail if there are any matches within the MASCOT website.

C Programming Part of the MASCOT Option

Once the GUI part of MASCOT was constructed, telling the computer what to do was a difficult task to accomplish. So the best way to approach this difficult task was to attempt on working on the reset, load, and save options first. The written C code is displayed in FIGURE 6 and FIGURE 7.

FIGURE 6 – C Coding for the Reset Option

```
int CVICALLBACK Reset_Profile_CB (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)
{
  switch (event)
    {
    case EVENT_COMMIT:
      DefaultCtrl(mascot_handle, MASCOT_SEARCH_TITLE);  //Blanks out search title
      DefaultCtrl(mascot_handle, MASCOT_TAXONOMY);   //Reset back to "All Entries"
      DefaultCtrl(mascot_handle, MASCOT_FIXED_MOD);  //Reset back to "AB_old_ICATdO (C)"
      DefaultCtrl(mascot_handle, MASCOT_VARIABLE_MOD);//Reset to "AB_old_ICADdO (C)"
      DefaultCtrl(mascot_handle, MASCOT_MISSED_CLEAVAGES);  //Missed returned to 0
      DefaultCtrl(mascot_handle, MASCOT_PROTEIN_MASS);//reset to "0.000"
      DefaultCtrl(mascot_handle, MASCOT_ICAT);  //ICAT is unchecked
      DefaultCtrl(mascot_handle, MASCOT_PEPTIDE_TOL);  //peptide tolerance is 0
      DefaultCtrl(mascot_handle, MASCOT_PEPTIDE_TOL_SCALE); //reset to Da
      DefaultCtrl(mascot_handle, MASCOT_PEPTIDE_CHARGE); //Reset to 1+
      DefaultCtrl(mascot_handle, MASCOT_PMZ);   //Reset parent m/z to 0
```

```
      DefaultCtrl(mascot_handle, MASCOT_PCHARGE);//Reset parent charge to 0
      DefaultCtrl(mascot_handle, MASCOT_MS_MS_TOL);  //MS/MS tol is reset to 0
      DefaultCtrl(mascot_handle, MASCOT_MS_MS_TOL_SCALE); //Reset to Da
      DefaultCtrl(mascot_handle, MASCOT_MONOISO);
      DefaultCtrl(mascot_handle, MASCOT_AVERAGE); //Set monoio=1, average=0
      DefaultCtrl(mascot_handle, MASCOT_INSTRUMENT);   //Set instrument to default
      DefaultCtrl(mascot_handle, MASCOT_PRECURSOR);//Precursor to zero
      DefaultCtrl(mascot_handle, MASCOT_REPORT_TOP);  //Set to default
      break;}
  return 0;
}
```

FIGURE 7 – C Coding for the Save & Load Option

```
int CVICALLBACK Load_Profile_CB (int panel, int control, int event,
    void *callbackData, int eventData1, int eventData2)  //Information loaded from memory
{ char loadfile[260];
  int  status;
  switch (event)
    {
    case EVENT_COMMIT:
      status = FileSelectPopup ("", "*.*", "", "Load File for Profile",  VAL_LOAD_BUTTON, 0, 0, 1, 1, loadfile);
      break;}
  return 0;}

int CVICALLBACK Store_Profile_CB (int panel, int control, int event, //Information  stored into memory
    void *callbackData, int eventData1, int eventData2)
{ char    filename[260];
  int  status;

  switch (event)
    {
    case EVENT_COMMIT:
      status = FileSelectPopup ("", "*.*", "", "Select File for profile", VAL_SAVE_BUTTON, 0, 0, 1, 1,        filename);

      break;
    }
  return 0;
}
```

FIGURE 6 displays what to do when the reset option is activated by the left-click button from the mouse (recall that EVENT_COMMIT means when the left-click button is used). Once the button is activated, all of the information that controls the settings of the ion sample will be set back to their default value or position using the 'DefaultCtrl' function. The 'DefaultCtrl' requires the name of the panel used in the GUI part of the mascot and the constant name used to identify what button, ring, etc. needs to be reset back to their default value.

FIGURE 7 displays what to do if the load or save button is activated by a left-click mouse button. In these two options, an integer value called 'status' is needed. 'Status' will return a value in where the value returned will tell the computer if there was a problem with loading or saving a profile. For the both the load and save option, the function 'FileSelectPopup' is required. This will bring up a popup option that would require the title of the popup, a load/save button (depending on what option the programmer is working on), and if a cancel option will be needed. The remaining parts of the function ask the programmer if read-only, written-only, and hidden files will be included or not. Each response will accept either a '0' or a '1' where a '0' represents a 'NO' answer and a '1' represents a 'YES' answer.

Problems Encountered

With every project, problems are expected along the road. The problems occurred with the creation of the MASCOT option was the compiling errors which can be a time consuming problem for any programmer especially if it is intertwined with multiple programming files. Other common problems was not providing enough information to a called function or accidentally calling the wrong item in a function. This can be quite problematic if there is an uncertain knowledge about a function or if there are too many items on the GUI that are too similar in name.

What is Needed

What needs to be accomplished at this point is to create a test profile in where all of the options are tested in the MASCOT option. This includes the ability to create a new profile, resetting a profile's information, be able to store information into memory and to be able to retrieve the information. Once those areas are done, then the big task of sending the information to the MASCOT website needs to be approached. This includes sending in a sample profile to the MASCOT website and be able to attain a sample result through a given e-mail. Once that is done, then the MASCOT option is complete.

Conclusion

With the help of LabWindows CVI 7.0, the MASCOT option gives researchers the ability to store their ion sample results as well as to send their information to other resources to gain more insight of the molecular structure of their ion sample. By providing a depth of

how CVI works with the GUI and the C language, it gives readers an understanding of how the computer science behind FT-ICR group can play a vital role in medical advancement.

References

Marshall, Alan G., Hendrickson, Christopher L., and Jackson, George S. (1998). “Fourier Transform Ion Cyclotron Resonance Mass Spectrometry: A Primer”. John Wiley & Sons, Inc.

Khalid, Shahid F. (2000). LabWindows/CVI Programming for Beginners. Prentice Hall PTR.

"Mascot Matrix". http://icr-mascot.com