EIGENVECTOR RESEARCH INCORPORATED
Research, Training, and Software

3905 W. Eaglerock Dr.
Wenatchee, WA 98801
www.Eigenvector.com

Using PLS_Toolbox and MATLAB Online

Key words: ActiveX, Client/Server, Compiled m-files, Online, OPC

Introduction: For many chemometrics users, much of the value obtained through the use of these methods is not truly realized until their models are deployed in the field. Taking multivariate models "online" not only represents a valuable tool for process monitoring but is rapidly becoming an industry standard in many fields. Implementing an online model can allow increased utilization of process monitoring assets and extract value and efficiency from existing infrastructure. .

Depending on the application, such deployment demands “real-time”, or on-line execution of these models, to newly-generated data. A distinct advantage of PLS_Toolbox and MATLAB for such on-line applications is the multitude of different on-line model implementation options that are available This flexibility allows one to work efficiently with legacy systems, create customized, yet cost effective solutions from scratch, and enable more economical scale-up of laboratory applications to pilot/production applications. Some options for using PLS_Toolbox and MATLAB online include the following:

- PLS_Toolbox / MATLAB as a ***Server***
- PLS_Toolbox / MATLAB as a ***Client***
- Compiled Programs
- OPC

This paper gives a brief description of each of these options and presents their advantages and disadvantages .

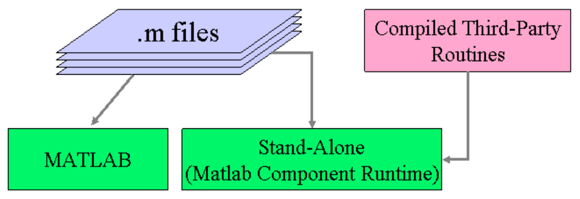

.m files
Compiled Third-Party Routines
MATLAB
Stand-Alone (Matlab Component Runtime)

Figure 1: Schematic of various on-line implementation options of PLS_Toolbox and MATLAB. *m-code* can be called from MATLAB or a Stand-Alone Runtime server of MATLAB.

PLS_Toolbox/MATLAB as a *Server*: In this scenario, MATLAB (with PLS_Toolbox installed) is running in the background on a system, ready to respond to requests made by client programs. In one example, a Distributed Control System (or, “DCS”, the “Client”) needs to apply a process model to newly-acquired data. It issues a specific “model execution” request to MATLAB, along with the new data, and the model to be applied (if it was not previously loaded). Upon receiving this request from the client, MATLAB “serves” the client by applying the model, using a set of data processing instructions encoded in MATLAB *m-code*, and returning a set of results back to the DCS.

This model implementation option takes advantage of the flexibility and versatility that PLS_Toolbox and MATLAB provide. Because MATLAB uses a high level programming language (commonly called *m-code*), one is able to rapidly prototype, modify and troubleshoot new on-line model application algorithms. There are also a wide range of “m-code” algorithm resources available for the MATLAB environment, including Toolboxes provided by The Mathworks, freeware from the MATLAB community, and other third party software.

Disadvantages of this option include a relatively steep learning curve for those new to MATLAB, costs associated with MATLAB and any additional required products, and establishing communications with data-generating systems that don't have common interfacing technology supported by MATLAB (this can also be an issue with proprietary systems).

PLS_Toolbox/MATLAB as a *Client*: when acting as a *Client*, a PLS_Toolbox/MATLAB instance is the “master” of the application. PLS_Toolbox/MATLAB initiates a request to one or more servers (other data sources) to provide data needed for a specific model application. The server returns the requested data to the client and the client then applies the model to the data and returns and/or displays the results.

This option has been made more appealing in recent years through enhanced data visualization and GUI capabilities of PLS_Toolbox and MATLAB. One example of this option might be MATLAB acting as a simple DCS, asking various instruments for data, processing it, and displaying results to an operator.

The advantages and disadvantages of this option are similar to those of the previous scenario. One additional consideration is scalability of MATLAB as a DCS in the case that a system is to be used on a large scale. Such applications may require significant custom development.

Compiled Programs: Programs written in MATLAB m-code can be compiled as "stand-alone" programs, where they can then be integrated as components in a larger system. The Mathworks supplies several products for compiling m-code, and Distribution Licenses for m-code contained in the PLS_Toolbox are available upon request. MATLAB is compliant with Microsoft's .NET/COM framework, and thus can be compiled as an EXE, DLL, COM/ActiveX object, or called from within another program (like Excel using VBA) as an ActiveX object. Programs can also be compiled for use on other platforms (Unix/Linux).

This option is often considered when one wants to distribute a MATLAB model-application to several different systems. One could purchase separate PLS_Toolbox and MATLAB licenses for each system, but this might be costly. Alternatively, one could use the MATLAB Compiler and a PLS_Toolbox Distribution License to compile the program into a “stand-alone” EXE, DLL or COM/ActiveX Object, and distribute it with a much lower fixed cost. The compiled objects run with the MATLAB Component Runtime stand-alone server.

Advantages with this type of approach are reduced cost and portability. Cost reductions come in the form of limiting the number of software licenses needed (as discussed above). Making your program portable and available to other programmers (particularly those using Microsoft technologies) within your company may be as easy as exporting it as a COM object. Finally, there are some situations when compiling a program to a DLL, for instance, will yield an increase in speed of execution.

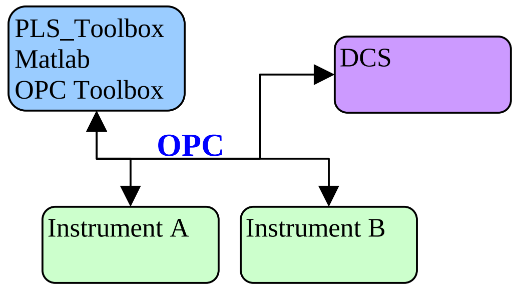
Possible disadvantages of this approach include higher initial costs (if additional software products are needed), less flexibility, and more difficult troubleshooting. Depending on the changes and design of the application, one may have to re-compile and re-distribute a new application every time there's a change. This also complicates troubleshooting a program.

OPC: OPC is a set of standards for communicating in a “process” environment. The standards outline an exact protocol for communication between devices and enables devices from different manufacturers to exchange data in a standardized fashion.

With the aid of the Mathworks' OPC Toolbox, MATLAB and PLS_Toolbox can act as an OPC Data Access (DA) client, sending requests to OPC servers for data. In one such example, a PLS_Toolbox/MATLAB client requests data from an OPC server associated with instrument A, applies a model to that data, and then writes the results to a ***different*** OPC server that is associated with instrument B.

The advantages of using OPC are numerous. The technology is robust and provides a cross platform solution for online systems. The OPC Toolbox provides programming tools via GUIs and command line functions/objects. It also allows one to assign MATLAB Callbacks (specific program code) to a number of different server events, which is an easy and powerful way of triggering a program to respond in an online environment. One can also view "live" data as well as logged data over time. All MATLAB command line functions and objects can be compiled. As a result, OPC can be especially useful during application development, to test model execution code in a real-time environment before actual deployment.

Disadvantages of OPC include costs and unsupported/legacy systems. Most process hardware manufacturers support OPC, but in some cases implementing the standard can require the purchase of or updating of SCADA systems. Along these lines, a legacy device that is not OPC-enabled may require the purchase of third party software/hardware to support its integration into an OPC system.



Additional Information:
PLS_Toolbox
- www.software.eigenvector.com

Other Mathworks Products -
- OPC Toolbox www.mathworks.com/products/opc/
- MATLAB Compiler www.mathworks.com/products/compiler/
- MATLAB Builder for .NET (COM) www.mathworks.com/products/netbuilder/

OPC
- www.opcfoundation.org