

Jouni Savolainen, Jari Lappalainen, Hannu Mikkonen
and Mikko Jegoroff

**Methods to link the sub-models between
Aspen and Apros dynamic simulation
softwares**



CLEEN LTD
ETELÄRANTA 10
P.O. BOX 10
FI-00130 HELSINKI
FINLAND
www.cleen.fi

ISBN XXX-XX-XXXX-X
ISSN XXXX-XXXX

Cleen Ltd.
Research Report nr 305

Jouni Savolainen, Jari Lappalainen, Hannu Mikkonen
and Mikko Jegoroff

**Methods to link the sub-models between Aspen and
Apros dynamic simulation softwares**



ccsp

Carbon Capture and Storage Program

Cleen Ltd
Helsinki 2010



Report Title: Methods to link the sub-models between Aspen and Apros dynamic simulation softwares

Key words: Apros, Aspen Plus, dynamic modelling, Matlab OPC

Abstract

This report summarises the study of how to connect Apros and Aspen Plus Dynamics simulators. The aim of this study was to evaluate different linking techniques to integrate two separate dynamic modelling programs, Aspen Plus Dynamics (Version 7.2) and Apros (Version 5.10.05). After successful linking the next target will be to use earlier developed models and link further process main flows between models.

The process concept consists of a CFBC boiler which can be driven both in air- and oxy –mode and it is modelled Apros. Because of better air distillation and CO₂ –separation properties Aspen Plus Dynamics was used for building up Air Separation Unit (ASU) and CO₂ purification unit (CPU) –models. This is the reason why linking techniques have to be evaluated.

Helsinki, April 2012



Table of contents

1	Introduction	2
2	Initial screening of dynamic linking techniques.....	2
2.1	OLE for process control (OPC).....	2
2.2	Linking supported by Excel	5
3	Evaluated linking alternatives	5
3.1	Direct OPC connection.....	5
3.2	Excel-based connection.....	6
3.3	Matlab/Simulink-based connection	6
3.3.1	System requirements	7
3.3.2	Configuring the system	8
3.3.3	Simulink configuration	8
3.3.4	Aspen connection configuration	9
3.3.5	Apros connection configuration	10
3.3.6	Simulation cycle of the system.....	12
3.3.7	Example.....	13
4	Conclusions and further research	14
5	References.....	15

1 Introduction

A study has been carried out in VTT Jyväskylä and Espoo, Finland in Autumn 2011. The aim of this study was to evaluate different linking techniques to integrate two separate dynamic modelling programs, Aspen Plus Dynamics (Version 7.2) and Apros (Version 5.10.05). After successful linking the next target will be to use earlier developed models and link further process main flows between models. The process concept consists of a CFBC boiler which can be driven both in air- and oxy –mode and it is modelled Apros. Because of better air distillation and CO₂ –separation properties Aspen Plus Dynamics was used for building up Air Separation Unit (ASU) and CO₂ purification unit (CPU) –models. This is the reason why linking techniques have to be evaluated.

2 Initial screening of dynamic linking techniques

Aspen Plus Dynamics is a trade mark of Aspentech and from there started the evaluation of the techniques. A specialist of Aspentech (Emmanuel Lejeune) informed of the possibilities from the Aspen Plus Dynamics point of view. Dynamic Data Exchange -technique (DDE) is not supported in Aspen Plus Dynamics. Also using of Dynamic Link Library –technique (DLL) was not supported. This led to available alternatives being Ole for Process control –technique (OPC) and linking supported by Excel.

2.1 OLE for process control (OPC)

OLE for Process Control (OPC), which stands for Object Linking and Embedding (OLE) for Process Control, is the original name for a standards specification developed in 1996 by an industrial automation industry task force. The standard specifies the communication of real-time plant data between control devices from different manufacturers. After the initial release, the OPC Foundation was created to maintain the standard. Since then, standards have been added and names

have been changed. As of June, 2006, "OPC is a series of standards specifications". (Seven current standards and two emerging standards.) "The first standard (originally called simply the OPC Specification)", is "now called the Data Access Specification", or (later on the same page) "OPC Data Access", or OPC Data Access Specification. While OPC originally stood for "OLE for Process Control", the official stance of the OPC Foundation is that OPC is no longer an acronym and the technology is simply known as "OPC". One of the reasons behind this is while OPC is heavily used within the process industries, it can be, and is, widely used in discrete manufacturing as well. Hence, OPC is known for more than just its applications within process control.

The OPC Specification was based on the OLE, COM, and DCOM technologies developed by Microsoft for the Microsoft Windows operating system family. The specification defined a standard set of objects, interfaces and methods for use in process control and manufacturing automation applications to facilitate interoperability. The most common OPC specification is OPC Data Access, which is used to read and write real-time data. When vendors refer to OPC generically, they typically mean OPC Data Access. OPC DA itself has gone through 3 major revisions since its inception. Versions are backwards compatible, in that a version 3 OPC Server can still be accessed by a version 1 OPC Client, since the specifications add functionality but still require the older version to be implemented as well. However, a Client could be written that does not support the older functions since everything can be done using the newer ones, so a DA 3 compatible Client will not necessarily work with a DA 1.0 Server.

OPC was designed to provide a common bridge for Windows based software applications and process control hardware. Standards define consistent methods of accessing field data from plant floor devices. This method remains the same regardless of the type and source of data. An OPC Server for

one hardware device provides the same methods for an OPC Client to access its data as any and every other OPC Server for that same and any other hardware device. The aim was to reduce the amount of duplicated effort required from hardware manufacturers and their software partners, and from the SCADA and other HMI producers in order to interface the two. Once a hardware manufacturer had developed their OPC Server for the new hardware device their work was done to allow any 'top end' to access their device, and once the SCADA producer had developed their OPC Client their work was done to allow access to any hardware, existing or yet to be created, with an OPC compliant server.

OPC servers provide a method for many different software packages (so long as it is an OPC Client) to access data from a process control device, such as a PLC or DCS. Traditionally, any time a package needed access to data from a device, a custom interface, or driver, had to be written. The purpose of OPC is to define a common interface that is written once and then reused by any business, SCADA, HMI, or custom software packages.

Once an OPC server is written for a particular device, it can be reused by any application that is able to act as an OPC client. OPC servers use Microsoft's OLE technology (also known as the Component Object Model, or COM) to communicate with clients. COM technology permits a standard for real-time information exchange between software applications and process hardware to be defined.

It is important to note that some OPC specifications are published, others are available only to member of the OPC Foundation. So whilst no company "owns" OPC and anyone can develop an OPC server, whether or not they are a member of the OPC Foundation, non-members will not necessarily be using the latest specifications. Anyone can integrate OPC products, and there is no pre-requisite for the system integrator

to belong to any organization. It is therefore up to each company that requires OPC products to ensure that their products are certified and that their system integrators have the necessary training /1/.

2.2 Linking supported by Excel

Using Excel macros was one option for the connection. It is possible to write the macros to drive Aspen Plus Dynamics from Excel. The advantage of this approach is that process made by Aspen Plus Dynamics is fully controlled. The disadvantage is that code writing has to be made. A script can be carried out for example in a way that the it opens Aspen Plus, runs and export as f-driven, then closes Aspen Plus. Then the excel file has to be opened and Open button has to be pressed to open the dynamic simulation file you've created.

3 Evaluated linking alternatives

Three alternatives for linking Apros and Aspen Plus Dynamics were evaluated:

1. Direct OPC connection between Apros and Aspen Plus Dynamics
2. Excel-based connection
3. Matlab/Simulink-based connection.

These are described in more detail below.

3.1 Direct OPC connection

The direct OPC connection was realized with Aspen Plus Dynamics's OPC tools and Apros's OPC Server. First, the test was carried out with a "toy model" which did not directly relate to Oxyfuel processes. It was already in these tests noticed that connecting the simulators with this technique seems to be infeasible. The main reason for this was that both Apros and Aspen Plus Dynamics will simulate (i.e. perform numerical integration of their model equations) independently of each other. In other words, there is no guarantee that Apros's simulation clock will run at the same pace a Aspen Plus

Dynamics's. This will pose problems especially when the two simulators run at considerable speed difference, since the faster simulator's clock will run ahead of the slower's. Thus the values transferred from simulator to another will not have any relevance to each other. After this was noticed, this alternative was abandoned.

3.2 Excel-based connection

The second connection alternative was to use Excel as a mediator between Apros and Aspen Plus Dynamics. The basic idea was to have Excel command both simulators to simulate for a certain period, say 1s, and then wait until both simulators have done their job. Next data would be transferred from simulator to another and the process repeated. This way the two simulators' clocks would remain synchronized. Aspen Plus Dynamics offered readymade connection to Excel and example Visual Basic codes to achieve this. On the other hand, there is no direct way to connect Apros to Excel. Thus, it was investigated whether we could utilize OPC to connect Excel and Apros. Three free Excel OPC toolkits were tested, namely

- the OPC Office Link by Rensen Information Services Ltd. (<http://www.ris.co.nz/>)
- Resolvica's OPCEX (<http://www.resolvica.com/>)
- Cogent's OPC DataHub (http://www.opcdatahub.com/Features/OPC_to_Excel.html).

All three products offer easy ways to get data from an OPC server to Excel sheets, but controlling a simulator through the link proved to be difficult or even impossible. Such control actions which were found to be missing was for example loading simulation models. For this reason the Excel-based connection was abandoned too.

3.3 Matlab/Simulink-based connection

The third connection alternative was to use Matlab's Simulink as a mediator between Apros and Aspen Plus Dynamics. The

connection from Simulink to Aspen Plus Dynamics would be realized with Aspentech’s own interface and the connection from Simulink to Apros with OPC (using the “OPC for Matlab” - product by Ipcos (http://www.ipcos.com/en/opc_for_matlab)).

The entire linking system is depicted in the following **Error! Reference source not found..**

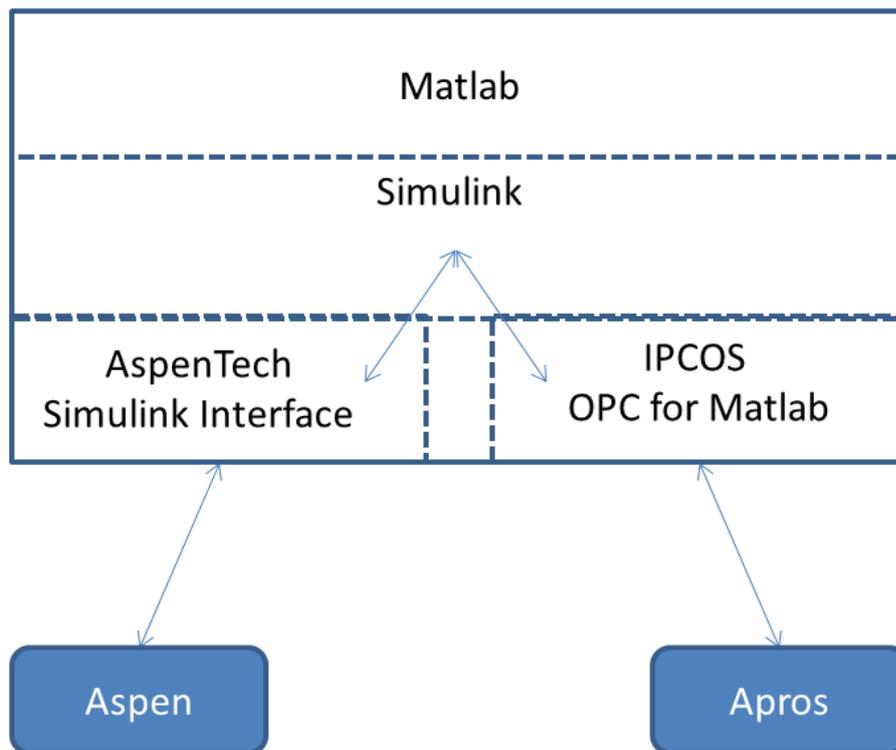


Figure 1. Matlab/Simulink-based communication.

This system was tested with “toy models” and with and ASU+Boiler model (ASU in Aspen Plus Dynamics and boiler in Apros). Although some performance issues were raised this way of linking proved to be most promising. Details of the link are described next.

3.3.1 System requirements

The Apros-Aspen link using Matlab/Simulink needs the following:

1. As fast a PC as possible
2. A PC with as much RAM as possible
3. Aspen Plus Dynamics-software

4. Apros-software
5. IPCOS OPC for Matlab-software, note that the IPCOS tool requires 32-bit Matlab installation
6. Matlab-software with Simulink

3.3.2 Configuring the system

When configuring the system for the first time the following input information is needed:

1. List of variables to be transferred from Aspen Plus Dynamics to Apros
2. List of variables to be transferred from Apros to Aspen Plus Dynamics
3. Name of utilized Aspen Plus Dynamics model
4. Name of utilized Apros model
5. Name of Apros OPC Server, host name where this server runs, and OPC communication frequency

3.3.3 Simulink configuration

The system is started by opening Matlab and typing *simulink*. This opens Simulink and the empty connection model *AprosAspenLink (.mdl)* can be opened. The model is depicted in Figure 2.

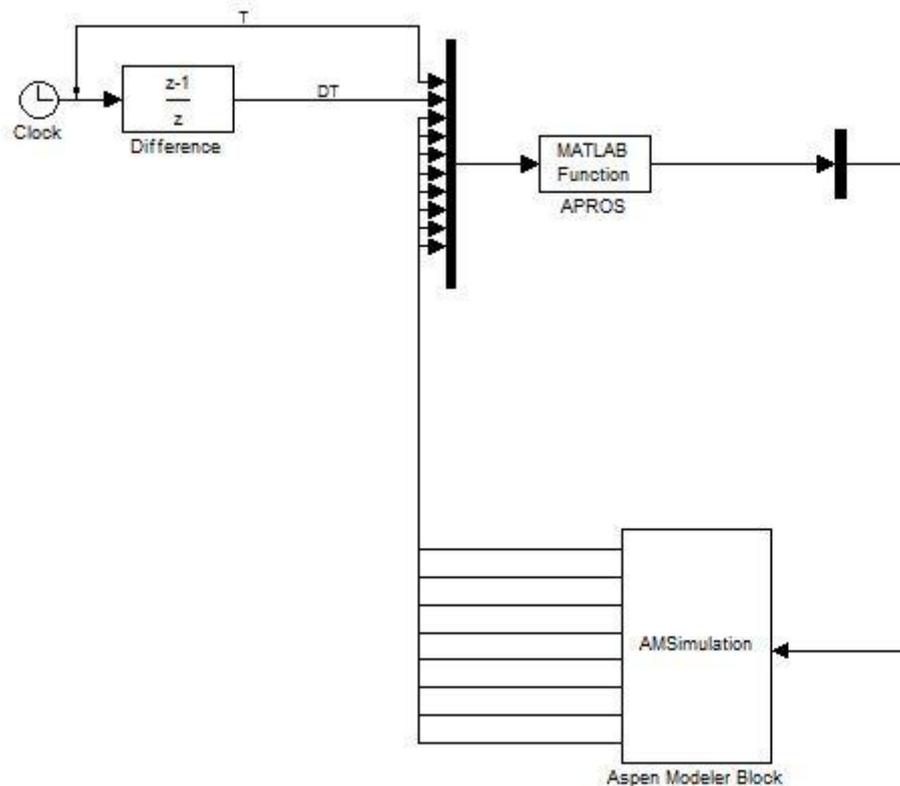


Figure 2. Example of the Apros-Aspen linking tool in Simulink.

In Figure 2's example we have one variable transferred from Apros to Aspen and eight variables transferred from Aspen to Apros. The number of variables is configured to both APROS and Aspen Modeller blocks as describe later. In addition to these, the APROS block takes in the current Simulink simulation (T) and the time step (ΔT).

3.3.4 Aspen connection configuration

The configuration of the Aspen Modeler Block is done by opening its configuration dialog by double-clicking the symbol on the Simulink diagram. For the Aspen Modeller Block one first defines which model (.dynf) is to be used. Next from the Input and Output tabs one defines which values are written to Aspen (Input) and which are read from it (Output). An example of this configuration dialog is shown in Figure 3.

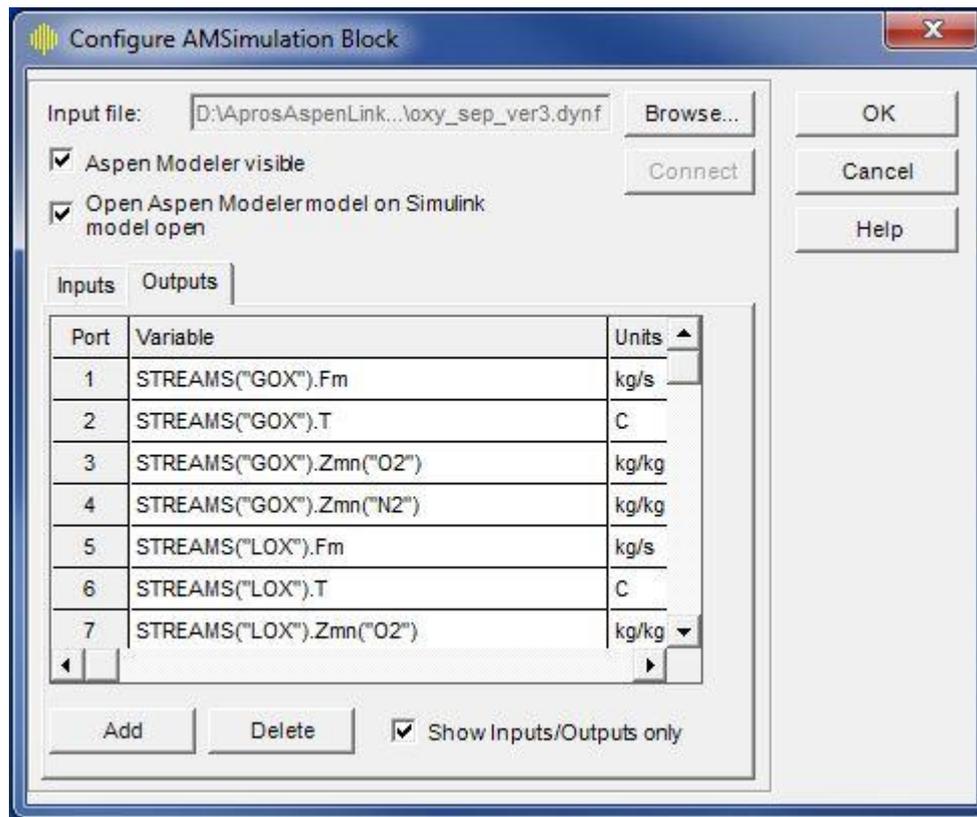


Figure 3. Aspen Modeler Block configuration dialog.

On the top of Figure 3 there is a "Browse..." -button which is used to select the correct Aspen model.

The Aspen Inputs and Outputs are defined at the bottom part of the dialog. To add a new variable first the Add-button is pushed and the drop-down menu appearing on the new line is used to select the correct Aspen model variable.

3.3.5 Apros connection configuration

For the APROS block configuration of the block is done in the Matlab-function `simApros`, which is in the m-file `simApros.m`. This code is called every Simulink time step and performs the needed simulation operations.

In the configuration phase the two variables `readTag` and `writeTag` need to be edited by the user in order to accommodate the case at hand. Both of these are vectors, each element of the vector containing a string. Each string defines one tag for the Apros OPC connection. The general form of a tag is:

`<module name>!<attribute name>`

For example to access the pressure of a POINT named N1_PO01 the tag would be:

N1_PO01!PO11_PRESSURE.

In the variable *readTag* the user defines all tags that are read from Apros to Simulink and in *writeTag* the tags written from Simulink to Apros are defined. It should be noted that each row in *readTag* must be of the same length, so trailing spaces may need to be appended. The same applies for *writeTag*.

Furthermore, in the configuration phase of the Apros-part the user must define how the OPC-connection is made. This is done in the m-file *connectToApros.m*. Here the user needs firstly to define the name of the simulation model to be used, for example 'abc7d'.

Next the name of the OPC-server is defined. The easiest way to get this is to select from Grades the option 'Exchange - OPC Server name to clipboard' and to paste this to the m-file. A typical example of the OPC server name is:

VTT.AprosOPCServer+CombustionProject+Workspace:8924

Next the IP address of the machine in which Apros is running needs to be defined. In most cases where Apros and Matlab/Simulink reside on the same machine, this variable is set to '*localhost*'.

The last parameter which defines the frequency of the OPC connection can be left to 1000 (ms). This parameter does not have much meaning since Simulink and simApros control the communication over OPC to Apros. Nevertheless, it needs to be defined.

After these parameters are defined the connection from Matlab/Simulink to Apros needs to be made. This is done by running the function *connectToApros*. This function

1. makes the connection,
2. loads the defined model, and
3. sets the Apros simulation clock to 0.0.

After this the system is ready for simulation.

3.3.6 Simulation cycle of the system

Simulation of the system is done from Simulink. In Simulink the user defines how long the simulation is to be, in seconds. Also the user has a choice of numerical integration method, used by Simulink. As there is no numerical integration to be done in Simulink it is advisable to follow the recommendations of AspenTech: “Within Simulink, the variable step ode45 (Dormand-Prince) integrator is recommended. However, you can use other variable step integrators”. Having defined these the running of the simulation is started by pressing the *Play* button of Simulink.

When the simulation is started the Aspen model is reset to time zero. For the Apros model this is done already in the connection phase by *connectToApros*. Next the simulation cycle proceeds block-by-block in Simulink. For example, if the cycle starts from the Aspen Modeller block the following steps are taken:

1. Data is written to Aspen Plus Dynamics (*AMSimulation.m*)
2. Aspen Plus Dynamics is ordered to simulate for ΔT seconds (*AMSimulation.m*)
3. Data is read from Aspen Plus Dynamics (*AMSimulation.m*)
4. Data is written to Apros (*simApros.m*)
5. Apros is ordered to simulate for ΔT seconds (*simApros.m*)
6. Data is read from Apros (*simApros.m*)

The time to simulate during one cycle, ΔT , is given by Simulink's solver. This six-step cycle is repeated until the defined final time of simulations, T , is reached. During the simulation data can be logged to files or to the screen from Aspen Plus Dynamics, Simulink or Apros. By default, the Simulink connection model does not log anything.

After the simulation is finished it is advisable to close the connection to Apros by giving the command

```
mxOPC('close')
```

at the Matlab command prompt. The reason for this is that if a new simulation is immediately done, the Apros clock will remain in the value where the previous simulation ended but Aspen's clock is reset when the second simulation starts. To reset the Apros clock the *connectToApros* function needs to be run.

3.3.7 Example

This example describes the test case with ASU in Aspen Plus Dynamics and the boiler in Apros. Models utilized were:

Aspen: *oxy_sep_HM_pressure_driven_compressor.dynf*
Apros: *abc7d*

The transferred data is defined Table 1.

Table 1. Data items transferred in the test case.

#	Description	Aspen Plus Dynamics variable	Dir.	Apros Tag
1	Air compressor controller setpoint	B14_RSP	←	INT_XA29!ANALOG_VALUE
2	GOX mass flow, kg/s	STREAMS("GOX").Fm	→	INT_XA01!ANALOG_VALUE
3	GOX temperature, C	STREAMS("GOX").T	→	INT_XA03!ANALOG_VALUE
4	GOX O ₂ mass fraction, kg/kg	STREAMS("GOX").Zmn("O2")	→	INT_XA02!ANALOG_VALUE
5	GOX N ₂ mass fraction, kg/kg	STREAMS("GOX").Zmn("N2")	→	INT_XA04!ANALOG_VALUE
6	LOX mass flow, kg/s	STREAMS("LOX").Fm	→	INT_XA05!ANALOG_VALUE
7	LOX temperature, C	STREAMS("LOX").T	→	INT_XA06!ANALOG_VALUE
8	LOX O ₂ mass fraction, kg/kg	STREAMS("LOX").Zmn("O2")	→	INT_XA07!ANALOG_VALUE
9	LOX N ₂ mass fraction, kg/kg	STREAMS("LOX").Zmn("N2")	→	INT_XA08!ANALOG_VALUE

The procedure is summarised as:

- start Matlab
- start Simulink
- start Apros
- open correct Simulink model, in this case `simulinkAprosAspenPressureDrivenCompressor`
- make Aspen Modeler block definitions according to Table 1
- make modifications to `simApros` (according to Appendix A)
- make connection to Apros:
`connectToApros('VTT.AprosOPCServer+CombustionProject+Workspace:8920','abc7d',1000)`
- define Simulink simulation end time: 60s
- simulate
- close connection to Apros: `mxOPC('close')`

4 Conclusions and further research

This report presented results from the evaluation of techniques to connect Apros and Aspen Plus Dynamics simulators. The conclusion was that using Matlab/Simulink as the intermediary is recommended. The tests showed that

- data can be transferred from simulator to another
- simulator clocks can be kept in synch
- models can be (re)loaded and simulation controlled using the system.

The evaluation raised some future research issues. Firstly, it was noticed that when Aspen Plus Dynamics is ordered to simulate for ΔT every simulation cycle, it takes a while to do some preparation actions, before the simulation actually starts to run. This slows the system down. Thus it is recommended that the computer in which the system is used, be as powerful as possible. This issue can also be alleviated by increasing the communication cycle (i.e. Simulink time step), as much as the Apros and Aspen models allow. Secondly, it was noticed that the simulation models should be individually tested before connecting them. In particular it was noticed that the tested Aspen model was rather unstable and long simulations were impossible.



5 References

1. Wikipedia. URL
<http://en.wikipedia.org/wiki/OLE_for_process_control>. Referred
3.4.2012.