Technical Note: SN-503

PERSPECTIVES ON DATA SYNCHRONIZATION AND ALIGNMENT FOR PAT

In implementing process analytical systems, one is often faced with the need to insure that all of the measurements refer to the same process state, or at least to states which are related in a well defined and validatable fashion. Poor synchronization or alignment of observations can significantly devalue the quality of models or inferences which can be made. The task is compounded by the fact that some types of measurements will involve data acquisition over en extended period of time while others – such as chromatographs – involve time delays between sample acquisition and reporting of the measurement to the data system.

This Technical Note addresses the use of the Proficy RX and Symbion-RX/DX process analytical software suites to insure optimum synchronization and alignment of process data. Our approach is twofold. First, when designing a measurement configuration for a new process analytical installation, we suggest imposing synchronization by the controlled coordination of instruments. By triggering the acquisition of various instruments in a controlled and organized fashion, the data will naturally populate into the repository with a meaningful timestamp for a-posteriori analysis and model development. The data will also arrive as appropriate for use with on-line and coordinated deployment strategies. The first part of the discussion below addresses situations which are amenable to measurement synchronization. The second part of the discussion provides an approach to dealing with data that has already been stored or processes that are not amenable to controlled coordination of instruments.

Part I: Coordinated Control Example:

This example assumes the inclusion of an FT-NIR instrument, a rapid-scan NIR spectrometer, an HPLC, and a process input such as temperature of the process. The data from the various devices are to be combined into a single model and then information from the model will be posted to a SCADA/HMI system.

Controlled Coordination and Synchronization:

The coordination of data acquisition from the various devices is crucial to the quality of the overall process observation. If the various devices are simply allowed to run in an uncontrolled fashion, the quality of the information that can be gleaned from these process observations may be compromised. However, if we can coordinate the data acquisitions from the various instruments, so that the independent observations are highly correlated with each other, then the mathematical post-treatments and overall modeling/inference from the data for offline or line work will be greatly improved.

In this example, we propose synchronizing the instruments so that the FT-NIR is started first, followed by the NIR spectrometer, followed by the temperature input, and lastly the injection time for the HPLC. This overall coordination is designed to center all of the observations around the injection time of the HPLC for obvious reasons. Mathematically the timing for each instrument would be represented as

$$t_{\text{FT-NIR}} = t_i$$
$$t_{\text{NIR}} = t_i + delay_1$$
$$t_{\text{Temp}} = t_i + delay_1 + delay_2$$
$$t_{\text{HPLC}} = t_i + delay_1 + delay_2 + delay_3.$$

where t_i represent the start time for each coordinated experiment, $delay_1$, $delay_2$ and $delay_3$ represent the necessary delays to center the collections about the HPLC injection time. The instruments with the longest collections times are started first.
The calculations of each of the delays could be computer on-line from the instrument parameters or could be determined by experimentation. In most cases, the delays can be determined by the specified collection parameters for each instrument. For example, let the coordinated sample time be denoted as t_s. In this hypothetical example, the sample time would be equal to the HPLC injection time $t_s = t_{\text{HPLC}}$. The total collection times for the various instruments can be related to the above timing definitions with the collection centered at the HPLC injection time as

$$t_s - t_{\text{FT-NIR}} = \frac{1}{2}(\text{FT-NIR Collection Time})$$
$$t_s - t_{\text{NIR}} = \frac{1}{2}(\text{NIR Collection Time})$$
$$t_s - t_{\text{Temp}} = \frac{1}{2}(\text{Temperature Collection Time})$$

and the required delays can be computed from the solving for the above equations simultaneously yielding

$$delay_1 = t_{\text{NIR}} - t_{\text{FT-NIR}} = \frac{1}{2}(\text{NIR Collection Time}) - \frac{1}{2}(\text{FT-NIR Collection Time})$$
$$delay_2 = t_{\text{temp}} - t_{\text{NIR}} = \frac{1}{2}(\text{Temp. Collection Time}) - \frac{1}{2}(\text{NIR Collection Time})$$
$$delay_3 = t_s - t_{\text{temp}} = \frac{1}{2}(\text{NIR Collection Time}) - \frac{1}{2}(\text{FT-NIR Collection Time})$$

It is assumed that the start time for each coordinated collection t_i would be received from a fixed-frequency clock signal. The separation between successive start times $t_i - t_{i-1}$ would have to be longer than the time to complete the longest observation in a given coordinated collection.

RX Control Script:

Since the RX embedded automation language is designed specially for PAT related scenarios, a set of RX scripts that could accomplish this coordination would be straightforward. Since many of instrument vendors software only allow for synchronous operation, a combination of multiple scripts are necessary to start the staggered acquisitions.

Note: Many of the instrument vendors have created OPC interfaces that can operate instruments asynchronously in recent years. Thus, there scripts assume the worst-case scenario where the instrument vendor's software assumes synchronous operation (non-event-driven). The Analect and Yokogawa drivers for RX have both synchronous and asynchronous modes of observations due to the underlying instrument vendor's software.

```
set x1 [ControlInstrument Bruker 64 Absorbance {xpm = C:/OPUS/XPM/MYXPM.XPM}, process = 1]

Pause 120000

set x2 [ControlInstrument Axsun 20 Absorbance {xstart = 1250 , xstop = 1550 , xstep = 0.5}, process = 2]

Pause 20000

set x3 [ReadAO OPC  0 0 1,  process = 3]

Pause 2000

set ts [ GetCuurentTimeStamp]

set x5 [ControlInstrument HPLC "" "" ""]

set x6 [Concat { $x1 , $x2 , $x3 , $x4 } ]

SaveHistSpec BatchID $x6 "timestamp = $ts"

set x7 [RunPrediction Simca $x6 {cal  = C:/mymodels/myuspmodel.usp} 0]

SaveHistTag BatchID.tagname1 $x6 "timestamp = $ts"

WriteAO OPC 0 0 1 {} $x7

PlotRTTag …
```

This script will start the operation of each instrument in sequence and will then wait at the Concat command for all data to arrive. The combined data will be stored with the BatchID and the coordinated timestamp (time $t_s = t_{HPLC}$). Individual instrument observations could be also be stored. However, for brevity of this example, multiple instanced of the SaveHistSpec commands have been omitted. The Pause commands are in milliseconds. The individual concentration tags can be saved with the HPLC injection timestamp (time $t_s = t_{HPLC}$). .

Assumptions/Developments:

Implementation of the above scenario requires some assumptions and developments. These are as follows.

1. As stated above, some spectrometers may already operate asynchronously, and as a result, will not require multiple instances of RX. If this is not the case, then the spectrometers run from the ControlInstrument instances must be operated from different copies of RX. With the current version of RX, multiple copies can be run on the same processor via VMWare. Future releases will allows for multi-threading. The ControlInstrument Command will be enhanced to allow for the specification of which copy of RX to be run.

2. The temperature observation could simply be obtained by polling a moving-average temperature value which would not require a separate process to be run.

3. This script assumes the existence of an HPLC driver, yet to be developed.

4. The RX Concat command will need to be modified to have a wait-state to wait until all inputs have arrived.

5. A command GetCurrentTimeStamp will need to be created in order to obtain the current clock time at the HPLC injection time

6. The RX SaveHistSpec command will need be modified to allow for an external clock input.

All of the previous modifications are relatively simple with the exception of the HPLC interface.

Chemometric Processing Delays:

The previous example assumes the delay in the processing of the chemometric model (t_s - t_{chemo}) is short enough to avoid overruns into the next observation cycle. Here, t_s is – as before – the sample time (in this example the same as HPLC injection time) and t_{chemo} is the time in the cycle in which the chemometric processing is completed. This may not always be the case. To handle this situation the Concat command requires an input to point the input data to the database versus a local variable (x_1, x_2 …). If it is of interest to the reader an example can be provided.

Tiered Control:

The delay constants, collection parameters, and calibration models can be made variables (parameterized) so that they can run in a generic script from any process tier (1, 2, and 3).

Part II: Coordinating Data from Un-Coordinated Sampling

The RX product already has the basis for a tool to coordinated independent observations that occur at the same time. For example, the `SimcaExport` command (See Simca Supplement) allows for the synchronization of instrument observations and other independent univariate observations (process and processed) in order to produce a structured observation matrix amenable for import into the major chemometrics packages. The synchronization of multivariate and univariate data is via nearest neighbor approach. A typical combined observation matrix M_{exp} would look like:

$$M_{exp} = \left[T \mid X_1 \quad X_2 \quad \cdots \quad X_n \mid Y \right]$$

which expands to

$$M_{exp} = \left[\begin{array}{c|cccc|cccc} t_1 & x_{11} & x_{21} & \cdots & x_{n1} & y_{11} & y_{21} & \cdots & y_{m1} \\ t_2 & x_{12} & x_{22} & \cdots & x_{n2} & y_{12} & y_{22} & \cdots & y_{m2} \\ \vdots & & \ddots & & & & \ddots & & \\ t_k & x_{1k} & x_{2k} & \cdots & x_{nk} & y_{1k} & y_{2k} & \cdots & y_{mk} \end{array}\right]$$

where

T	=	time index for each observation such that $T \in \{t_1 , t_k\}$
X_1	=	first univariate observation matrix such that $x_1 \in x_{11}, x_{12}, \ldots x_{1k}$
X_2	=	second univariate observation matrix such that $x_2 \in x_{12}, x_{22}, \ldots x_{2k}$
X_n	=	nth univariate observation matrix such that $x_n \in x_{n2}, x_{n2}, \ldots x_{nk}$
Y	=	spectral observation matrix, such that $y_1 \in y_{11}, y_{21}, \ldots , y_{m1}$
m	=	size of the independent spectral observation
k	=	total number of observations
n	=	number of univariate non-spectral observations.

In addition to the spectral data, meta data that was stored with the spectra is placed into columns just prior to the actual numerical data. This allows for sorting of instrument data on the various instrument meta data columns. A user may wish to use only data from spectrometers with a unique serial number.

For the previous example, the combined spectral data would be stored into the rows of the *Y* matrix. Other instances of the SaveHistSpec commands could be added to store each of the instrument observation independently. The data could be combined using this approach as well. The spectral data meta data block would contain the name of each instrument along with other instrument meta data for use with combining coordinating data sources.

This export utility would need to be enhanced for situations like the integration of HPLC data. Additional query inputs would need to be added to allow for searching the different instrument inputs. Allowing for a delay times associated with different instruments would aid in forming a proper Mexp. If the time offset between different instrument observations is known, then this numeric value could be incorporated into the query so as produce a Mexp that contains the best possible correlated experimental observations and thus will lead to enhanced model development and data inference.