How to Use Subspace Identification

-A Tutorial and Application Tips

Aspen Technology, Inc.

Table of Contents

1 Progress in Model Identification from FIR to Subspace Methods

1.1 FIR Model Identification by Least Square (LS) algorithm

FIR model identification has been available for several decades, and has been used in thousands of successful DMCplus applications. When the original [DMC] controller technology was built around two decades ago, the alternative model identification algorithms, especially those based on transfer function models, did not work well enough for typical process applications. Low order transfer functions have the benefit of always resulting in smooth models, since only a small number of parameters are used to explain an entire data set. However, for processes with significant heat integration, recycles and interacting PID controllers, these model structures yield poor results. Given the state of the art in 1980's, it was simply not possible to fit complicated responses using low order transfer functions, and the more flexible FIR model structure was selected.

1.1.1 Benefits of the FIR Model Identification Method

In contrast to the limitations of transfer function methods, Finite Impulse Response (FIR) methods have a major advantage in being able to fit very complicated process responses, since a large number of model parameters are available, typically 90 to 120 coefficients per curve. It is essentially a free-form non-parametric identification method and offers a high degree of freedom to fit process response curves. However, this comes at a price: It often finds a solution with a lower least squares cost function by "bending" the model curves to fit disturbances as dynamics, often regarded as "over fitting" and it also tends to be sensitive to correlation between MVs. This makes it important to conduct a good step test with minimal MV cross correlation, make many and large enough steps in the MVs, include all available feed forward information in the cases, builds event vectors, and carefully slice out or modify bad data pieces.

Because of the high degree of freedom that a FIR model can offer when fitting a model to a dataset, the FIR models can easily fit high order effects. In the example shown below, note that several of the FIR models have complicated under-damped responses with overshoot, and some MV/CV relationships display an inverse response where the CV first go in the "wrong" direction before turning around and going in the expected direction and then lining out. In general, FIR is capable of fitting accurate models to the relationships that benefit from a high Signal to Noise (S/N) Ratio. A high S/N ratio results from significant movement in the CV due to steps in that MV.

1.1.2 Disadvantages of the FIR Model Identification Method

The disadvantage of a large number of model parameters is that the models can be noisy, especially if the data set is small or the signal to noise ratio is poor. FIR model identification is classified as non-parametric model ID algorithm; generally it requires

high quality test and more data in order to achieve good quality models. In model structure, FIR algorithm needs to solve many multi-input single-output (MISO) sub-model ID problems for a general multi-input multi-output (MIMO) ID problem. As a result, the cost of building a MIMO process model is high. In addition, the FIR ID algorithm has following shortcomings in practice:

1.1.3 Effect of Slicing on Data Loss

If the data consists of multiple slices, for each data slice one full Time to Steady State (TTSS) of data has to be used for prediction re-initialization. For long TTSS processes, this obviously leads to significant data loss, and it is important to prevent valve saturation etc. to minimize loss of data.

1.1.4 Truncation Errors

If the user selects a model with too low a Time to Steady State (TTSS) value, the model curves will be truncated (cut off) too early, and the gains will be inaccurate due to significant truncation errors. It is therefore important that the user makes large steps that are also held for long enough in plant test to ensure that gain accuracy is good. On the hand, a long TTSS means higher model order and more parameters in the model have to be estimated.

1.1.5 FIR Model Identification When the S/N Ratio is Poor:

For short data sets with a low S/N ratio, the FIR model curves may display some unwarranted squiggles. A poor Signal to Noise Ratio could be due to several factors:

- Low S/N Ratio is inevitable for Weak MV-CV Relationships: For the weaker process responses, it is simply not possible to make big enough steps to properly perturb the CVs with weak responses against this MV. This is due to other CVs having strong relationships (large gains) relative to this MV. These CVs will hit their CV limits first, and will limit the size of the MV step. In practice, those weak MV-CV relationships are often ignored in the final model matrix though.

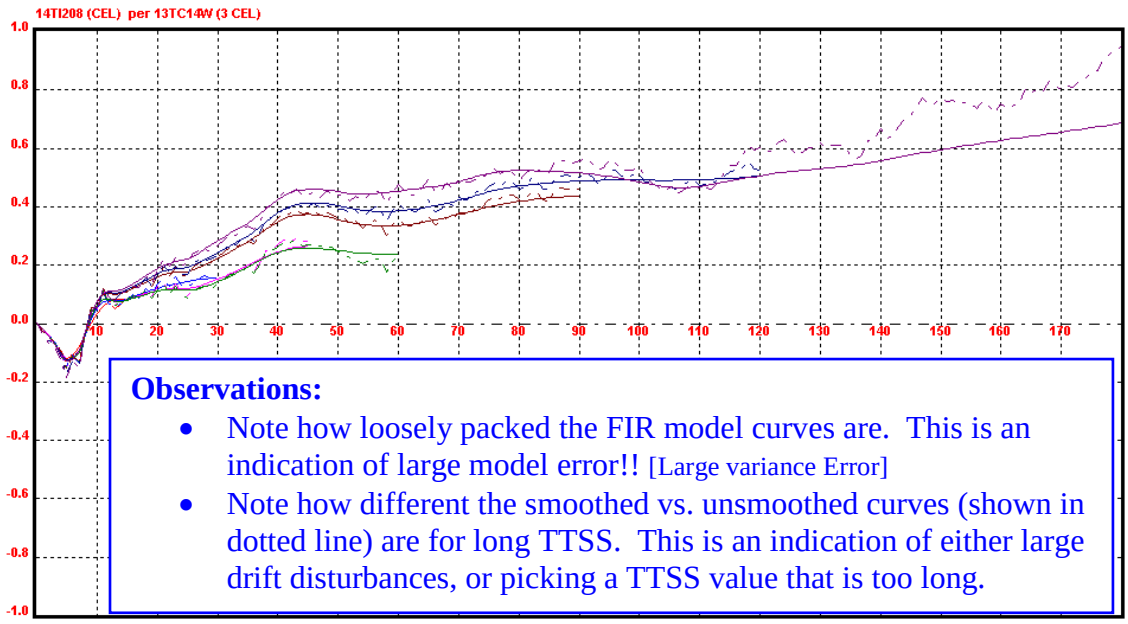
- Presence of Strong Disturbances that CANNOT be measured: If major unmeasured disturbances are present, then the models could be adversely affected. The solution is to find or build suitable FF signals, or to slice out this data.

- MV Steps are Too Short: The MV steps are not held for long enough to determine accurate gains.

- Lack of Fast MV Moves: The MV signal does not contain enough high frequency content to remove unwanted and unrealistic squiggles from the model curves that appeared due to fast large disturbances for which we did not have FFs.

1.1.6 Example of a Model Fitted to Poor Signal to Noise Ratio Data

Below is an example of a CV that suffers from a poor S/N ratio:



1.1.7 FIR Smoothing Algorithm

In the example shown above, the inverse response as well as the squiggles and cycles in the model curve may not be real. This is mostly due to large disturbances. As can be seen above, the FIR smoothing algorithm works fairly well to remove noise effects from the model and reduce the model uncertainty. It also helps to suppress the unwanted squiggles.

1.2 What is Subspace Identification?

Subspace Identification algorithms have evolved significantly over the past 10 years and have now matured to a point where it can provide a good platform for developing empirical dynamic models that exhibit more physical realism. Different from FIR ID method, the subspace methods fit process input and output data into a parametric state space model and directly identify an MIMO model with optimal model order and minimum prediction errors. The general state space model structure is shown below:

$$\mathbf{x}(k+1) = \mathbf{A}\,\mathbf{x}(k) + \mathbf{B}\,\mathbf{u}(k)$$
$$\mathbf{y}(k) = \mathbf{C}\,\mathbf{x}(k) + \mathbf{D}\,\mathbf{u}(k)$$

where **y** is the output vector containing the CVs
u is the input vector containing the MVs
x represent the internal dynamic states.

1.2.1 Differences between FIR and Subspace Identification

A difference between the FIR and State Space (SS) models is that state space model has an "intermediate layer" state vector **x** between the inputs and outputs. Though both FIR and SS model can map similar dynamical input-output relations, the SS model has more flexibility in model order selection and capability of sharing "common" states. Contrast to the FIR ID algorithm, the subspace ID algorithm searches and control the final model's order based on an optimal balance between the model size and the goodness of data fitting, whereas in FIR ID, model order is pre-specified based on TTSS and number of coefficients. The goodness of data fitting is the only objective of model ID. As a result, the Subspace ID returns smooth model yet with balanced model order. For large MIMO model, the effects of model order optimization will be significant as many internal states that offers similar dynamics will be eliminated. Therefore, the SS model structure and Subspace ID is more suitable to MIMO model identification.

For APC applications in process industry, Subspace Identification has been available in DMCplus Model since October 2000 (AMS 3.0 and later). It attacks the problem of fitting smooth realistic models from the opposite direction compared to the FIR method. The problem with all previous attempts at using parametric models in the process industry was the use of low order transfer functions to fit accurate models to relatively complicated dynamics. In the process industry, complicated dynamics are very common due to interacting PID controllers, recycles, heat integration and opposing process effects with different dead times, resulting in complicated inverse plus dead times models. All previous attempts at using parametric models have produced mixed results, and Subspace ID was the first parametric metric to perform well in these applications.

1.2.2 Summary of Subspace ID Capability

- **Use a MIMO Model Structure:** Subspace is a true MIMO identification algorithm, while the FIR algorithm treats process as a set of MISO models. Combining the right CVs in the same case actually helps noisy CVs to become more accurate as the clean CVs share the same states. The method first fits all the states using a MIMO approach, then uses a Least Squares approach to combine the states to form the final model. This approach helps to ensure accurate estimates of the underlying dynamics (states) especially if some MV/CV pairs have weak relationships.
- **Use Reliable Numerical Methods:** Use Singular Value Decomposition (SVD) to estimate states, which can guarantees numerical stability and accuracy. Optimal model order is automatically determined. This approach automatically groups together similar dynamics shared by different MVs and different CVs, leading to a smooth model with a minimal state count.
- **Automatic Model Order Determination:** The method auto-determines the optimal model-order, to capture balanced high and low frequency dynamics, without over-fitting.
- **Convert MIMO State Space Model to FIR:** A MIMO parametric model is identified and is then used to generate the step-response models, so the model curves are nearly always smooth.

- **Apply Data De-Trending to Suppress Drift Disturbances:** A data de-trending method is used to eliminate the effect of drift disturbances, increasing the accuracy of the model.
- **Efficient Slicing:** Subspace identification is very efficient in terms of slicing. If the maximum model order is 20, and TTSS=180, it will only use the first 20 samples after a slice to initialize, and much less data is lost due to slicing, compared to FIR that will lose 180 data points.

1.3 Basic Theory for Subspace Identification

There are many versions of subspace identification algorithms, but all are based on similar fundamental principles. (**Recommendation:** If you are not particularly interested in the mathematical details, just skip over this section.)

1.3.1 Innovation Form of State Space Model for Subspace ID

Consider a multi-input multi-output linear process model that can be represented by the following state space equations:

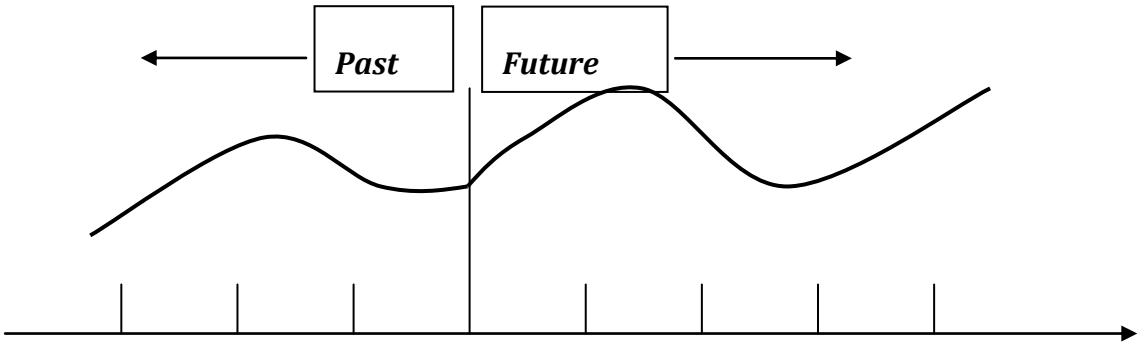
$$\begin{cases} \mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{K}\mathbf{e}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{K}\left[\mathbf{y}(t) - \mathbf{C}\mathbf{x}(t) - \mathbf{D}\mathbf{u}(t)\right] \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{e}(t) \end{cases} \quad (1)$$

Where the **u**(t) and **y**(t) denote the inputs and outputs; **x**(t) is the state vector and assumed a **n**-dimension vector; and **n** is the model order; **e**(t) denotes a zero-mean white noise; Matrices (**A, B, C, D)** and **K** formed process and noise model, they are unknown and to be estimated from process data.

1.3.2 Typical Subspace ID algorithm

Typically, the subspace ID algorithm uses multiple steps of **p**ast data from multiple inputs and outputs (independents and dependents) to predict multiple steps of **f**uture dependents (CVs) responses. This can be considered as a multiple variables regression problem with a sliding time-window. Assume the time-window's width to be **m**. First define the **p**ast and **f**uture data vectors as following:

$$\mathbf{u}_\mathbf{p} = \begin{bmatrix} u(t-m) \\ u(t-m+1) \\ \vdots \\ u(t-1) \end{bmatrix} \quad \mathbf{y}_\mathbf{p} = \begin{bmatrix} y(t-m) \\ y(t-m+1) \\ \vdots \\ y(t-1) \end{bmatrix} \quad \mathbf{u}_\mathbf{f} = \begin{bmatrix} u(t) \\ u(t+1) \\ \vdots \\ u(t+m-1) \end{bmatrix} \quad \mathbf{y}_\mathbf{f} = \begin{bmatrix} y(t) \\ y(t+1 \\ \vdots \\ y(t+m-1) \end{bmatrix}$$



Then based on state space model equation (1), **m**-step forward iterations on **y**(t) and **m**-step backward iterations on **x**(t) will generate the following extended equations:

$$\mathbf{y_f}(t) = \mathbf{L_f}\mathbf{x}(t) + \mathbf{H_f}\mathbf{u_f}(t) + \mathbf{W_f}\mathbf{e_f}(t) \qquad (2)$$

where

$$\mathbf{L_f} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA^{m-1}} \end{bmatrix} \quad \mathbf{H_f} = \begin{bmatrix} \mathbf{D} & 0 & \cdots & 0 \\ \mathbf{CB} & \mathbf{D} & \vdots & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{CA^{m-2}B} & \mathbf{CA^{m-3}B} & \mathbf{CB} & \mathbf{D} \end{bmatrix} \quad \mathbf{W_f} = \begin{bmatrix} \mathbf{I} & 0 & \cdots & 0 \\ \mathbf{CK} & \mathbf{I} & \vdots & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{CA^{m-2}K} & \mathbf{CA^{m-3}K} & \mathbf{CK} & \mathbf{I} \end{bmatrix}$$

Furthermore, the process equation (1) can also be rewritten as follows:

$$\begin{cases} \mathbf{x}(t+1) = \tilde{\mathbf{A}}\mathbf{x}(t) + \tilde{\mathbf{B}}\mathbf{u}(t) + \mathbf{K}\mathbf{y}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{e}(t) \end{cases} \qquad (1a)$$

With the definition of $\tilde{\mathbf{A}} = (\mathbf{A} - \mathbf{KC})$ and $\tilde{\mathbf{B}} = (\mathbf{B} - \mathbf{KD})$.

If iterating equation (1a) m steps backward, it can be shown that the state vector **x**(t) is only a function of past inputs, past outputs and its own past state values:

$$\mathbf{x}(t) = \sum_{i=0}^{m-1} \tilde{\mathbf{A}}^i \left[\mathbf{K}\mathbf{y}(t-i-1) \quad \tilde{\mathbf{B}}\mathbf{u}(t-i-1) \right] + \tilde{\mathbf{A}}^{\mathbf{m}}\mathbf{x}(t-m) \qquad (3)$$

Assuming $\tilde{\mathbf{A}}$ is stable, when m is large enough, $\tilde{\mathbf{A}}^m \to 0$; this implies that the current state **x**(t) is a linear combination of the past inputs and outputs. Then from equation (2) we can further know that the future output $\mathbf{y_f}(t)$ consists of two parts, a linear combination from past inputs and outputs data (through state **x**(t)) and a contribution from future inputs. $\mathbf{L_f}$ **x**(t) forms a predictable subspace (part of future output space) from past data. If we are able to calculate the state estimate

$$\hat{\mathbf{x}}(t) = \Gamma\, \mathbf{p}(t) \qquad (4)$$

Where Γ is a matrix of unknown coefficients and **p**(t) a vector containing only past data:

$$\mathbf{p}(t) = \left[\mathbf{y}^{\mathbf{T}}(t-1) \quad \mathbf{y}^{\mathbf{T}}(t-2) \quad \cdots \quad \mathbf{y}^{\mathbf{T}}(t-m) \quad \mathbf{u}^{\mathbf{T}}(t-1) \quad \mathbf{u}^{\mathbf{T}}(t-2) \quad \cdots \quad \mathbf{u}^{\mathbf{T}}(t-m)\right]^{T}$$

Then with known {**y**(t)**, u**(t)} and estimated {**x**(t)} available, the state space model (A,B,C,D) can be estimated from equation(1) with Least-Squares regression. There are several ways to estimate the predictable subspace, which lead to slightly different subspace identification algorithms. All the subspace identification algorithms follow the following steps:

Step 1: Estimate the predictable subspace $\mathbf{L_f X(t)}$ from data:

Alternative 1: Assuming that $\mathbf{u}_f$ is completely independent of $\mathbf{L_f}\ \mathbf{x}(t)$ in Eq.(2), by regressing out $\mathbf{u}_f$ from $\mathbf{y}_f$ (MOSEP):

$$\mathbf{z}(t) \overset{\Delta}{=} (\hat{\mathbf{L}}\hat{\mathbf{x}}) = \mathbf{y_f}\left(\mathbf{I} - \mathbf{u_f}^{T}(\mathbf{u_f}\mathbf{u_f}^{T})^{-1}\mathbf{u_f}\right)$$

Alternative 2: By estimating $\mathbf{H_f}$ and then removing $\mathbf{H_f}\,\mathbf{u_f}$ from $\mathbf{y_f}$ (CVA, N4SID):

$$\mathbf{z}(t) \overset{\Delta}{=} (\hat{\mathbf{L}}\hat{\mathbf{x}}) = \mathbf{y_f} - \hat{\mathbf{H}}_\mathbf{f}\mathbf{u_f}$$

Step 2: Extract state variables x(t) from estimated subspace:

Apply Singular Value Decomposition (SVD) on the estimated subspace $\left(\hat{\mathbf{L}}\hat{\mathbf{x}}\right)$, which maximizes the cross-correlation between the past measurements and their causal future outputs. The best predictable variables (state variables) are found in a descending order.

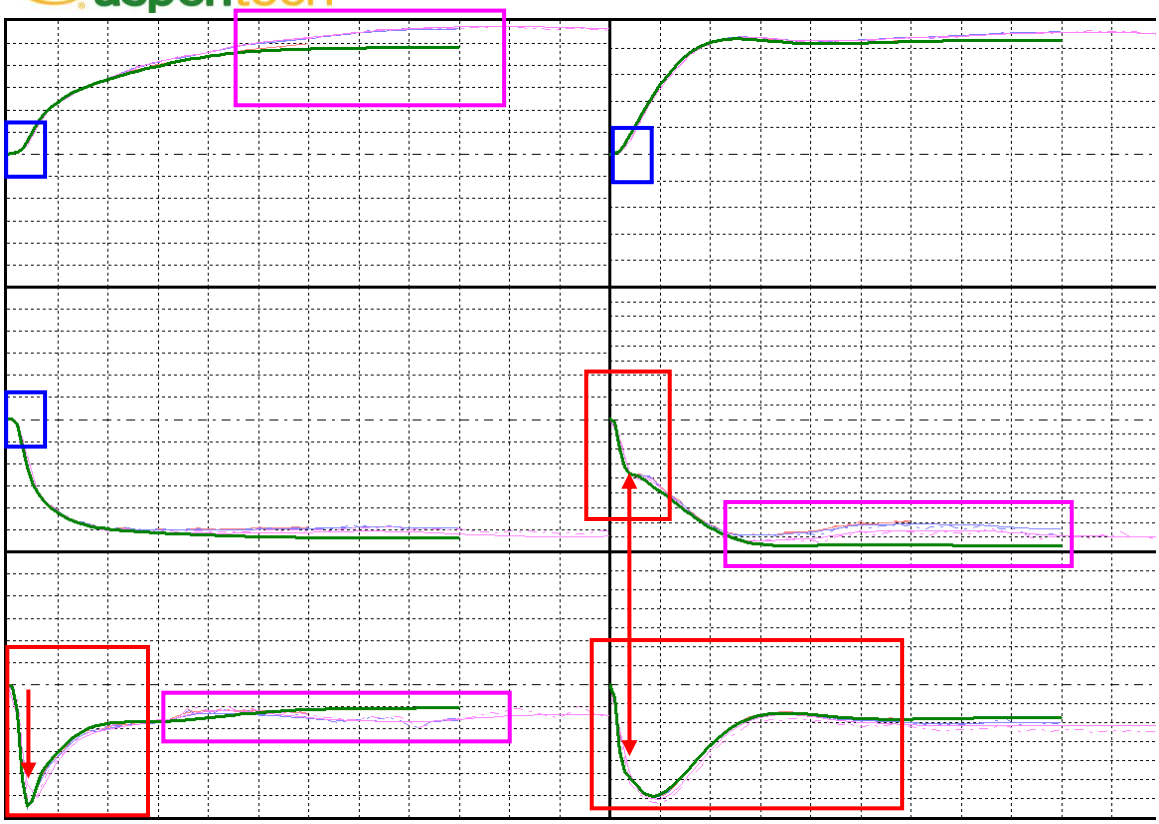
Step 3: Determine state space order m, either by using truncation of singular values (MOSEP, N4SID), or by using Akaike Information Criterion (AIC) to determine a balanced model order (CVA method). The implementation in DMCplus Model uses the CVA method.

Step 4: Estimate **A, B, C, D** state space matrices from equation (1) and data **u**(k), **y**(k) and **x**(k) by Least Squares regression.

Step 5: Generate step-response curve and FIR model coefficients.

1.4 Case Study – De-Isohexanizer Column

Column Tray Temperatures: This is an example showing Subspace ID accurately fitting complicated tray temperature models for a de-isohexanizer column:



Legend: Shared Dead Time States are shown in the blue boxes; Fast (Shared) Initial Dynamic is shown in red boxes; Differences between FIR and Subspace Gains are shown in pink boxes.

Subspace Models are Almost Always Smooth: The bold green trace is the subspace model using standard default parameters (using de-trending) with a maximum model order of 20, while all the other plots are FIR models. The FIR model matrix (all the other model curves) contains several thousand FIR parameters, while the subspace model contains only 7 states and a total of (7x7 + Ind x7 + Dep x7) parameters to fit all of the MV/CV relationships in the case. It is clear that the subspace model is less "noisy" (cleaner) than the FIR models and display more process realism.

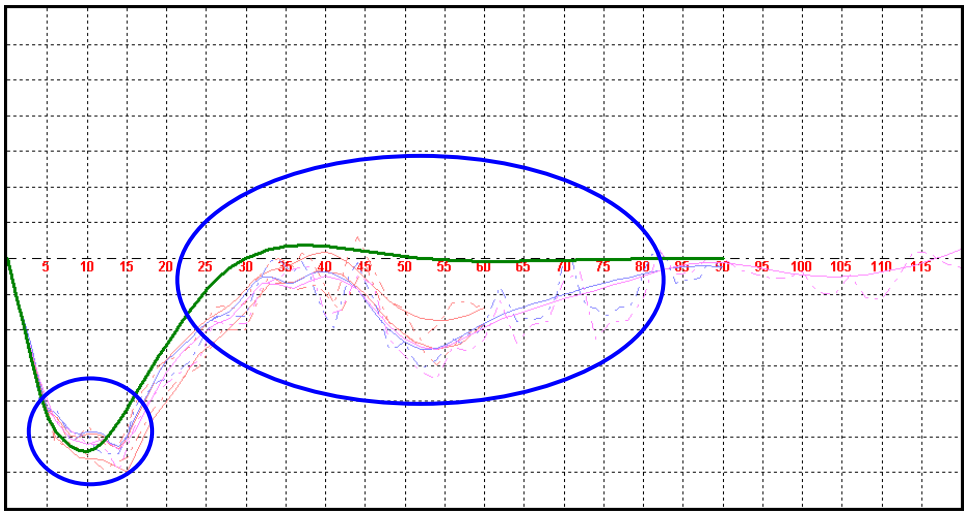
Dynamic Co-linearity: Also note the dynamic interaction between the various tray temperatures indicated by the arrows. This helps to make the future CV predictions and the resulting MV control moves more accurate during closed loop control when all the MVs are moving simultaneously to generate decoupling moves, i.e., it deals with the dynamic co-linearity and dynamic robustness problem as well. This is especially powerful if the step test data set was generated using aggressive pulsing to generate high frequency data content, stepping multiple MVs at the same time. Aspen SmartStep® is capable of stepping dozens of MVs at the same time, using the paired MV feature or multi-test feature on the last day or two of the step test.

Fit High Order Dynamic Effects: In this example, the subspace models as well as the FIR models have accurately captured these high order effects shown in the boxes. Both

methods also dealt correctly with dead time. Some of the credit in this case goes to Aspen SmartStep® for generating exceptionally high quality data.

Offset between FIR and Subspace Model Does NOT Mean Subspace is Wrong! Some differences between the subspace and FIR model are also apparent (top left). This does not mean that the subspace models are less accurate than the FIR models simply because they are different. It is natural to think that FIR sets the benchmark in terms of accuracy, since we have had so much success with this method over such a long period of time. In reality, this is not necessarily the case, as the FIR method is a simple least squares method and is sensitive to feedback correlation. Even if all the FIR model curves are right on top of each other (usually an indication of an accurate model), it could still be a few percent off due to the presence of MV cross correlation or feedback correlation. Subspace ID has special ways of dealing with these problems, and in general is less sensitive to these problems, leading to more accurate models. A tightly grouped set of FIR model curves is an indication of low variance error, but it is not an indication of low bias error. Bias errors often result due to feedback correlation.

The Effect of Low Signal to Noise Ratio: An example is shown below where we know the gain between this specific MV and the closed level CV is zero. In this case, the signal to noise ratio is quite low, due to the small impact of the disturbance on the closed level, and we expect to fit a relatively poor model. The FIR models curves (the colors in the background) clearly indicate less than perfect models, as can be expected in this case:



The Effect of MV Periodicity and Disturbances on FIR models: The bold green trace above is the subspace model showing the response of a closed level controller against an upstream disturbance, and the gain should be exactly zero since the PID controller is in AUTO mode. The second dip in the FIR models after t=40 is due to unfortunate periodicity in the MV steps, and perhaps some unknown disturbances. It is of course not

surprising that a low order state space model will be far smoother than a high order FIR model, especially if the S/N ratio is poor. What is most impressive is how much the sub-space result agrees with our engineering expectation of how a closed level PV will respond to an upstream Feed Forward (FF) disturbance. The sub-space model exhibits a higher degree of process realism than the noisy FIR response.

2 Getting Familiar with the DMCplus Model User Interface and Subspace Identification Parameters

2.1 Understand the parameters and users' options on subspace identification

The Subspace Identification offers a list of new features and more flexibilities and capabilities to model complex MIMO process. On the other hand, it has also more parameters and options available for users to choose for their specific cases. In an effort to maximize the benefits of Subspace ID while facilitating the use of this relatively new ID technology, a two-level of parameters and options have been provided in DMCplus Model software. One level is for normal user and general cases, it has been made similar to the traditional FIR ID cases. In this level, the user is supposed to over type only the time to steady state (TTSS) if the default value (90 min) is not appropriate for the case. All other parameters/options have been assigned a set of default values which were constantly modified according to internal testing and users' feedback over the last ten years of application. In most cases, the default settings of subspace ID will produce reasonably good models. In only some special cases, the parameters and options may need to change in order to get improved models. For this purpose, a second level of tunings is provided for experienced users. There are many parameters and options are accessible in this level. They allow user to compare various resultant models from different settings and compensate for their imperfect test data, such as short test data, oscillating CVs, large drifting or high frequency disturbances, feedback correlations, etc. In the following sections, both default parameters and how to change their values for better models will be described.

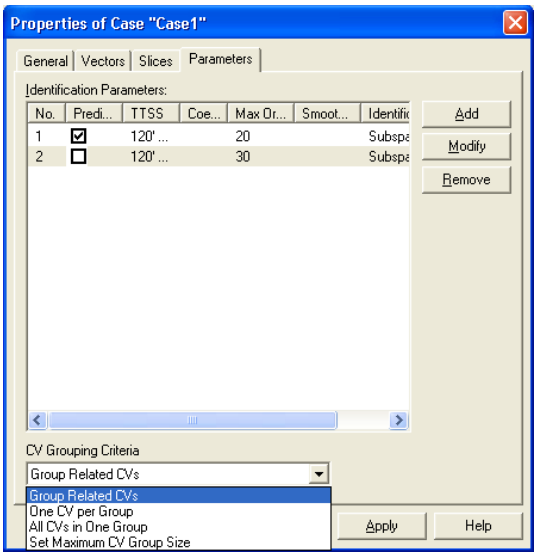
2.1.1 CVs Auto-grouping in Subspace Identification

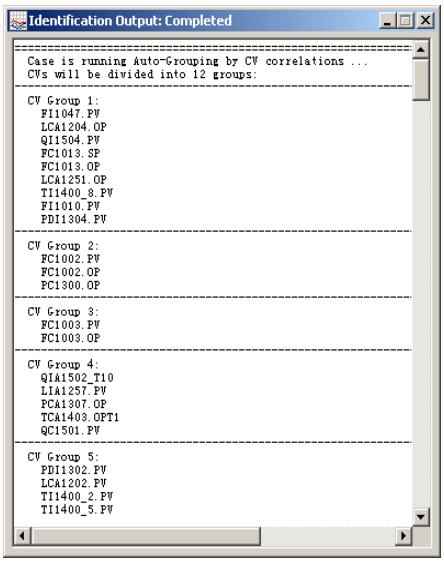
Initially, the Subspace ID algorithms in earlier releases of DMCplus software was set as a default to fit ONE Integrated MIMO state space model to all your MVs and CVs in a Case at once, with an intention to obtain the maximum benefits from a MIMO model identification. In practice, however, if you put dynamically related CVs in the same case, results will be more accurate as the model will share dynamics between CVs. If you put unrelated CVs in the same case, the larger size of the MIMO model slowing down the case run significantly while the benefits were limited compared to running multiple smaller cases.

Since the release of DMCplus 2006, a CV Auto-grouping algorithm was implemented to help users group CVs automatically based on an internal CV correlation analysis. The default option on CV grouping is "Group Related CVs". By this option, All CVs in a case

will be analyzed with a cross-correlation calculation, and then they are put into different CV groups so that the CVs in a group show high cross-correlations each other and a default maximum number of 10 CVs will be used for each CV group. Later, the subspace ID algorithm will process one CV group as an internal sub-case at a time to perform a MIMO model identification. Once all CV groups are processed, the algorithm will assemble all sub-case MIMO models into one large MIMO model for the original case. In this way, the subspace identification algorithm optimizes case runs by using CV grouping. By keeping related CVs together the underlying states will be identified with more certainty. By removing unrelated CVs the computation time per subgroup is decreased. CV grouping allows the creation of larger cases and can also help provide insight into the relationships between variables.

There are also other alternate options on CV grouping allowing user to choose as shown below:





- Group Related CVs (Auto-grouping, Default)
- One CV per Group (Forced MISO ID)
- All CVs in One Group (Forced large MIMO ID)
- Set Maximum CV Group Size (MIMO ID with certain number CVs per group)

In an earlier release, it was suggested that user should manually put all PVs of closed TICs, PICs, LICs etc. that affect the dynamics of the process in a cases while limiting the sum of MVs and CVs is not too large (e.g. > 20). Similarly, calculated delta temperatures for heat exchanger (preheat) trays also helps to clean up models where there is a lot of heat integration. For process units with large recycles, always include the recycle flow in all the cases. With the newer release, it is still recommended to put correlated PVs into one case even if they are not required for control.

By default, the newer release of the software will conduct an internal correlation analysis and group all the CVs showing high correlations into one sub-case for subspace ID

calculation. The default Maximum number of CVs in a sub-case is set to 10. For example, if you have a case with a total of 24 CVs, the "CV Auto-grouping" algorithm will calculate cross-correlations for all the CVs and form the first sub-case with a maximum of 10 CVs if high correlations for CV pairs are found and the top 5 pairs will be put into the group. Then the second 10 CVs according to their correlation rank will be put into another sub-case. The rest CVs with less or no correlations will be put into the last sub-case. Each sub-case is then processed as a smaller MIMO ID problem and the subspace ID result from each sub-case will be re-assembled together to form a model matching to the original large case.

Three options allow users to change the above default behavior:

(1) Select "One CV per Group", this will force the case to run as multiple MISO model identifications (i.e. for each CV, a MISO sub-case is created and run). With this option, the users are able to compare the MISO subspace ID with the FIR ID in a similar internal structure setup.

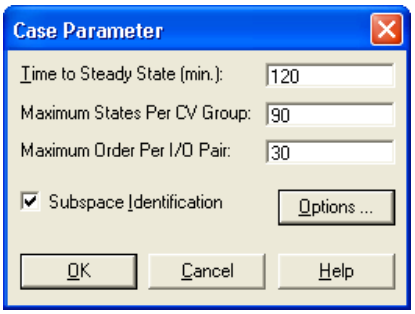
(2) Select "All CVs in One Group", it will set the original case as one large MIMO identification problem to solve and it will allow user to make comparisons to the results by using other structure. The only possible issue in practice is a significant slow-down of the subspace case run if the case is very large.

(3) Choose "Set Maximum CV Group Size" and input a number between 1 to total number of CVs. This option gives user a flexibility to modify the default Maximum CVs of 10 per group. When the user input 1, it will run multiple MISO cases (i.e. option(1)); if input "Total number of CVs", it turns to option (2); input any other valid number, for example, "M", the CV auto-grouping algorithm will still run and the only difference is that each CV group sub-case will contain maximum M CVs.

The implementation of "CV auto-grouping" and a significantly improved MIMO subspace algorithm in speed have well addressed a previous "speed issue with large MIMO cases" in application of the subspace ID. In addition, the new release allow users to gain/validate their knowledge about the CVs' correlations in their process, compare models with different setup, and re-tune the default parameters when necessary.

2.1.2 Case Parameters for Subspace Identification

There are three primary parameters with default values in the "Case Parameters" dialog when user creates a new subspace ID trial in a case as shown below:

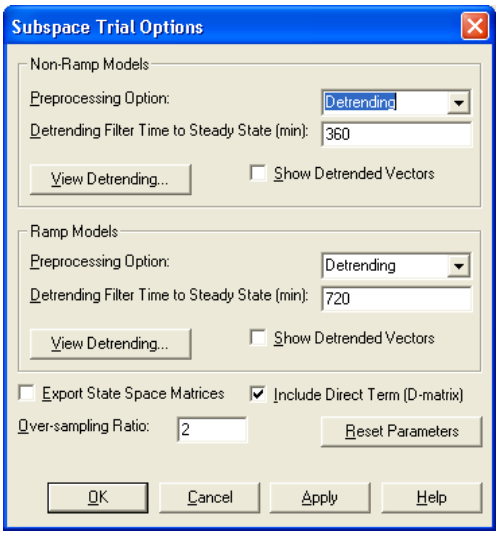


- **Time to Steady State (min)**
- **Maximum States per CV Group**
- **Maximum Order Per I/O Pair**

Time to Steady State Similar to the FIR identification, a **Time to Steady State** (i.e. TTSS) should be chosen for an appropriate FIR curve length in order to cover the complete dynamics of a model. In traditional practice, users often use multiple TTSS for a FIR model to validate the convergence of a FIR model. If the ID results show significant gain differences with multiple TTSS values, it means the models with high uncertainty, either the data is not large enough, the signal to noise (S/N) ratio is low, or the true relationship between an Independent and a Dependent in the process is weak.

For subspace ID, the TTSS values will also affect the model results, but in a different way. Theoretically, the TTSS values have no direct impacts on the subspace ID, because the TTSS is not a dependent parameter in subspace ID and it only determines how many FIR model (curve) coefficients will be generated from a state-space model identified. However, when user select (inexplicitly by default) "Detrending" as data preprocessing option, the "Detrending" filter will be calculated upon the TTSS values. Therefore, it is strongly recommended "To Use Realistic TTSS Values ONLY". For example, if you expect TTSS=120 minutes, then pick only 90, 120 and 180 minutes for subspace ID. Don't use very low TTSS values in subspace ID, as the de-trending filter will affect the models in a bad way. Very long TTSS means the de-trending filter is effectively turned off, but the excessive over-sampling will introduce phase lag.

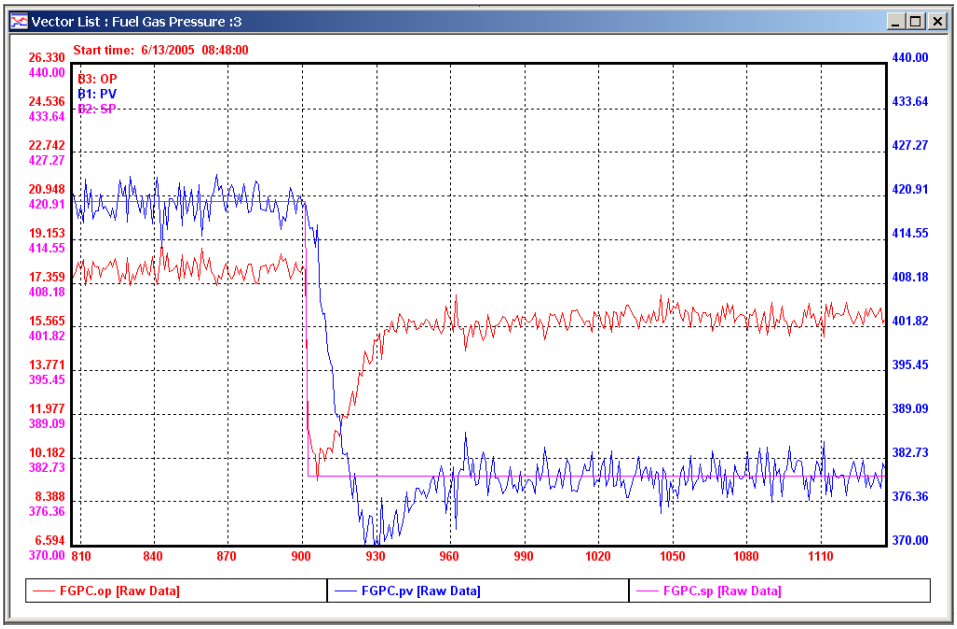
There is another way to eliminate the negative impacts of different "detrending effects" by modifying the default detrending filter. By clicking on "Options" button from the "Case Parameter" dialog, a new dialog window named "Subspace Trial Options" will allow user to change the options and parameter values for each subspace trial. The UI below shows the dialog of Subspace Trial Options:

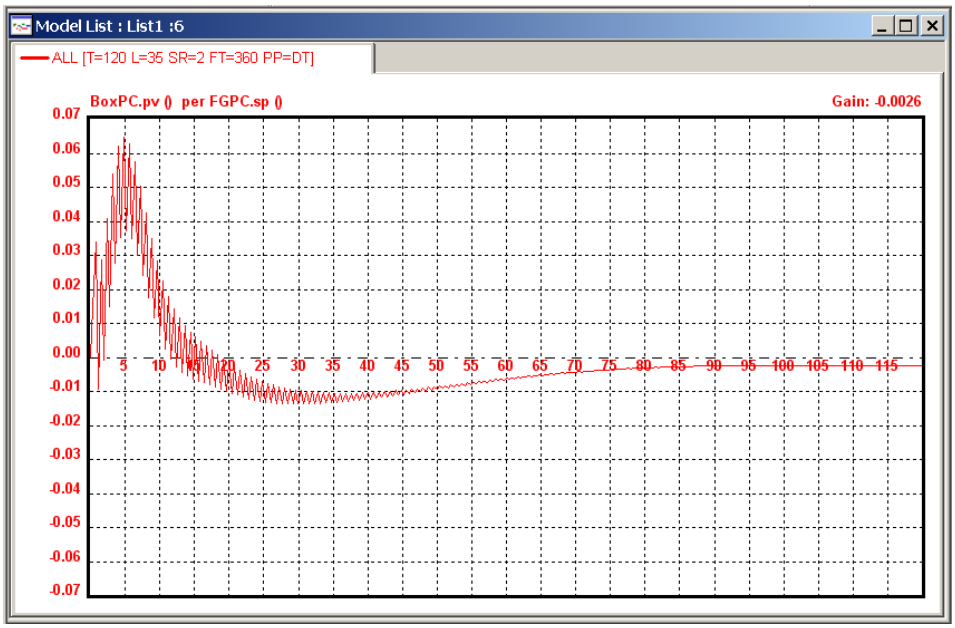


In this dialog, manually type in an identical "Detrending Filter TTSS" value for every trials with different TTSS values, the subspace ID results will be independent of the TTSS! Alternatively, if you select "Differencing" or "Zero-Mean" options for data preprocessing, the TTSS will have no any impacts on the subspace ID results either.

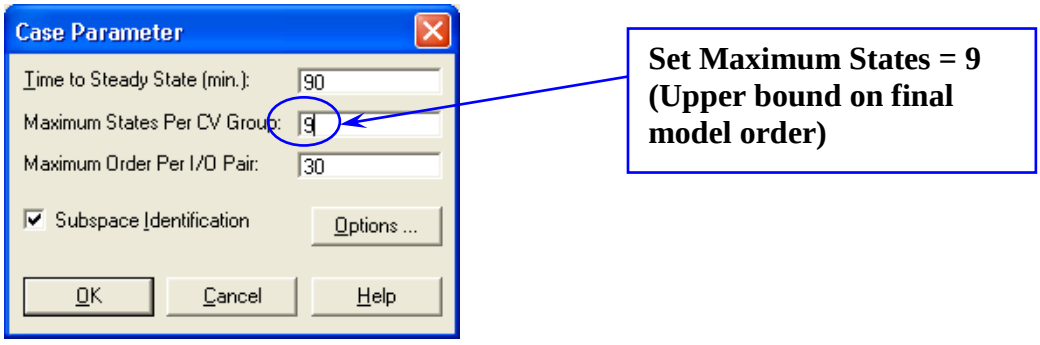
Maximum States per CV Group The second primary parameter for subspace ID is **Maximum States per CV Group** (or **Model Order Cutoff** in earlier release), which is an upper bound of the model order (i.e. the maximum number of states allowed) for each CV group sub-case. For each internal sub-case, subspace ID algorithm will search for an "Optimal" model order (equivalently, the total number of states) based on the so called Akaike's Information Criterion (AIC). The **Maximum States per CV Group** allow user to set a model order search range or force the subspace algorithm to fit a low-order model with truncation. Its default value of 90 is often sufficient for most cases. Therefore, it is recommended not to modify this default value unless you need to reduce the model order on special purpose like the example shown below:

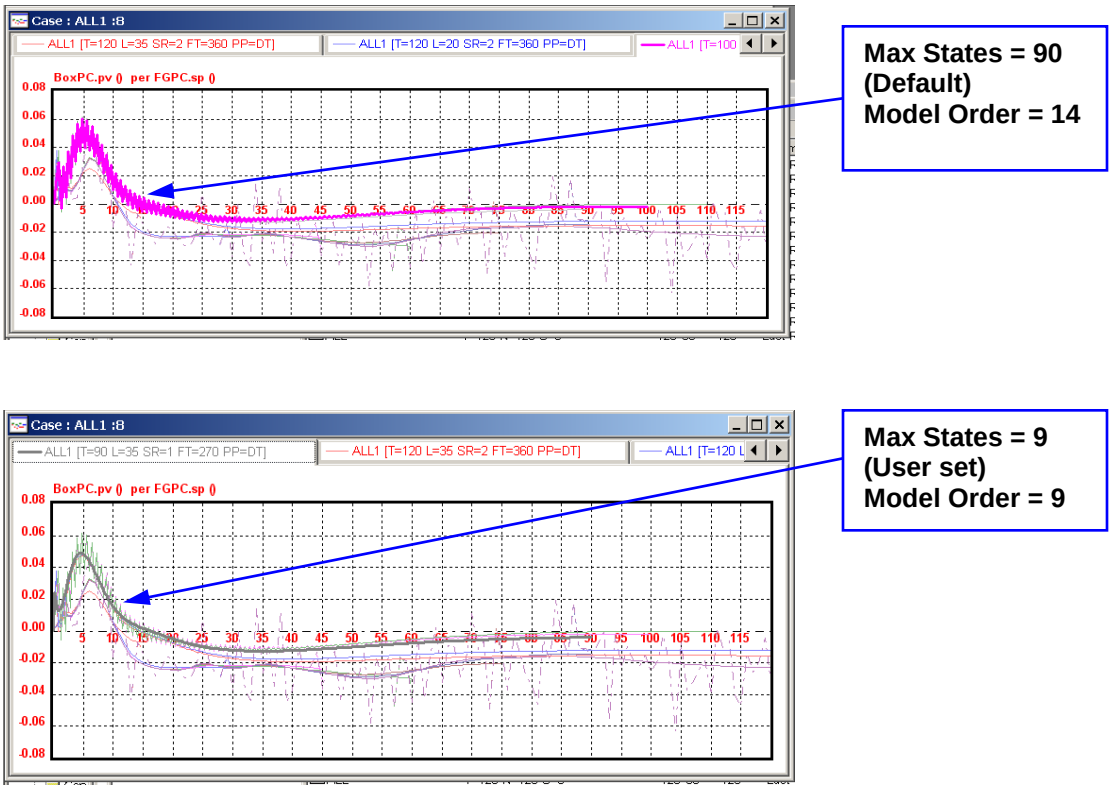
A PID loop of Fuel Gas Pressure control data shown here indicate obvious valve oscillations, i.e. the OP and PV run into infinitive cycling. For such a dataset, a subspace case run with all default parameter values generates a model with high frequency dynamics, i.e. the subspace model catches both slow and fast dynamics.



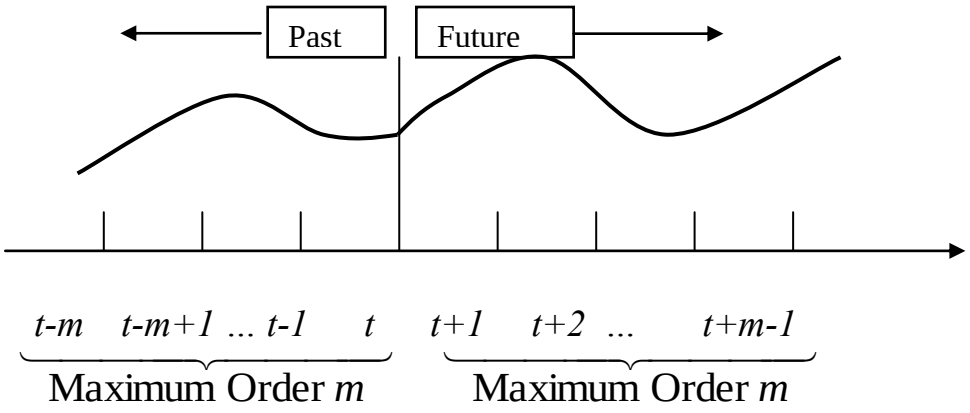


In this case the ID result is not what we like. The run-time message indicates that the final **model order = 14**. Based on experience we are aware that usually the high-frequency dynamic behavior of a model is caused by a high order model and a model reduction can help to damp the high frequency noise. For this case, we modified the parameter value of the **Maximum States per CV Group** from a default value of 90 down to 13, 12, 11, 10, 9,… and we got a satisfactory model when we forced the maximum states = 9. Compare the two models (14th order and 9th order), the reduced order model will be a good model for the MPC controller.





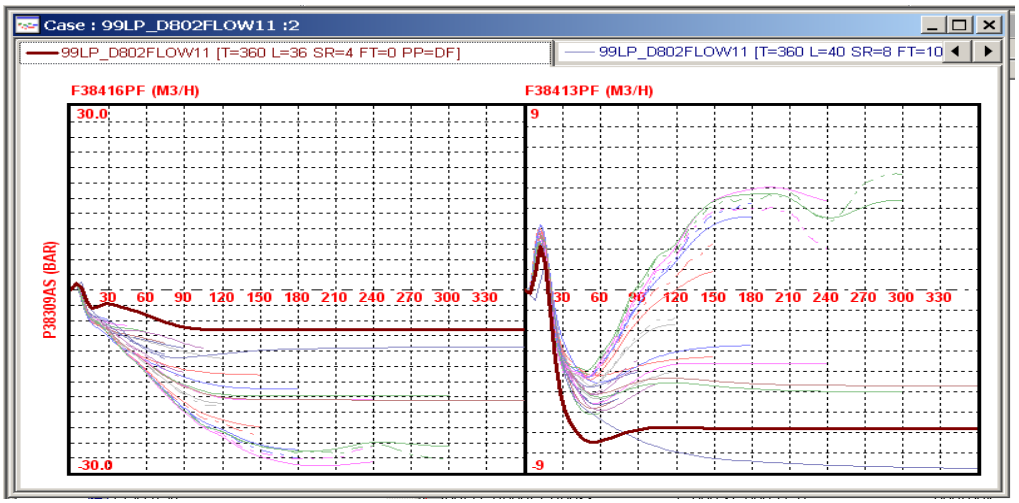
Maximum Order per I/O Pair The third primary parameter is the length of time horizon used for regression in subspace identification. It is actually the number of data samples that the algorithm will take at each sampling time, both for forward in the future and backward in past. This parameter represents the width of a time window which determines how long the past data series will be regressed at a time moment to form a predictor to predict the future outputs over the same length of a time horizon. The larger the Maximum Order per I/O Pair, the higher order model will be identified in order to cover longer (complex) dynamics. The computation load will also be heavier.



Maximum Order per I/O Pair Affects the Cost Function Calculation: The cost function of subspace ID is constructed by comparing the predicted CV values with the observed CV values, at every time step, *m* steps into the future, given *m* steps of data from the past. The cost function therefore emphasizes the LONG TERM prediction accuracy of the model, not just the one-step-ahead prediction accuracy. This is an important difference with traditional ARX identification methods. If very different Maximum Model Orders are used in a subspace case, then even if the same final model order is selected, the calculated models are bound to be different, because the model's prediction emphasis are shifted with the length of the time window. This is a little non-intuitive at first if you are used to FIR, but it is expected behavior. This is similar to the way that FIR models are expected to be different when you select different TTSS. FIR is affected by truncation error (your estimate of TTSS). Subspace is immune to truncation error but is affected by the user's choice of model dimension (maximum order) and data pre-processing options (the de-trending filter, which is dependent on TTSS, and the over-sampling ratio).

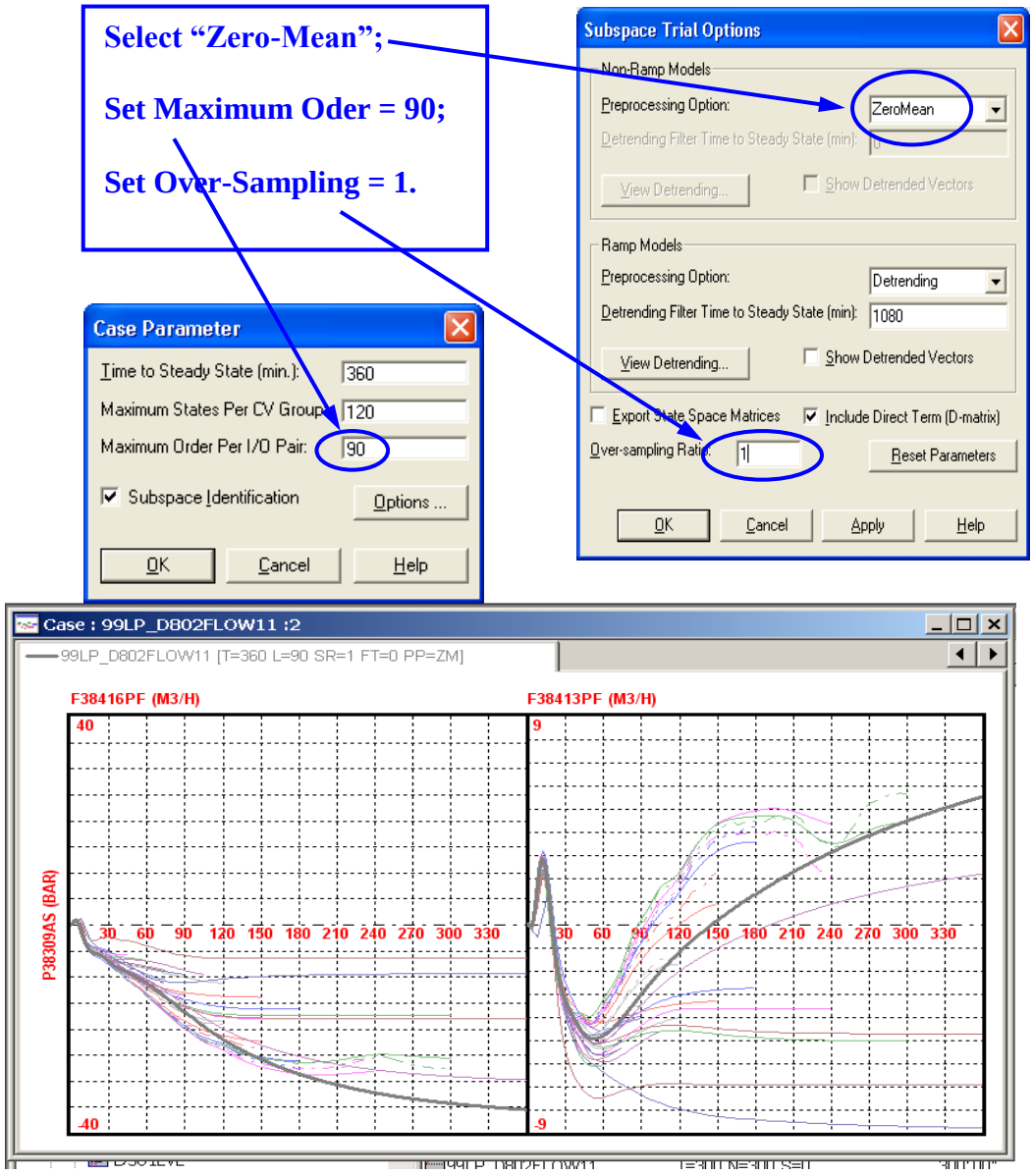
The default Maximum Model Order (i.e. the length of time window) is set at 30, it works fine for most seen processes. A very high Maximum Model Order will slow down the ID calculation, especially for large MIMO case. In some cases, however, a larger Maximum Model Order is necessary and beneficial in order to catch complex dynamics over a long TTSS time horizon. Otherwise, it is often seen that a subspace model curve is "flat-out" too early. The following example shows such a case where a much higher Maximum Model Order (=90) than the default (=30) will help.

A process with recycle has a long TTSS = 360 min, if the case is run with default parameter settings (Maximum Order = 36; Preprocessing = "detrending"; Over-sampling Ratio = 4), the subspace ID generated a model shown below:



Observation: The subspace model catches the process dynamics within the first 30 minutes (short-term) reasonably, but it "flat-out" after 90 minutes. This "matches" the default "time window" of 30 (Maximum Model Order).

After expanding the Maximum Model Order up to 90, and selecting the pre-processing option "Zero-Mean" to emphasize the "long-term" slow dynamics, re-run the case and the subspace ID resulted in a different model as shown below. It improves the "long-term" slow dynamics significantly.



Options on Data Pre-Processing There are several Preprocessing Options for subspace identification. Each option has its own feature and effect on the final model. An optimal preprocessing can enhance the Signal to Noise (S/N) ratio by filtering noise and

disturbances out from process measurements. The default option is set to "Detrending" which works well for most cases. User is allowed to switch it to other options from the UI. Available options on data preprocessing are listed below:

- Detrending
- Differencing
- Zero-Mean
- Double Diff

Detrending: In practice, process test data often contain low frequency drift that is induced by unknown disturbances. These low frequency disturbances will create negative impacts on identification and has to be eliminated. The option "Detrending" is designed to do the job.

Assuming a measured signal can be divided into a process signal and a trend signal:

$$y = y_{Process} + d_{Trend}$$

The pre-process of data with detrending will estimate the trend signal and remove it from the measurements. There are many different trend signals, such as Mean, Linear or random disturbance driven filtered signals, etc. and the trend type is normally unknown. To remove the unknown trend, a low-pass trend filter can be used to estimate the trend signal. Further, to avoid zero-phase distortion, the data signal will be filtered causal and anti-causal as show below:

$$\hat{d}_{Trend}(t) = F_{Trend}(q)\, F_{Trend}(q^{-1})\, y(t)$$

Then the detrending of data is performed by simply removing the estimated trend signal from process data as follows:

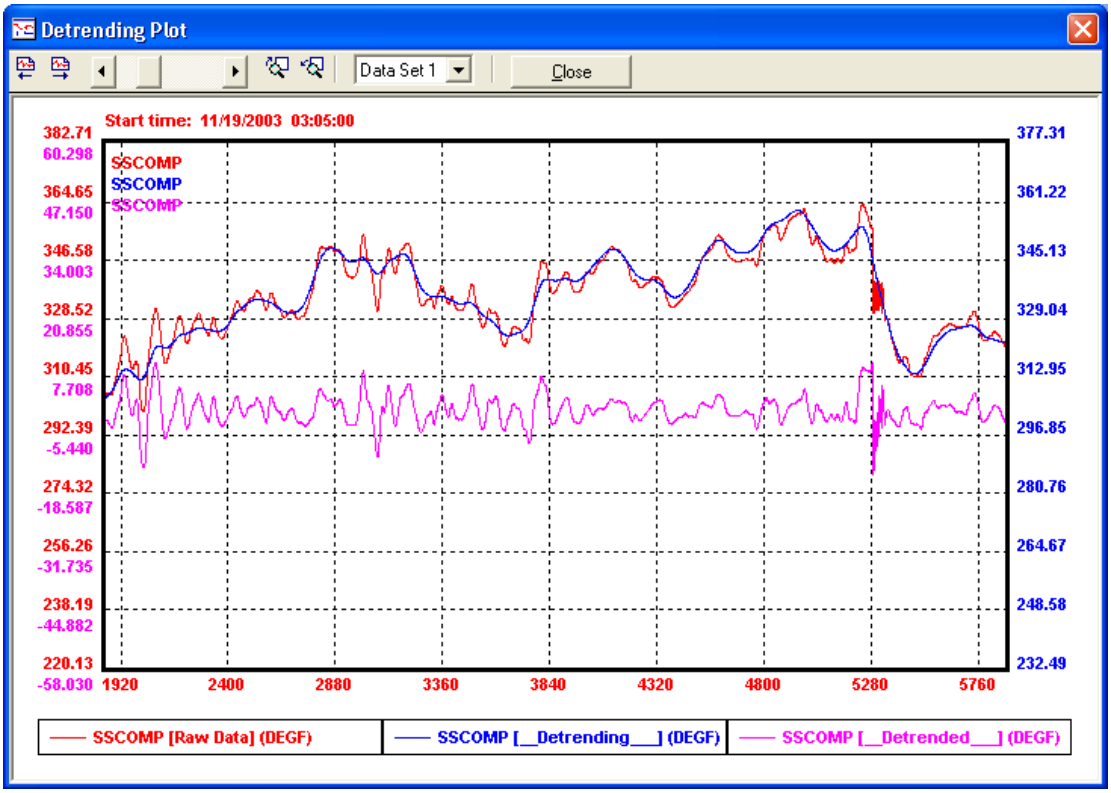
$$y(t) = y_{Measurement}(t) - \hat{d}_{Trend}(t)$$

By default, the time-constant of detrending filter is calculated based on the case's TTSS:

$$T_{Detrending} = 3 * TTSS \quad \text{for non-ramp sub-case}$$
$$T_{Detrending} = 6 * TTSS \quad \text{for ramp sub-case}$$

This default calculation is intended to ensure that all the interested process dynamics will not be removed by detrending while all slower drifts including constant offset will be filtered out. It should be aware that **A Too small TTSS value for a case may lead to an "over-detrending"** that may remove useful dynamic information from the data and cause inaccurate models. If users do want to see models with short TTSS, they can manually change the detrending filter TTSS from UI to avoid the negative impacts of a too-short TTSS.

The following picture shows an example of a CV's detrending:



Where the raw data, trend signal and detrended data are plotted with red, blue and pink colors respectively.

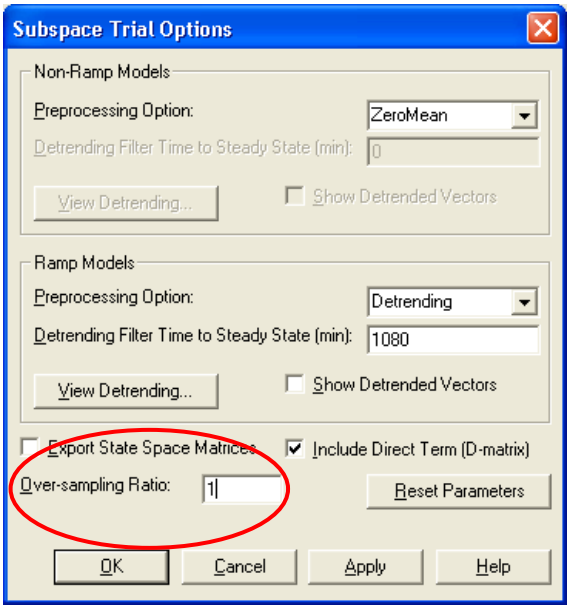
Differencing: Similar to detrending, differencing is also intend to remove slow drifts from the process data. The FIR model ID has been using differencing as a data pre-processing strategy since the DMC was first built. One of the reasons for using differencing is that it fits the incremental formula of a FIR model perfectly. In term of functions, it can remove all slow disturbances as a high-pass filter provided that the signal to noise (S/N) ratio of the test data is high enough. Given the traditional way to do a plant test by moving MVs once a time and holding that MV long enough, the S/N ratio is often good enough, differencing works well in most cases. On the other hand, differencing has two major shortcomings: For process data with low S/N ratio, such as weak CVs, or small perturbations, differencing may enhance high frequency noise and may lead to large gain errors; for ramp CVs, traditionally two-time differencing will be applied to the ramp data, and the ramp rate model can be off far away from its real value. Therefore, for better model gains and less stringent requirements on S/N the Detrending is recommended in place of differencing.

Zero Mean: This option is often used as a reference for comparison between other pre-processing options. In term of functions, it simply removes only an offset from raw data.

It should be aware that there will be multiple offsets if the data consists of several slices. The Zero-Mean operation will be performed on each data slices.

Double Difference: Double Difference was originally designed for Ramp and Pseudo-Ramp CVs in FIR ID operation. For subspace ID, it is no longer the only way to pre-process Ramp CVs. Instead, a new strategy has been developed and implemented behind the UI. Either "Differencing" or "Double Diff" as an option was selected, a single Differencing will be first applied to a Ramp or Pseudo-Ramp CV. Then Detrending (Zero-Mean if the slice is short) or Difference are applied to both raw MVs and differenced Ramp/Pseudo-Ramp CVs. The subspace ID will internally identify a stable (Non-Ramp) MIMO model and add a pure (clean) integrator back as a post-step after model identification. Practice has demonstrated that the new strategy with option "Detrending" for Ramp CVs results in better Ramp models with improved gain accuracy. In addition, all Ramp CVs or Pseudo-Ramp CVs are always put into one or more Ramp CV groups by the Auto-grouping algorithm in order to separate Ramp CVs from Non-Ramp CVs. In such a way, unnecessary crossover-disturbances between Ramp and Non-Ramp CVs can be effectively avoided and the Ramp CV Models can be further enhanced.

Over-sampling Ratio: This is a parameter to control the model sampling frequency as shown below:

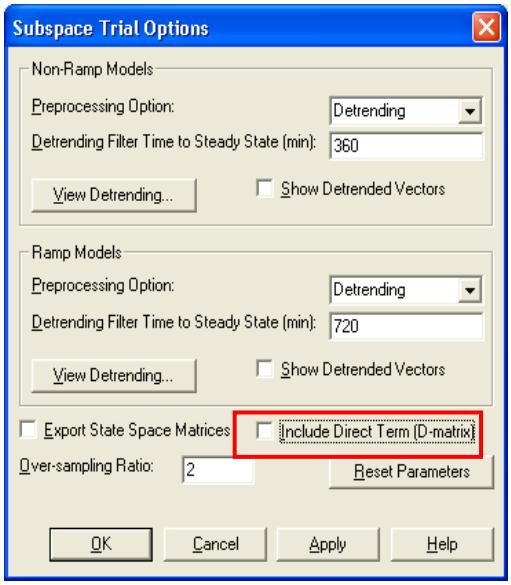


In the case where the process data are collected in a high frequency sampling and the final model for on-line control will be run in a slower control sampling frequency, a value of greater than one can be used to match the model's sampling rate with the controller's operation. Typical examples are product quality controllers with slower GC measurements. Another way of using the over-sampling ratio is to help the identification by adjusting its value into an optimal range according to the length of process's TTSS. For this purpose, the default values of over-sampling have been set up as following:

$$OR = \begin{cases} 1 & \text{if } TTSS < 120 \text{ min} \\ 2 & \text{if } 120 \text{ min} \le \text{TTSS} < 180 \text{ min} \\ 3 & \text{if } 180 \text{ min} \le TTSS < 240 \text{ min} \\ 4 & \text{if } TTSS \ge 240 \text{ min} \end{cases}$$

It needs be aware that a larger over-sampling ratio can results more accurate model gains, but may also introduce dynamical delays.

Include Direct-Term (D-Matrix) The subspace ID algorithm can identify a model with immediate output response to one or more inputs. This relationship is represented by the D matrix in equation (1). In most PID loops, a model of SP to OP will show such a direct response often called "initial kick of OP". If a FIR model ID is applied, this "initial kick" will not be shown in the model until later after the first FIR coefficient. In the DMCplus Model software, the Direct-Term is set on as a default option and it will help improve model's gain for most cases. If the S/N ratio is poor and the resulted models show unrealistic "initial kicks", it is suggested to try turning it off by uncheck the box as shown below:



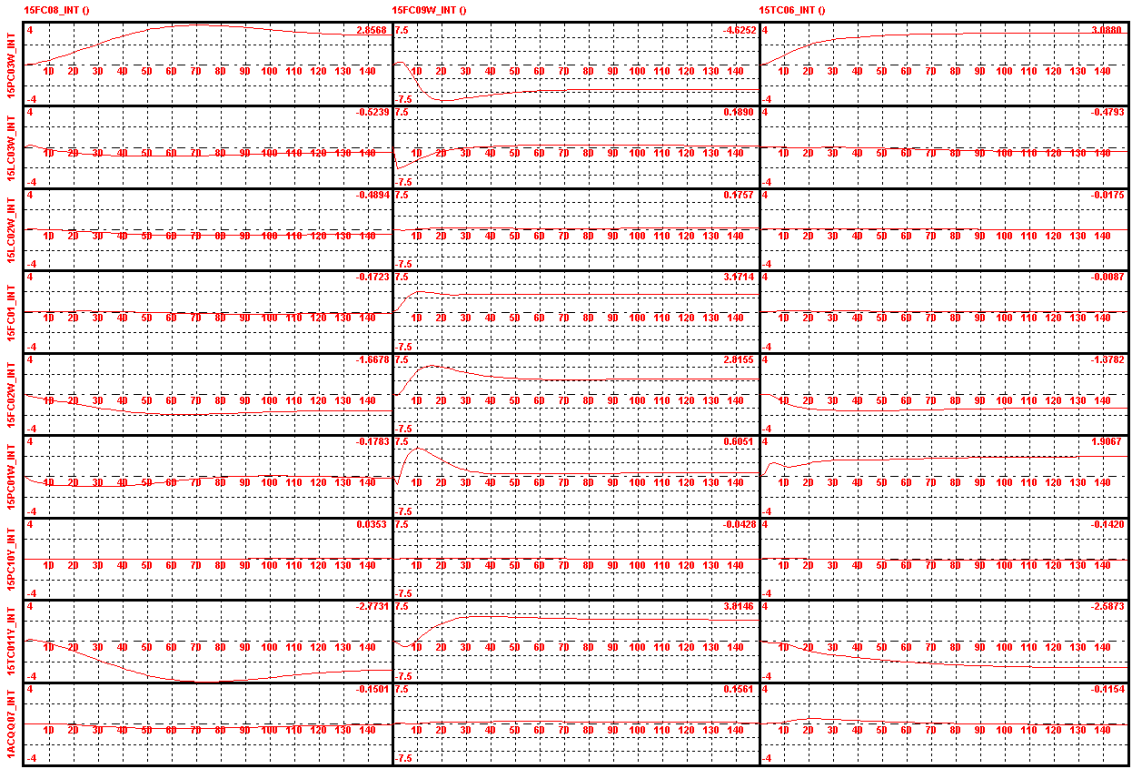
Export State Space Matrices There is an option for user to export a set of state space model equations. If the "Export State Space Matrices" is checked, a set of {A, B, C, D} matrices for each subspace trial will be exported. User can view and use them with other software.

2.1.3 Case Study Examples and Tips on Parameter Tunings in Subspace Identification

Improved Efficiency of MIMO model Identification:

As described earlier, Subspace ID algorithm fits correlated CVs into one MIMO model: where MVs (inputs) pass through a small number of internal dynamic states, and each CV is formed from connections to these internal states. Each dynamic state represents a hold-up like a process volume or some energy accumulation term, or a PID controller state (due to the integrator or I-term, and potentially the derivative or D-term). Every down comer volume, every sump or reflux level, and every tray contribute to the overall dynamic state of the system. In contrast, FIR ID effectively uses a multi-input single output (MISO) structure, and these model structures do not understand that CVs are dynamically linked, as they share internal states. Subspace models will tend to more accurately reflect the interrelationships between similar CVs. This kind of dynamic co-linearity is also important to ensure that the DMCplus move plan is correctly calculated, especially for aggressive MV tuning.

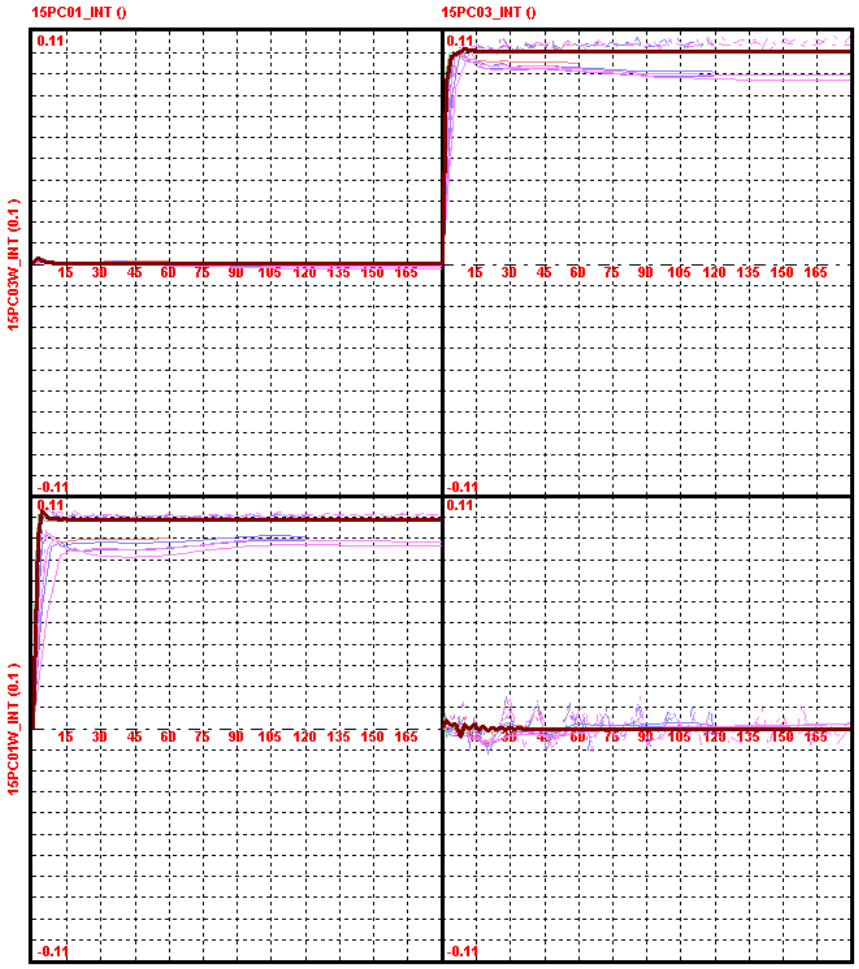
Example – Tray Temperatures Distillation Column: Below is an example of sub-space model identification for sump flow, reflux flow and tray temperatures for a de-isohexanizer column. The maximum order per I/O pair was set at 40, while the selected model order was 9. Note that we have a total of 27 individual model relationships or "model curves" resulting from 3 dependents and 9 independents. It is simply astonishing that a 9'th order model can fit ALL 27 individual MV/CV responses to such a high degree of accuracy. This would not have been possible with MISO FIR models, and demonstrates the power of sharing states in one MIMO state space model.



Interpretation: Note that several of the individual model responses are very similar, so they share similar dynamic characteristics. This means they share some of the 9 identified dynamic "states".

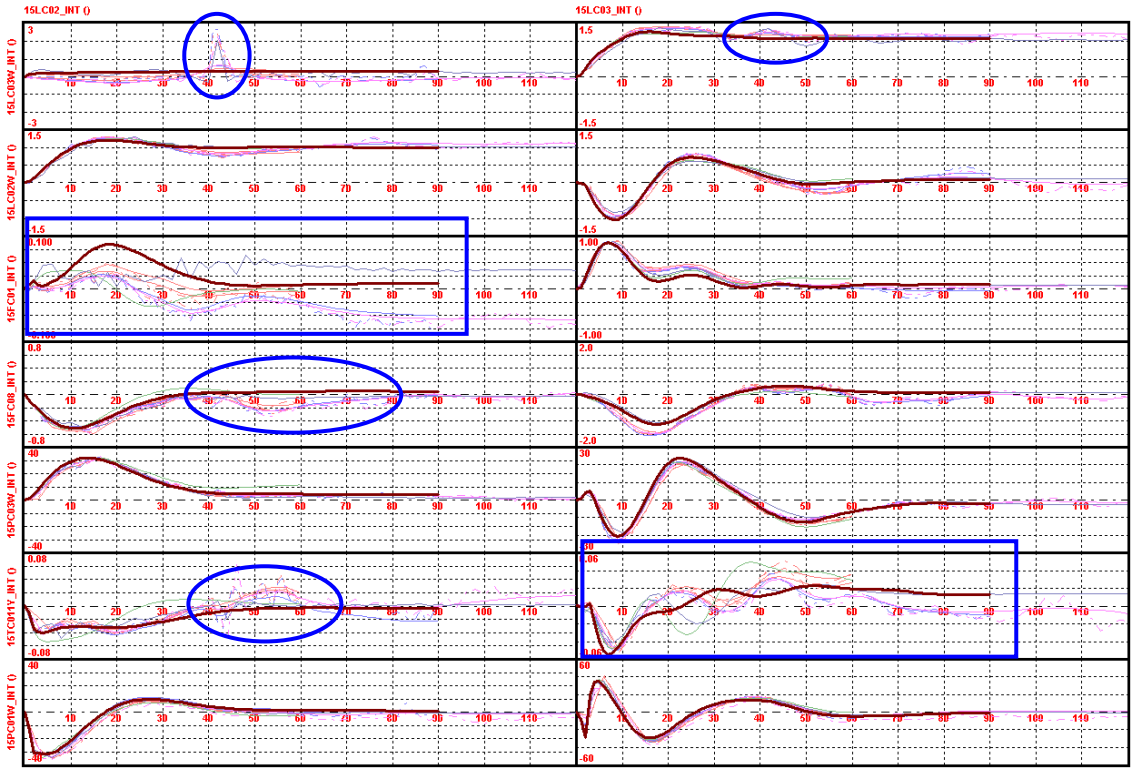
Tips -When to Pick Subspace Models: In many applications, the subspace and FIR results are very similar, and then the subspace results are selected simply because it "interpolates" the somewhat noisy FIR curves with a smoother curve. However, in some applications, especially where major disturbances and feedback correlation problems are contaminating the data set, subspace will often outperform the FIR method due to de-trending pre-processing on data and the power of selected Subspace ID algorithm.

FIR Smoothing Algorithm and Gain Accuracy: It is well known amongst DMCplus practitioners that the FIR smoothing algorithm will adversely affect model responses with fast initial dynamics, like valve positions. Most practitioners will use the un-smoothed response instead, and use curve operations to remove noise and squiggles from the model. Subspace ID now provides a better way to get a smooth response with an accurate gain estimate. Below is an example where we actually know the correct gain, as this is the model between SP and PV and should therefore have a unity gain, while the off-diagonal curves should have zero gain. The bold brown trace below shows the subspace model, and the gains were 0.987 and 1.005 respectively, or about 15% better than the smoothed FIR model gains. Typical move scaling is used below.



Interpretation: Note that all the smoothed FIR responses (the solid colored curves in the background) have 10-15% gain errors, while the un-smoothed responses (the colored dotted lines) achieve the correct gain, but are somewhat noisy. It is clear that the subspace model is both accurate and smooth. The FIR smoothing algorithm has been optimized to accurately fit typical MV/CV relationships, where the initial response is somewhat slower. Note that the un-smoothed FIR response is virtually the same as the subspace models.

Tips - Use PID.PVs and PID.OPs for Sanity Checks: In this specific example, we fit models to the PVs of closed PID controllers. All the other responses should have either a unity gain against the SP, or a gain of exactly zero everywhere else. Auto-scaling is used below.

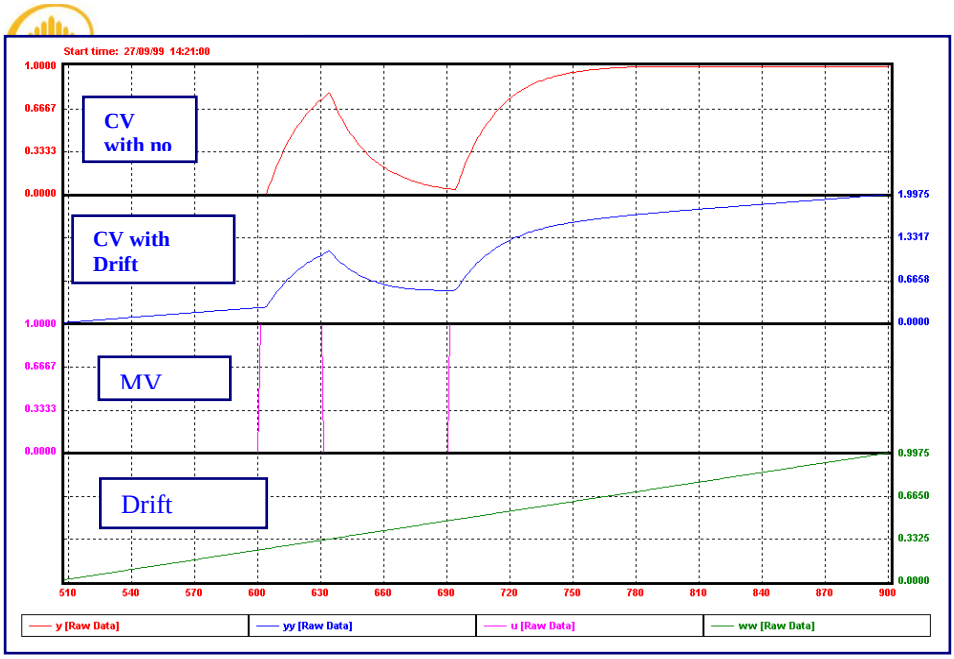


Interpretation: Note that the subspace models have gains that are a lot closer to zero than the FIR models. This is always a good sanity check if you are unsure whether you should trust the FIR or the subspace model gains. In this case, many of these off-diagonal models are so weak that they would have been deleted or zero-gained anyway. However, it is still comforting to know that subspace can accurately identify these gains even if the S/N ratio is very poor.

Use "Detrending" Dealing with Strong Drift Disturbances

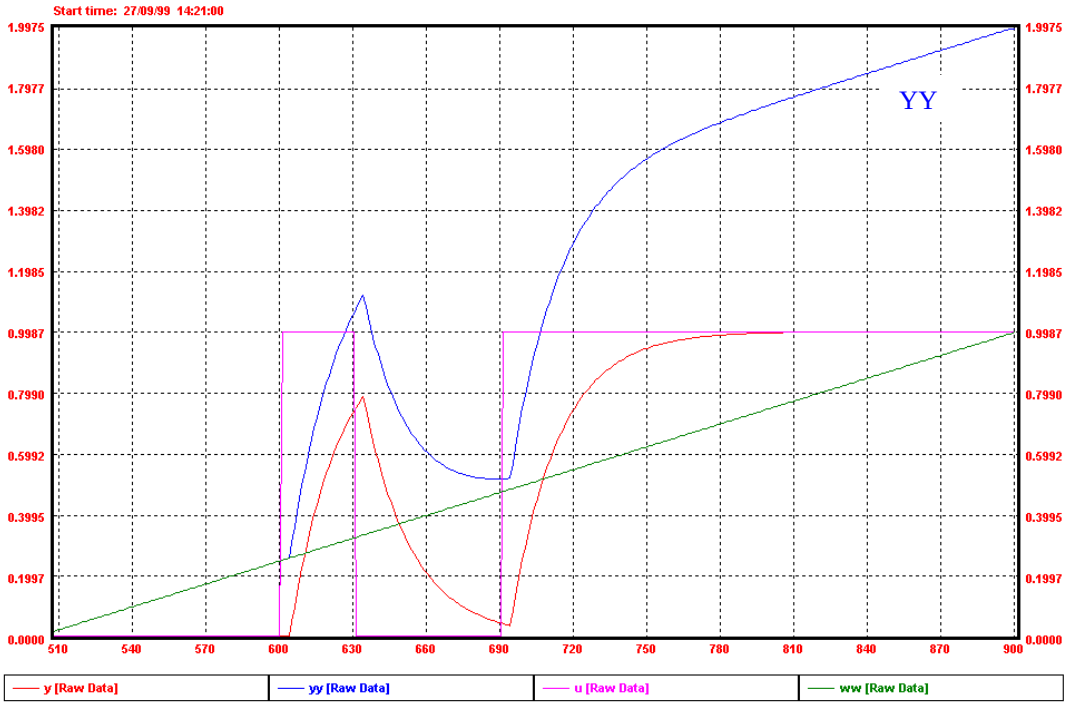
It is well known that FIR model ID is sensitive to slow drift disturbances, especially if the MV moves and the drift effect are correlated. This is often unavoidable as the APC engineer has to make opposing steps to keep qualities or levels within acceptable limits.

Case Study – A simulation example with drift disturbance: The simulation data below was generated using Matlab. Note that a large drift disturbance (WW) was added to the CV (YY).



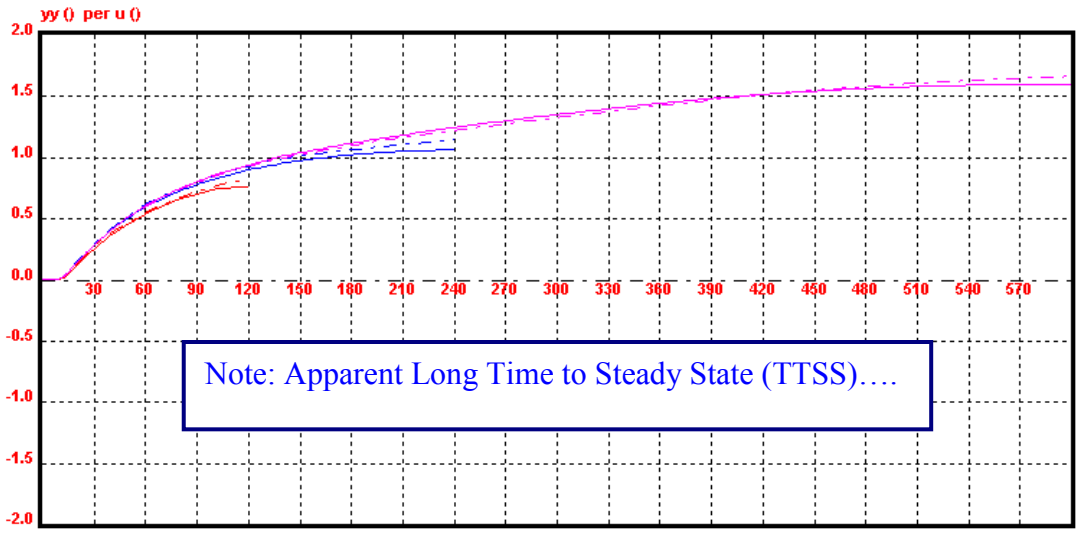
Legend: CV without Drift (y: red), CV with Drift (yy: blue), MV (pink: u), Drift Disturbance (ww: green)

Interpretation: Note that 3 steps are made in the single MV (u), and the last step ends higher (by +1) than the starting value. Note that the net movement in the process CV (y) is upwards, in the same direction as the drift disturbance (ww).



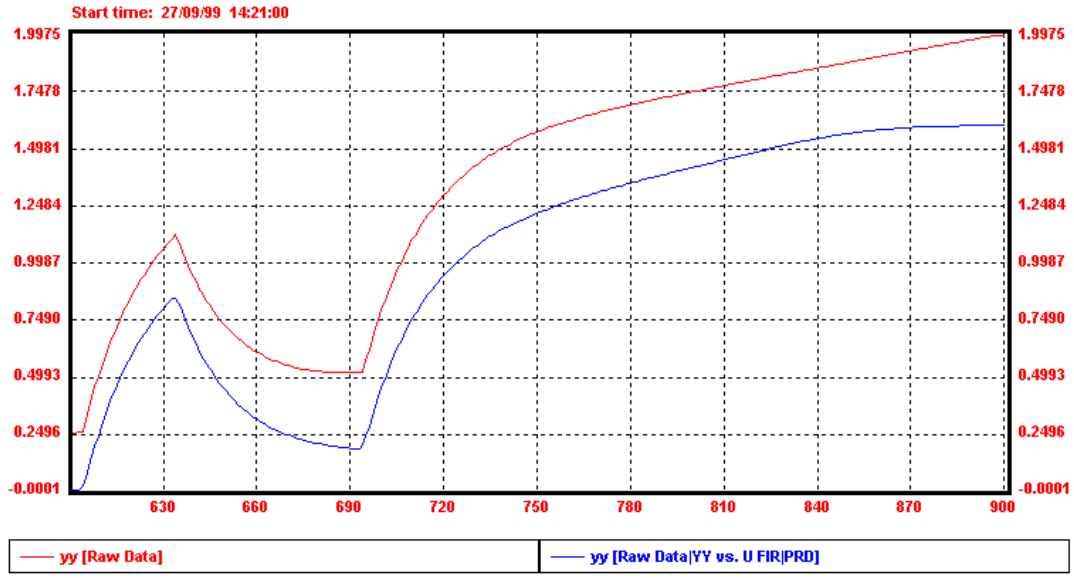
Legend: CV without Drift (y: red), CV with Drift (yy: blue), MV (pink: u), Drift Disturbance (ww: green)

FIR Model ID result: Let's assume that the disturbance is an unmeasured disturbance, i.e., we do NOT have a FF signal. If FIR model ID is used to fit the model <u>without</u> including the drift signal (ww) as a feed forward, then the following FIR ID results are obtained:



Interpretation: The fact that the FIR model curves are all close together, is usually an indication that the model fits the data quite well.

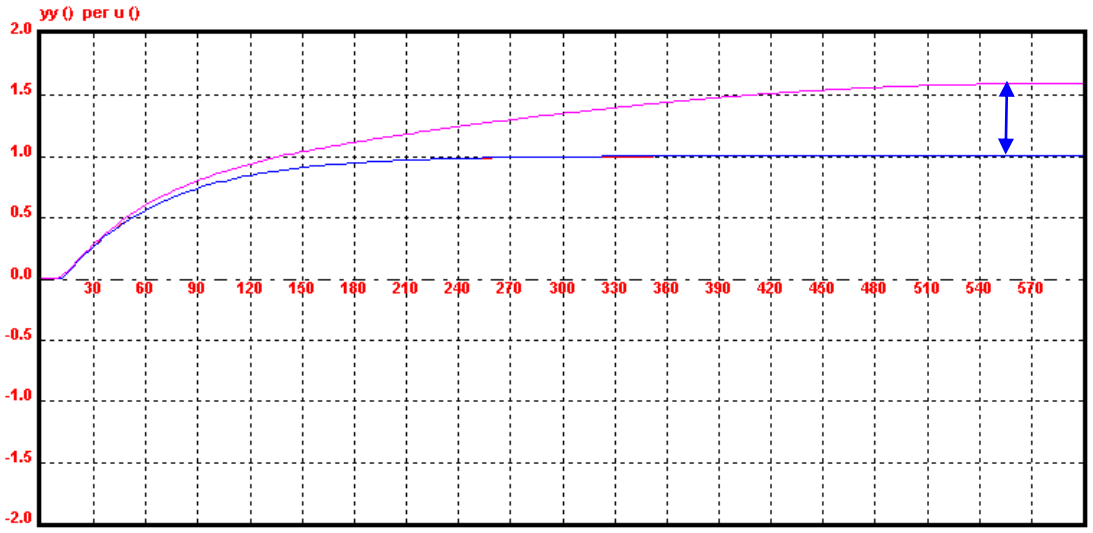
Compare Predictions: The CV prediction based on the FIR model is shown below.



Legend: CV with Drift (yy: red), FIR Prediction (y: blue)

Interpretation: At first glance, the CV prediction looks quite accurate. Since this is a simulation, we have the unusual luxury of knowing the true model: The actual model has a gain of 1.0, and the identified model has a gain that is 60% too high!

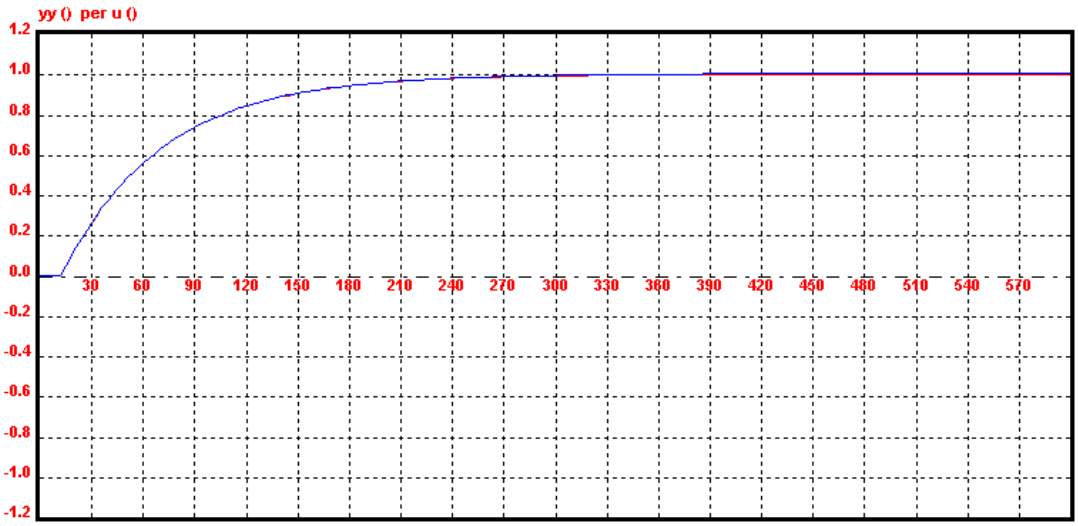
Compare Models: The estimated FIR model and the actual model (with a unity gain) are shown below.



Legend: True Model (red) (which is directly below blue curve), Subspace model without including Drift Disturbance (blue), FIR Model (pink)

Interpretation: The pink FIR model was adversely affected by the drift disturbance, and the gain is too high by 1.6x. The least squares FIR fit will always try to explain the data the best it can, and if the data is corrupted in some way that is highly correlated with the MV moves, then the Least-Square regression may create significant estimation bias and poor model results can be expected.

Comparison – Subspace ID Result: Let's see how subspace id deals with the drift disturbance problem:

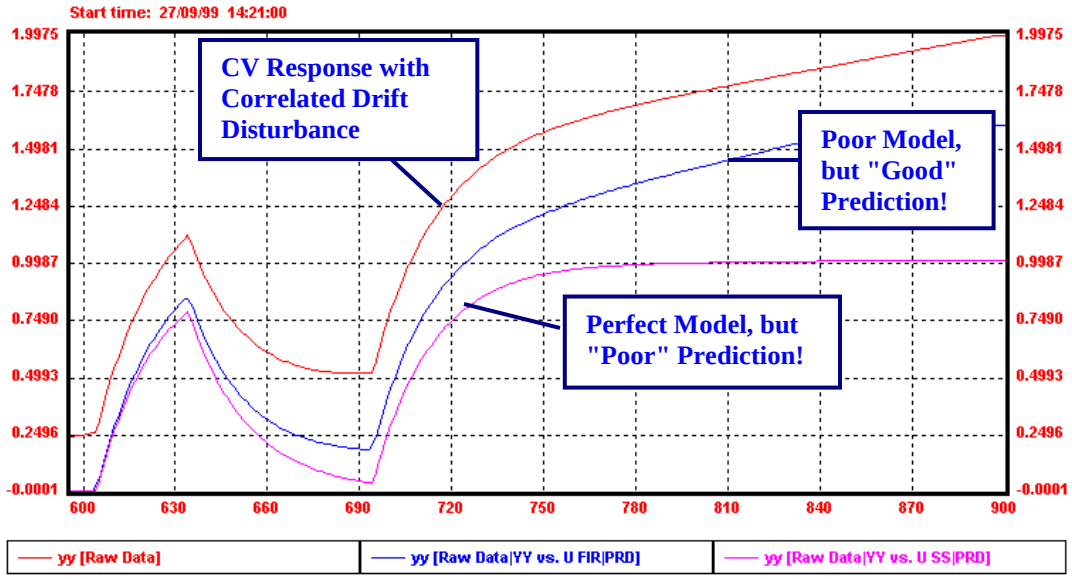


Legend: True Model (red), Subspace model without including Drift Disturbance (blue)

Interpretation: The two models are virtually on top of each other. The subspace model has a gain of 1.005 (instead of 1.000), it is therefore within 0.5% of the true value. It is clear that subspace id has a much better ability to deal with slow feedback correlated drift disturbances than the FIR method. The reason for the difference in performance is that subspace ID uses data de-trending to remove the effect of slow drift from the data by subtracting a low pass filtered (similar to a moving average) signal from the raw CV. This effectively acts as a high pass filter, and only changes faster than 3*TTSS is allowed to pass through the filter. It efficiently removes the drift disturbance.

Tips - Preventing Drift Disturbances from Affecting Model Accuracy: If an equal number of steps with similar step length have been made in both directions (i.e., if the average MV change was zero), then the effect of the drift disturbance would have been much less. It is always a good idea to make an even number of steps in opposite directions and of similar size, and to end the last of say 4 steps at the same value as the initial value before the first step. In this way, even FIR ID can fit reasonable models to the data.

Compare Predictions: Let's compare the FIR model and the subspace model predictions:



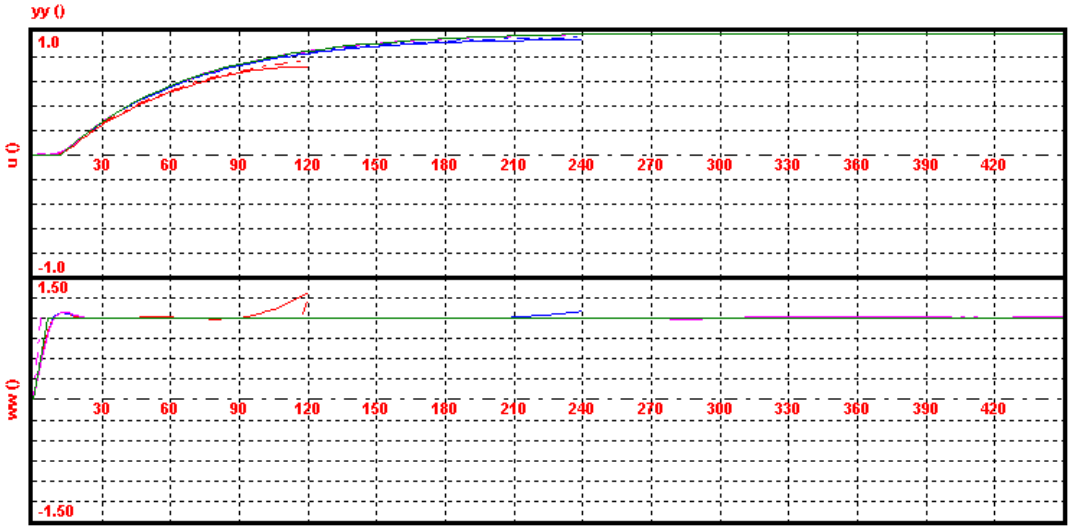
Legend: CV with Drift Disturbance (red), FIR model Prediction (blue), Subspace Model Prediction (pink)

Interpretation: Note that it appears as though the FIR model prediction is far more accurate than the subspace model prediction. The truth in this case is that the Subspace model is far more accurate than the FIR model. The important lesson is that predictions are not always a good indication of model accuracy.

Tips - Prediction Accuracy is NOT a Reliable Model Validation Method: It's always comforting to see good predictions, but if we use the same data for model identification AND validation, like in this example, we can make incorrect conclusions. It would have been better to use a new data set to confirm that the model is good. However, collecting a fresh validation data set is quite expensive if the process has a large MV count and is rarely done in practice. Always keep in mind that a high order model MUST fit the DATA better than any low order model if a statistical R^2 measure is used. If we have a degree of correlation between missing disturbance signals and the MVs, something that is virtually unavoidable in a real step test, then the low order model with a less than perfect prediction may actually be the most accurate model. In addition, by fitting the model to de-trended data, it may well predict the de-trended dataset better than the raw dataset. However, if an appropriate TTSS estimate was used, this apparently "less accurate" model may in fact be the most accurate. This may perhaps be counter-intuitive, but it is true.

Tips - Closed Loop Model Validation: The only definitive way to confirm the accuracy of one control relevant model over another is to put it into use. In our case, we can put one model into an on-line controller, give it fast (aggressive) tuning, make two steps in the active CV limit (or setpoint), then try the alternative model, and compare both observed closed loop responses with the DMCplus Simulate results. Whichever closed loop response matches the simulated closed loop response best, is obviously the more accurate model.

Tips - Feed forward Signals Removes Feedback Correlation: Of course, if you can identify a suitable disturbance (feed forward) signal, then much better models can be identified, and the predictions will look good as well.

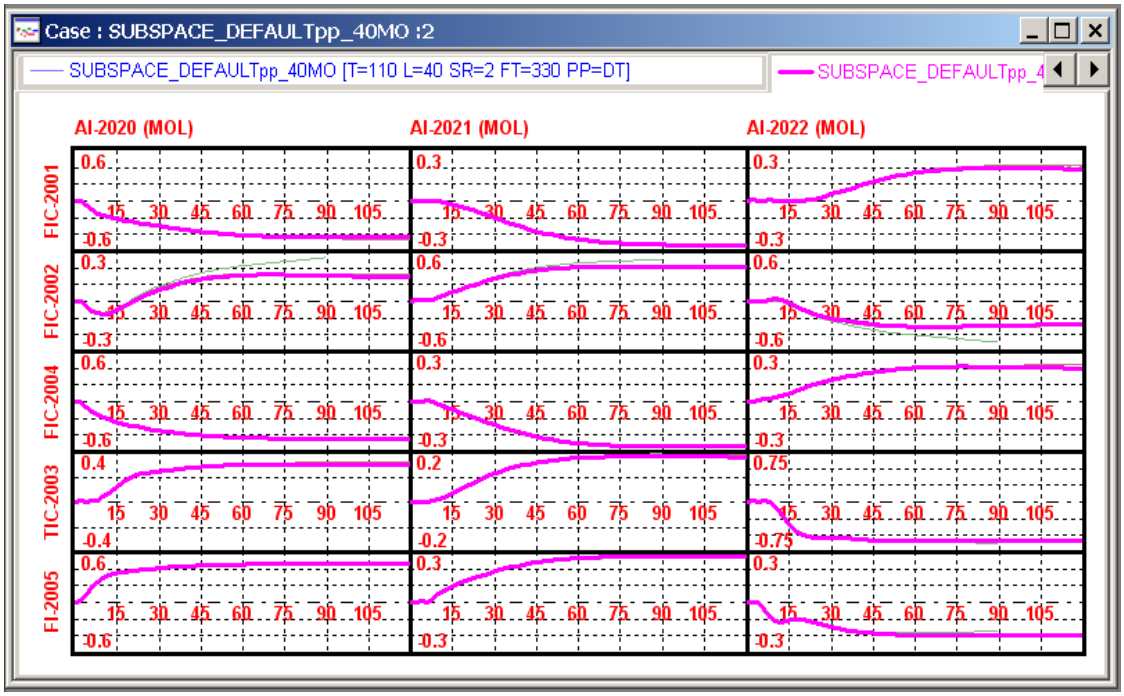


Interpretation: Note that all the models (FIR and subspace models) now give the same results, and the unity gain for both the MV (u) and the feed forward signal (ww) has been correctly identified. In many cases, an integrated prediction error can be used as a feed forward signal to estimate the drift disturbance, but this needs to be approached carefully.

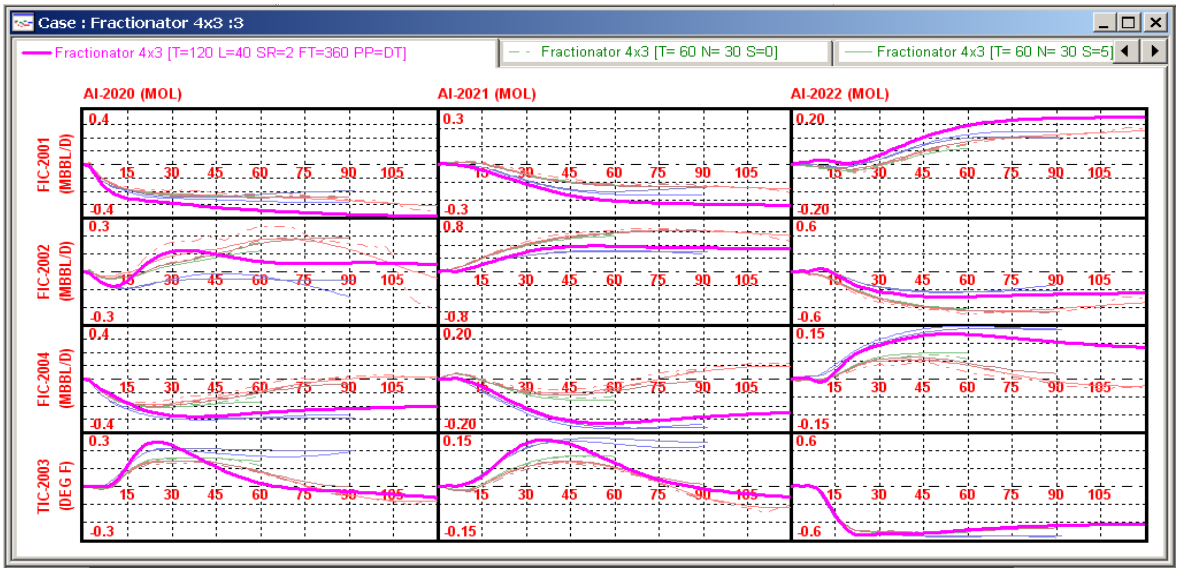
Use "Differencing" to Deal with Unknown Disturbances

Unknown disturbances: It is also often seen in APC practice that there are always some unknown, "un-modeled" disturbances among your data. The following simulation example show you how to get help from the data processing option of "Differencing".

The well known 5x3 Fractionator example included in the DMCplus Model consists of 4 MVs, 1 FF, and 3 CVs, a standard FIR and subspace case run generates the following "perfect" 5x3 models:



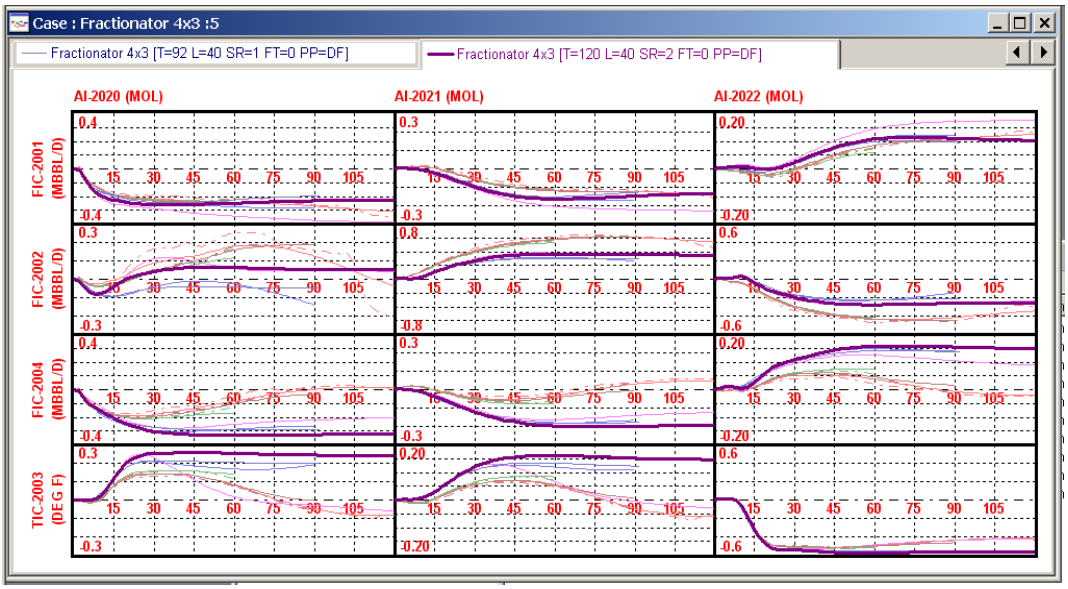
Where the 4 MVs have clean step moves in the dataset, while the Feedforward variable FI-2005 has only slow and drifting changes as a feed flow. For simulation and test purpose, remove the FI-2005 from the original case and re-run the 4x3 case without changing any other default settings. It is found that both FIR trial and subspace trial produced poor ID results: all the models have been messed up, model gains are wrong and dynamics are shifting around as shown below:



Interpretation: The missing feed flow FI-2005 becomes an unknown disturbances to the 4x3 model, as the CVs' data contain those responses from feed's changes. The difficulty

for a ID algorithm to distinguish this disturbances from feed is that the feed changes in a slow, but at similar frequency as MVs to affect the CV responses.

However, in this case all the 4 MVs are perturbed with clean step moves and the Signal/Noise ratios are good. When the data preprocessing options was switched from the default "Detrending" to "Differencing", re-run the 4x3 case and resulted the following models:

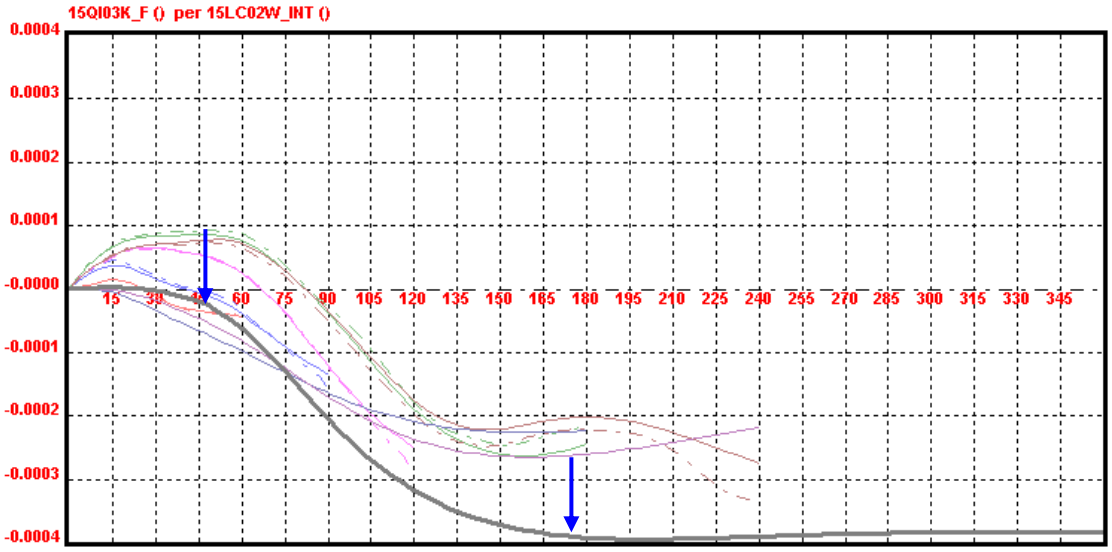


Surprisingly, the correct models came back from this simple switching of the data preprocess by selecting "differencing". Behind the "magic", it is the "differencing" heavily removed the slow disturbances because a "differencing" filter can filter out all slow drafting disturbances. At the same time, this "differencing" filter will also "weaken" the S/N ratio from the data at low frequency range. If the other MVs have relatively "Big" moves and the S/N ratios are strong enough, you may use "differencing" as an effective way to remove unknown slow disturbances.

3 Special Issues and Handling Tips

3.1 How to handle high dead time CVs

Dead Time: In most cases where the process dead times are not very large, subspace models tend to have clean dead times. Unsubstantiated inverse responses are generally avoided. Below is an example of models with dead time: the background colors represent the FIR models, while the bold gray model is the subspace ID result.

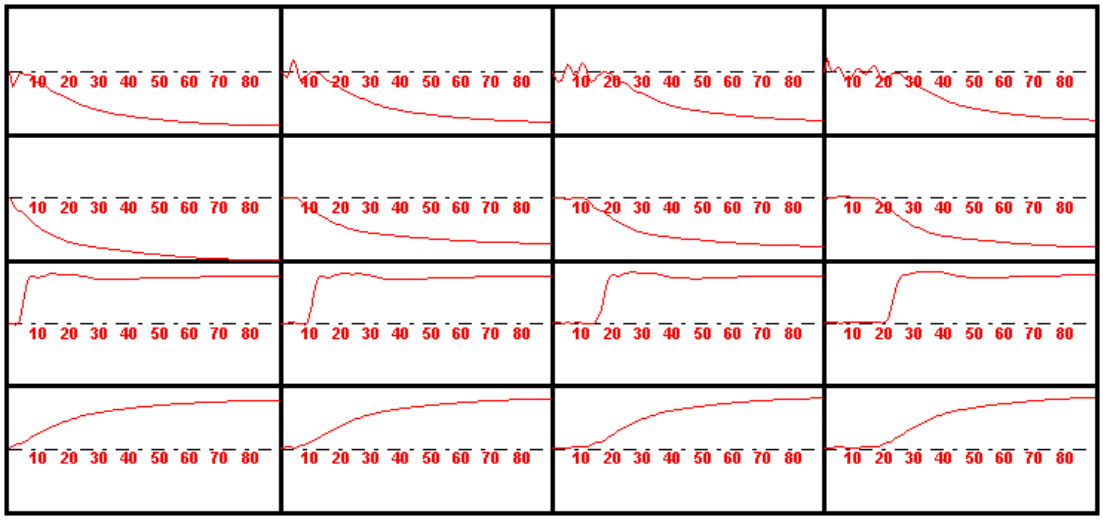


Interpretation: Note that the subspace model (the bold gray trace) displays pure dead time while the FIR models display an inverse response. Note how the FIR traces need to be shifted down to get to the subspace results. This is a typical error that can appear in FIR models with high dead time. In this instance, the subspace result is more accurate.

Subspace id requires one additional model order for every step of dead time, per MV. If dead time becomes excessive, especially for cases with a large number of CVs, then a very high maximum model order has to be used. This is not always practical. It is usually a better approach to shift the CVs with long dead time, and keep the number of CVs in the case low so that a maximum model order in the range 20-45 can be used. Simply shift the model curves back again in the assembled model, see the following for details.

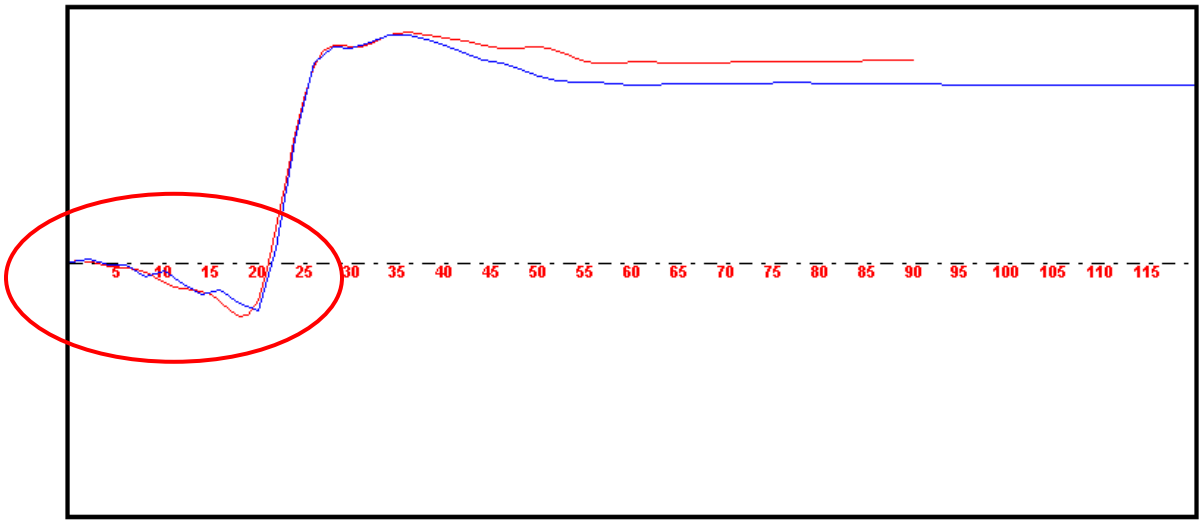
Handling of large dead time:

For large dead time, shift the CV to remove most (but not all) the dead time and use higher Maximum Order Per I/O Pair (30 to 40). If required you could also use shifted versions of the MVs to deal with a large range of dead times. Always inspect the ID algorithm selected model order, if it comes within 3 of the maximum number of states, then increase the model order and run again, or reduce the number of CVs by splitting the case. From AMS 6.0 release, it will issue a warning if the selected model order is within 3 of the specified maximum model order. Note that higher over-sampling ratios (3 or more) will reduce the number of states required to represent dead time. Below is an example of a model for several actual temperatures in a series of Isomerization reactors in series. Every subsequent reactor has more dead time than the previous one.



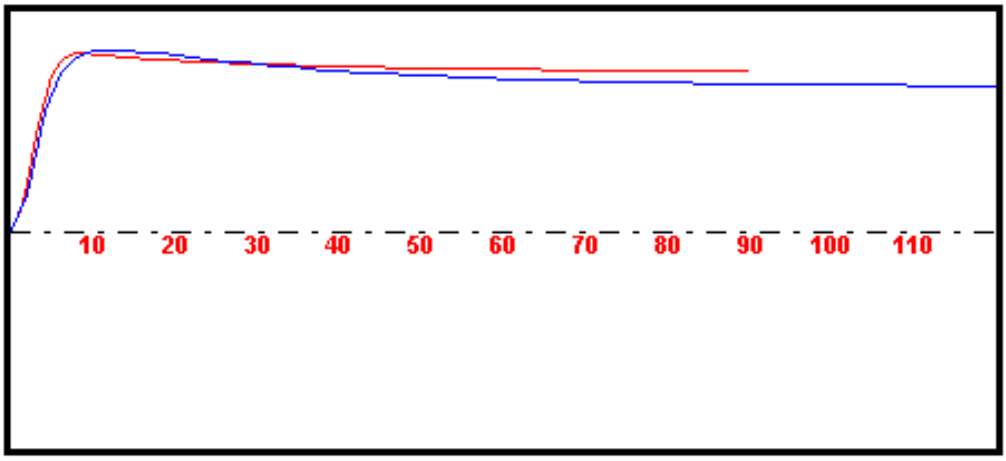
Interpretation: Note that minimum dead time increases from 3 minutes for the first reactor, to 10 minutes for the inlet temperature of the second reactor, to 13 minutes for the outlet of the second reactor, to 20 minutes for the last reactor. This is the expected behavior. As mentioned before, every step of dead time can potentially consume a single model order, and the selected model order will increase significantly compared to if the CVs were shifted to remove the common dead time. [Note that the dead time for the models against the first MV has been approximated as a series of small squiggles. This is the so-called Pade approximation at work, where exact dead time is represented approximately as a series of nearly canceling poles and zeros.]

The Benefit of Shifting CVs to Remove Dead Time: Let us look at an example by selecting the CV shown above with the largest dead time, and for now let's only concern ourselves with the major MV. If we do not shift the CV at all, we get a slight inverse response as shown below:



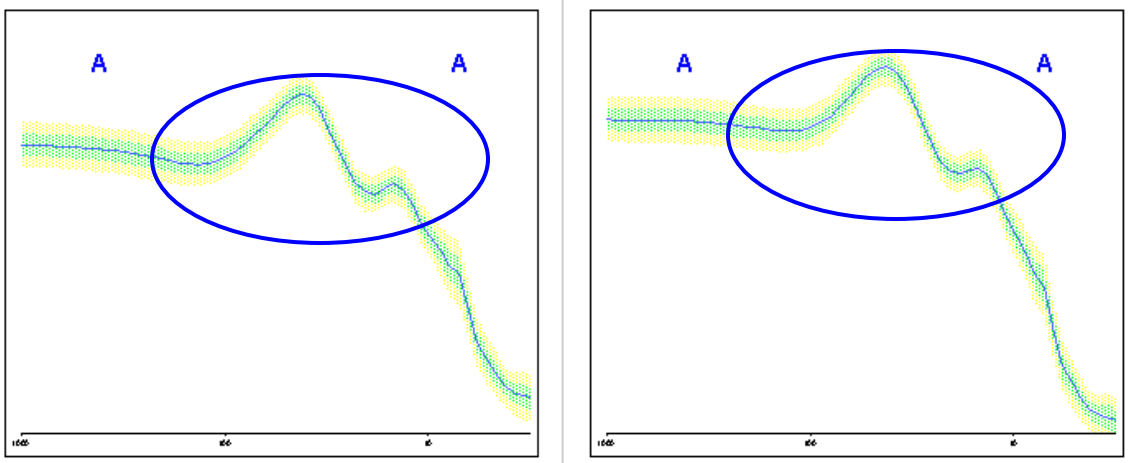
Interpretation: Selected model order was 9 and 8 respectively. This is significantly less than we would expect for a process with 20 minutes of dead time and a fairly complicated high order process response. [Clearly, the model order selection algorithm is taking a shortcut and fitting the long dead time with a Pade-like set of poles and zeros.]

Now Shift the CV and Fit the Model: If we shift the CV by 20 minutes, we get the following result:

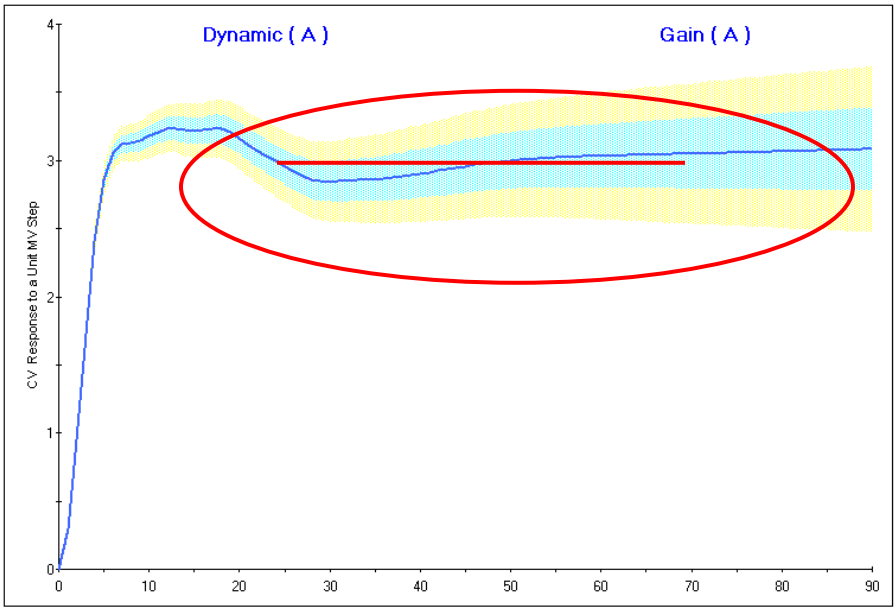


Interpretation: Note that the dead time is gone, and that the model is simpler than before (lower model order). Selected model order was 3 and 2 respectively, which is lower than what we expect.

Tips - How Can We Know If the High Order Dynamic are Real or Not? One way to do this is to calculate model uncertainty for the reactor temperature CVs. Let's inspect the frequency domain (Bode plot) uncertainty for these CVs:

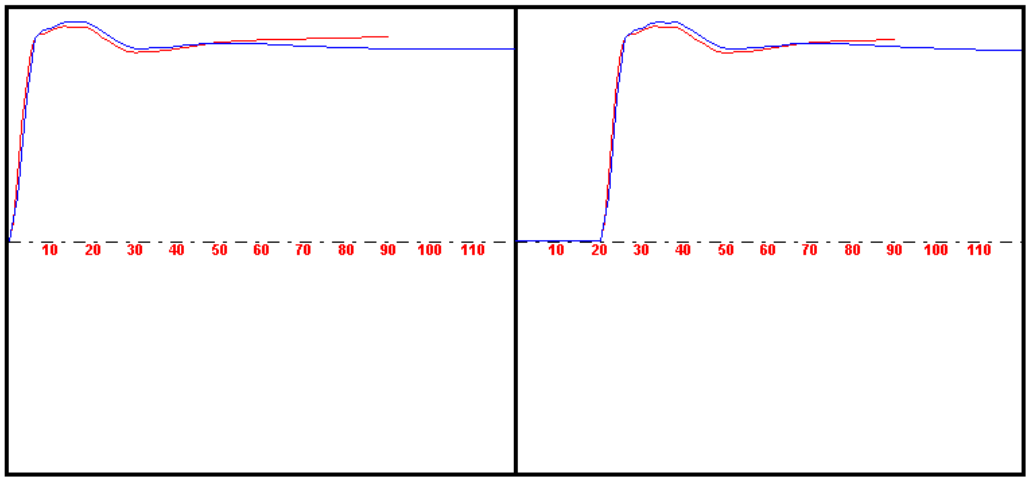


Interpretation: The high order process dynamics appear to be real, as the uncertainty band is very narrow and the size of the high order effects in the Bode plot exceeds the amount of uncertainty present in the Bode plot by a large amount. In addition, the curves got A-grades for both gain and dynamics. Let's inspect the time domain (step response) uncertainty just to make sure:



Interpretation: If we extend a line from the peak of the step response to the right, we can find out if a simple first order plus dead time (or at least an over-damped second or third order model) would have worked just as well. If the line stays inside the 2-sigma bounds, then there is a chance that the squiggle is not real after all and could have been removed. In this case, the uncertainty bounds are narrow enough to ensure that this is not possible. We therefore have to conclude that the high order dynamic is real, and that it was dropped for the case that contained only a single CV. [This behavior is not totally unexpected, as the automated model order determination method works better if multiple dynamically related CVs are included in the same case. In tends to under-fit model order for cases with only a single CV, or a small number of MVs.]

What Happens If We Include BOTH CVs in the Same Case? If we copy the case and include both the shifted and un-shifted CVs in the same case (with only the major MVs included), we get a better result:



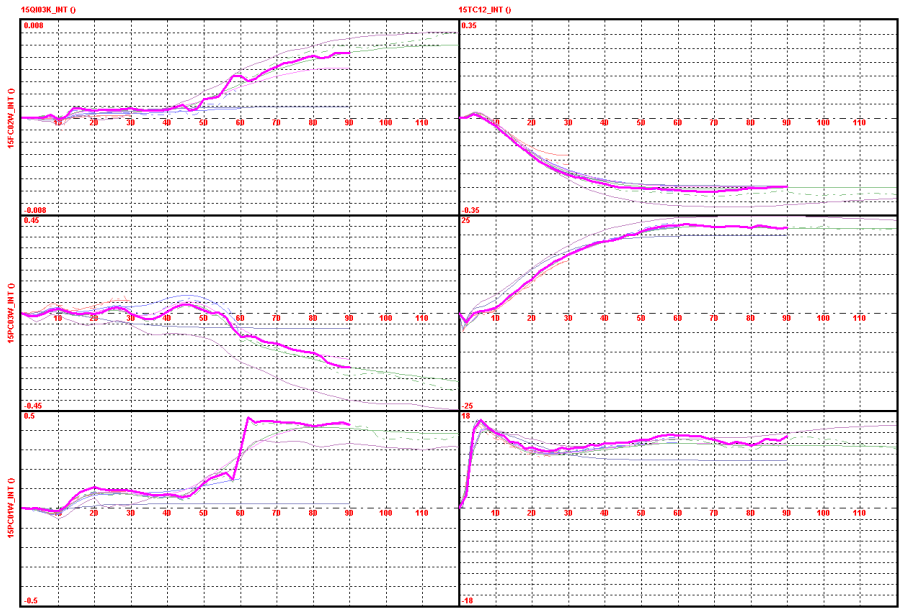
Interpretation: Note that the shifted and un-shifted CVs have exactly the same dynamics, and the dynamics now include the slightly reduced response starting at 20 and 40 minutes respectively (which is real due to the presence of a recycle). It is interesting (but not surprising) that the second squiggle is a whole multiple of the dead time. The time it takes for the upstream effect to reach the last reactor is 20 minutes, and then it recycles back to the first reactor and takes another 20 minutes to reach the last reactor a second time (transport lag). However: Note that the model identification algorithm produced exact dead time in this case rather than a series of small (Pade-like) squiggles. [This is NOT an assembled model where we shifted the CV via a curve operation!] For the TTS=90 minute model, the selected model order was 25 (20 to represent the dead time, and 5 to capture the complicated process response). For the TTSS=120 minute model, over-sampling of 2x is used, and therefore there are only 10 steps of 10 dead time (with every step = 2 minutes). The selected model order for the 120 minute model was 15 (10 to represent the dead time, and 5 to capture the true dynamics.

Tips - Model Identification Methodology for High Dead Time CVs: This is a pretty impressive results and points the way to a better methodology to enforce the dead time of CVs with long delay:

- **Use a Shifted CV:** Shift the CV left by an amount slightly less than the minimum dead time. [Make sure that this minimum dead time applies to all the useful MVs and remove the insignificant MVs from the case.]
- **Add Both CVs to the Case:** i.e. both the raw CV and the shifted CV in the same case, so you can eventually drag the models of the un-shifted CV into the assembled case with no need for a Shift Left/Shift Right Curve Operation.
- **Modify Slicing:** Remember to change the slicing for the shifted CV at the end of the vector, as it will be shorter in length than the raw CV by the amount of shift.
- **Use the Un-Shifted CV in the Assembled Model:** Drag the Model of the un-shifted CV into the assembled model, as this ensures that no further dead time correction is needed (no Shift Right Curve operation).

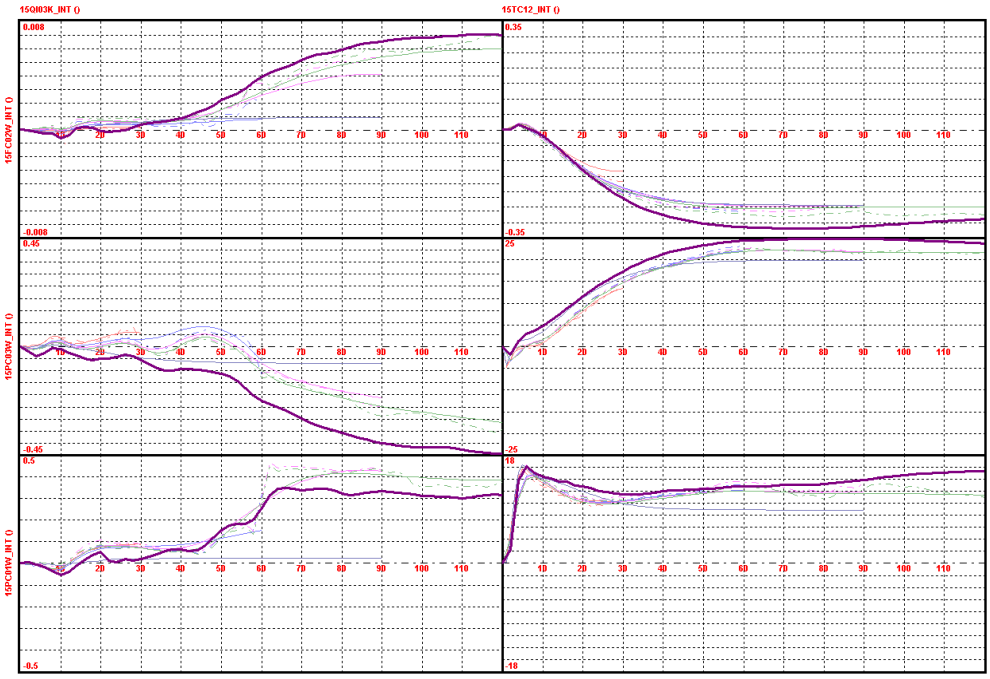
3.2 How to Deal with Analyzers CVs

Use of Filters to Smooth Out Analyzer Signals: You must first smooth analyzer signals using first a central average of at least 1.5 * analyzer update time, then use a small time constant (exponential) filter to round the edges, or it may fail. If the analyzer update time is 15 minutes, then set the CAV filter to about 1.5x, say 25 steps, followed by an exponential filter of 5-7 minutes. The example below shows the un-smoothed FIR model:



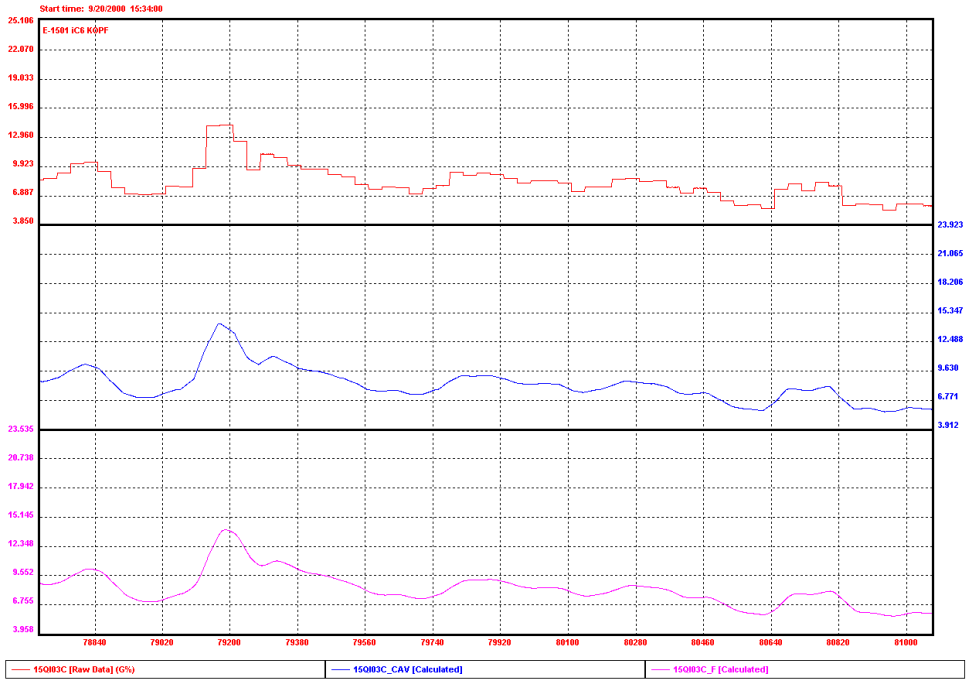
Interpretation: The unsmoothed FIR (S=0) fit is clearly very poor. If the FIR smoothing factor is used (S=5), the FIR model curves improves significantly. This indicates that smoothed FIR models are not particularly sensitive to discrete analyzer signals.

Subspace Model Identification Results for Analyzer Signals: Below is an example of an analyzer CV that was fitted with subspace ID with NO filtering:



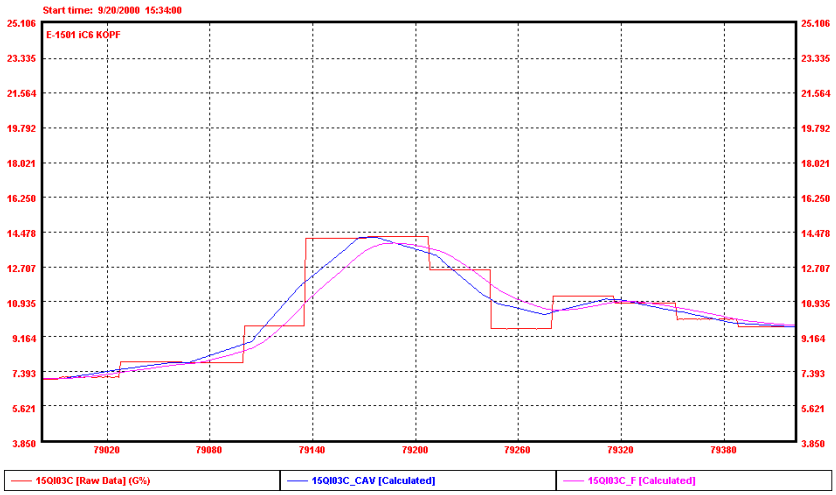
Interpretation: It is obvious that the subspace model tried to explain the staircase movement in the signal with high-order model.

Tips - Recommended Approach When Dealing With Analyzer Signals: The best approach to improve the accuracy of both the FIR and subspace models is to apply proper filtering to this CV as shown below:

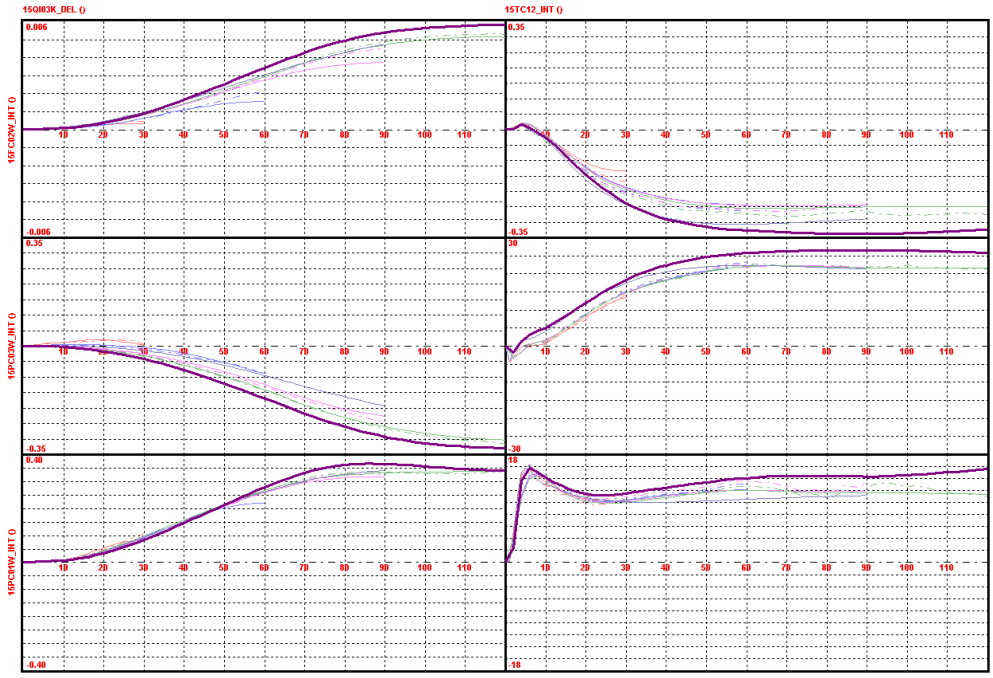


Legend: Raw analyzer Signal (in red); CAV filtered signal (blue); exponentially filtered CAV signal (pink).

Interpretation: Note that the CAV filtered version (in blue) is similar to a linearly interpolated version, and applying the exponential filter smooth out the sharp edges. Also, the CAV filter causes a phase shift to the left (a non-causal effect, meaning the CAV filtered version may responds before the MV steps). The exponential filter causes a phase shift to the right, and to some extent cancels out the shift to the left introduced by the CAV filter (negative phase errors) as shown below.



Model ID Results for Smoothed Analyzer Signals: The resulting models are below:



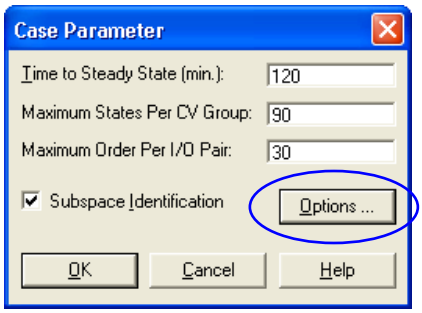
Interpretation: In this case, the QI03K was CAV filtered and then an exponential (time constant) filter was applied to further smooth it. We then set up a case and determined that the minimum dead time was 10 minutes at least, shifted the CV left by 8 minutes, and refitted the models. These models are suitable for control. [Of course, the assembled model curves will be shifted back again to make sure the dead time is correct.]

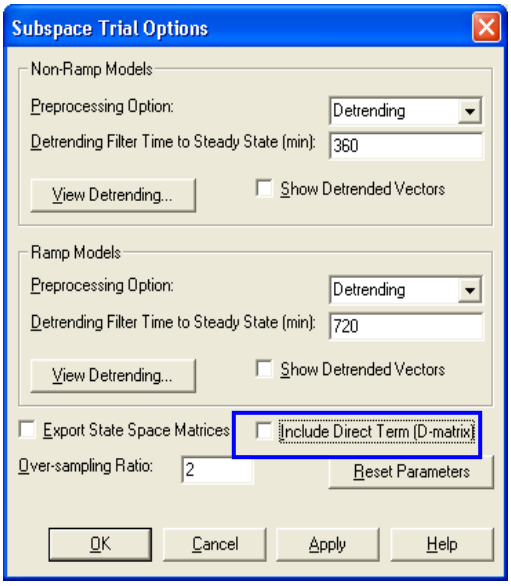
3.3 Removing Undesirable Direct Transmission Effects from the Model Curves

Introduction: For some CVs where we know there is no direct transmission term (no immediate kick in the CV), we can go to the Options Menu and turn off the direct transmission term, which means the models will not contain any direct kick anymore (the response at time instant 1 will be zero):

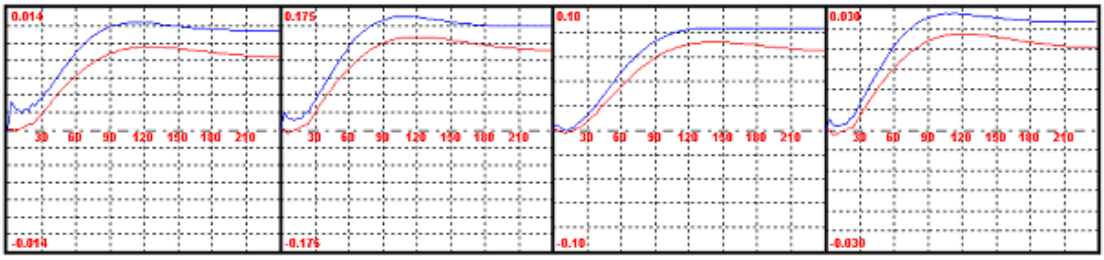
Workflow: The following steps can be followed:

- Step 1: Select a trial and click on Options :
- Step 2: Then uncheck the "Include Direct Term (D-Matrix)" (which is normally turned on):





Model Identification Results: A comparison is shown below of a subspace model fitted to the data using the direct term (which is not strictly required here) shown in blue, and a model where the direct term was turned off (in red):



Interpretation: Note that most of the inverse responses have disappeared from the red model curves, and the gains are somewhat different. All the curves have shifted down by approximately the direct transmission amount (by disallowing the direct initial kick in the CV).

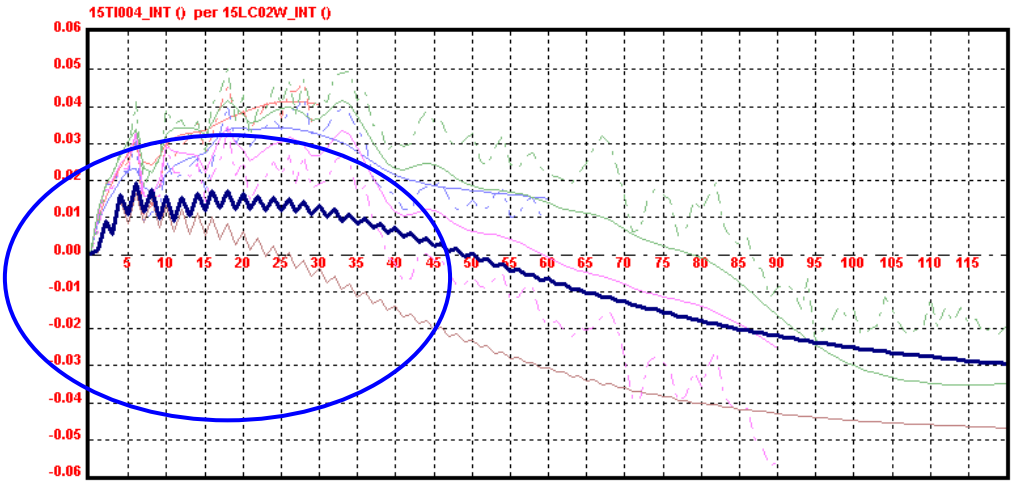
Benefit of Removing Undesirable Direct Transmission Effects: This feature helps a lot to reduce unwanted and unrealistic inverse responses in the model curves, especially if some closed loop (or Reposition mode) data is included in the cases, and makes the gains more accurate. Of course, if ANY curve in the multi-variable model has an immediate kick (e.g. between SP and OP), then we must NOT turn off the direct transmission term,

and if we do, the model will be very inaccurate. All CVs will then be inaccurate, not just the CV with the direct kick.

3.4 How to Deal with Periodic Limit Cycle

Subspace ID and Periodic Limit Cycles: If the CVs contain significant periodic limit cycles due to sticky valves, or actuator hysteresis due to loosen linkages, subspace ID may attempt to fit these cycles as it will assume that they are due to true process dynamics. This is not completely unexpected, and there are two simple ways to resolve the problem. The user may try to limit the "Maximum States per CV Group" when run a subspace case and use a "trial and error" approach by reducing the "Maximum States" gradually until a smooth model is achieved. The details have been described in section 2.1.2 (Case Parameters for Subspace Identification). The second way for both offline modeling and online deployment is to use a Moving Average (MA) filter with exactly the same filter length as the period of the cycle to remove the cycle and smooth out the CV.

Impact of Valve Cycles: An example is shown below for a relationship with a very poor S/N ratio where the CV has a small limit cycle due to a valve problem.



Interpretation: Note that both the smoothed and un-smoothed FIR responses in the background display a small cycle, but the cycle in the subspace model (the bold blue line) is more severe. In some ways, this is due to the subspace algorithm trying too hard to fit the data. In the final DMCplus application, this model would be deleted, as it is so small on typical MV move scaling, that it will contribute little or nothing to the predictions. If you find that subspace suffers from this problem, then simply filter the CVs using the @CAV(A,n) or @SHIFT(@CAV(A,n),n/2+1), where A is the CV vector, and n is the period of the cycle.

MA Filters Contribute to Overall Model Order: Just remember that every filter order contributes to the overall state count of the combined model. A moving average (MA) filter with a window length of 10 is effectively a 10'th order filter, while an exponential filter is a first order filter. Let's look at an example. Assume we have a total of 5 MA filtered CVs included in the case, and a 20'th order model would have worked well if the cycle was absent and we did not need to use the MA filters. The 5 MA filters each contribute 10 model orders to the case, in effect contributing 50 model orders purely due to the filters. Therefore, unless we specify the maximum order at 20 + 50 = 70, we will not be able to fit a good model. Ideally, when using slow MA or Central Average (CAV) filters, reduce the size of the case to one or two CVs, and set the max model order to Nr of CVs*Filter Order + 20.

3.5 Closed-loop Test and Subspace ID

Closed-loop Identification: it has been a popular topic and drawn a lot of attentions in recent years in both academic research and industrial APC practice. There are many known benefits if the identification can be successfully performed based on closed-loop data, e.g. interruptions and disturbance to process can be minimized, collinear models' gain can be accurately identified, and the models' uncertainty in identification can be reduced, etc. On the other hand, there are also many challenges to the existing identification algorithms, specifically in industrial APC practice. The major problem appears when unknown disturbances exist and they are correlated with MV moves. Typical case is where MV moves are trying to correct/compensate for CV changes due to unknown disturbances. In many published case-studies, different strategies have been proposed and investigated recently, including the so called High-order ARX (HOARX) approach and a few subspace ID algorithms. Though there have been several algorithms supposed to be able to do closed-loop identification in theory, there is still no guarantee to be successful in industrial APC practice as many assumptions cannot be satisfied. For example, most of the proposed approaches such as HOARX and many of closed-loop subspace algorithms produced large gain mismatches if the S/N is poor.

AspenTech's New Solution:

AspenTech has made consistent efforts in recent years trying to find a better solution for our product users to deal with this challenging closed-loop ID issue. As a result, **two good news for DMCplus users**: (1) **SmartStep®** as a proven new test technology to provide safe and efficient plant test with simultaneous multiple MV perturbations; It produces maximum S/N ratio and generates high quality test data for identification. (2) DMCplus Model uses **Canonic Variate Analysis (CVA)** based subspace ID algorithm that is ONE of the few "consistent" subspace methods capable of dealing with correlated closed-loop data. There have been a number of references about this algorithm.

Feedback Correlation Example: Subspace identification was available in the 90's and it is far superior to any previous parametric methods for MIMO process modeling. It

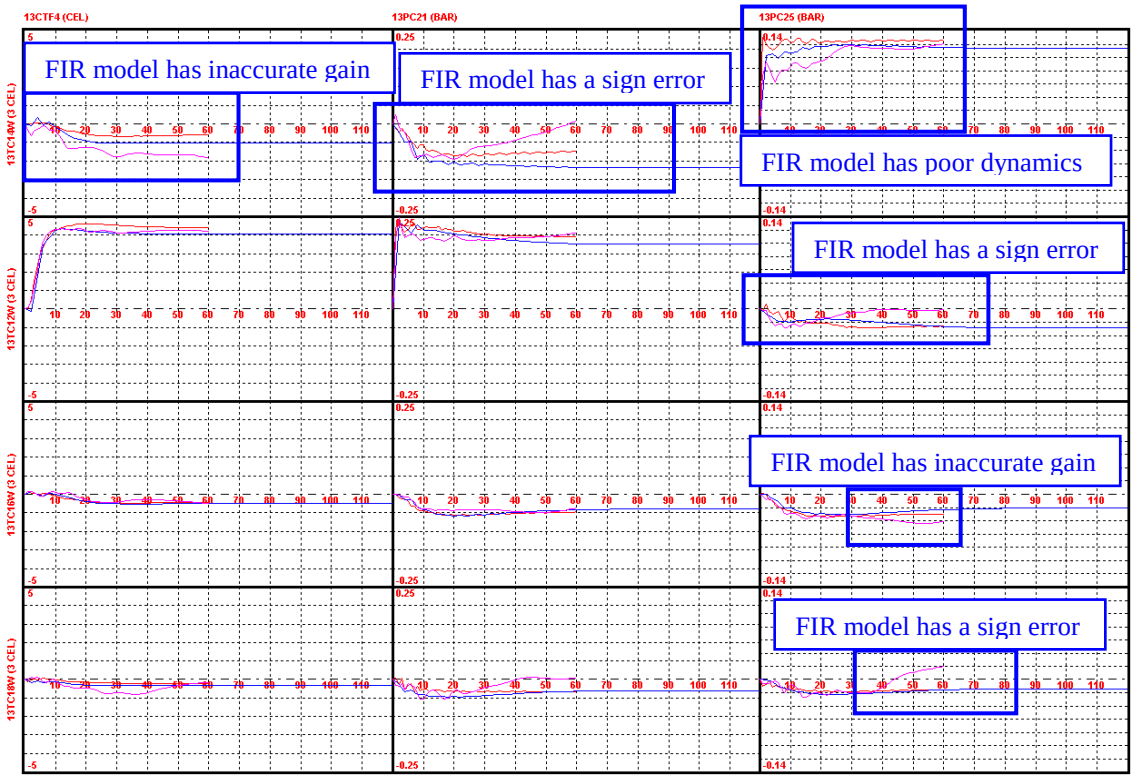
performs well in difficult process applications, especially where large unmeasured drift disturbances and unfortunate correlations between MVs and the unknown disturbance leads to significant feedback correlation. Feedback correlation is often inadvertently introduced if engineers are forced to counteract drift in the limiting CVs by making opposing correcting moves to keep the product quality on specification, prevent valves from saturating, or to keep levels within safe bounds. Subspace ID deals much better with autocorrelation (periodicity) in the MV data compared to the FIR method.

In the example below, we removed all the weak MVs from the case as well as the ambient temperature Feed Forward (FF) signal to introduce feedback correlation on purpose. We also limited the amount of data to be used for model ID and made sure that this data set contains a lot of correlation between the MVs in the case and the missing MVs.

Worst-Case Scenario: This is the worst-case test case we can construct:

- Small data set.
- Highly correlated MV steps.
- Significant feedback correlation by removing some MVs.

Comparison: Let's compare the FIR and subspace models for this highly degraded data set (with several missing MVs) to the models fitted to the most uncorrelated data set.



Legend: Best Available Model fitted to 8000 data points (red); FIR model with missing MVs and 800 data points (pink); Subspace model with missing MVs and 800 data points (blue)

Interpretation: Consider the red model to be our best estimate of the true process model as it is based on a total of 8000 data points from a high quality SmartStep® test (with all MVs included), while the blue and pink models are from a data set consisting of only 800 data points (10x less data), with several missing MVs. The pink curves are the FIR models (smoothed and unsmoothed) based on a short 800 point data set with lots of feedback correlation and high MV cross-correlation. The blue curves are the subspace model curves fitted to the same degraded data set. Note that the blue subspace curves are much closer to the red, than the pink FIR model curves, which has several sign errors as well. This is not surprising, as the subspace model identification was designed to outperform all other methods in such a difficult application. So what is the point of this experiment? Obviously, we are not on purpose going to forget to include important MVs that do have a strong effect on the CVs. The point is that we often do not have suitable Feed Forward signals for the major disturbances, and it is then impossible to include these as Independent Variables in the cases. Excluding MVs on purpose, as an experiment, as a useful way to demonstrate what could happen if we cannot include all the important feed forward variables. This is a problem in all MPC projects. This experiment clearly showed that subspace outperforms FIR if we have missing independent variables (whether they are MVs or FFs), especially if there is strong correlation between the MVs we did include and the missing independent variables. This effect is called feedback correlation, and is always present in every project we work on.

Tips – Using Closed Loop Data: In closed-loop identification, the key to success is a good S/N in closed-loop data. Once you received high quality data with good S/N ratio, combining with the use of "consistent" CVA based subspace algorithm, the closed-loop identification becomes feasible and practical. It should be pointed out that sufficient MV perturbations and clear CV responses are always required to satisfy closed-loop identifiability. Normal process operation data under APC or PID control may not meet this requirement. The SmartStep® test technology will be very useful to generate good data and is highly recommended to use.

In many practical applications, up to 30% of closed loop commissioning or SmartStep® Reposition mode data can be added to the open loop step test data to improve model accuracy, specifically gain ratio accuracy. As long as the closed loop data contains significant excitation e.g. from CV limit and tuning changes, and large process upsets are sliced out or an appropriate feed forward signal is added, the model ID results generally improve. In our experience, the sub-space results will be negatively affected if the closed loop data moves only in response to disturbances for which we do not have FF signals, or if the total amount of closed loop data exceeds 50% of the total data set. Data like this should be sliced out.