# Configure Dynamic ACM Model

To demonstrate the feature, an example named Van-de-Vusse reactor is provided in a subfolder “C:\Program Files (x86)\foqus\foqus\_2015.6.0\examples\DRMBuilder\VdV” under the FOQUS installation directory. The nonlinear reactor model is a benchmark process popular in control literature. The following are the steps to build a D-RM using DABNet framework and BP-based ANN training:

1. Copy the ACM file of the Van-de-Vusse reactor example “C:\Program Files (x86)\foqus\foqus\_2015.6.0\examples\DRMBuilder\VdV\VdV\_Reactor.acmf” to the FOQUS working directory. Confirm the user of the D-RM Builder has write-permissions to this folder.
2. Launch SinterConfigGUI from Window’s “Start” menu or its shortcut on the desktop. The GUI window as shown in Figure 1 displays.
3. Click the “Browse” button and browse to the “VdV\_Reactor.acmf” file in the working directory. Then Click the “Open File and Configure Variables” button. It takes a few seconds for the SinterConfigGUI to bring up a “SinterGonfigGUI Simulation Meta-Data” window as shown in Figure 2. An ACM window with VdV reactor model inside also displays as shown in Figure 3.
4. Enter “VdV Reactor Example” in the “Title” text box, “My first DRMBuilder tutorial” in the “Description” text box, and your name in the “Author” text box. Then click the “Next” button. A “SinterConfigGUI Variable Configuration Page” window displays as shown in Figure 4.
5. Enter “~” in the “Variable Search Pattern” text box and click the “Search” button. All the variables in the VdV reactor model are listed in the list box at the lower left corner of the window as shown in Figure 5.
6. Select the “VdVR.Feed.Ca” in the list box and click the “Lookup” button at the upper left corner. The variable is displayed in the “Preview Variable” section. Click the “Make Input” button and the variable is appended to the end of the “Selected Input Variables” table. Change the name of the variable to “Ca\_Feed” and check the checkbox in the “Dynamic” column. This makes the variable as the dynamic input variable.
7. Select the “VdVR.Feed.F” in the list box and repeat Step 6, make it the dynamic input variable and change the name to “F\_feed”.
8. Select the “VdVR.V” in the list box and repeat Step 6, make it the dynamic input variable and change the name to “V\_Reactor”
9. Change the default value of the input variable named “RunMode” in the “Selected Input Variables” table from “Steady State” to “Dynamic”.
10. Select the “VdVR.Prod.Ca” in the list box and click the “Preview” button. Then click the “Make Output” button and the variable is added to the “Selected Output Variables” table. Change the name of the variable to “Ca\_Product” and check the checkbox in the “Dynamic” column. This makes the variable as the dynamic output variable.
11. Select the “VdVR.Prod.Cb” in the list box and repeat Step 10, make it the dynamic output variable and change the name to “Cb\_Product”.
12. This concludes the configuration of ACM variables. Figure 6 shows the window after the configuration. Click the “Save” button at the lower right corner and a warning message box displays. Click the “Yes” button to ignore the message and a file named “VdV\_Reactor.json” is created in the working directory.
13. Click the “x” icon at the upper right corner of the window to exit the “SinterConfigGUI”. Close the ACM window too.
14. After the dynamic input and output variables are configured and the JSON file is created, start FOQUS GUI by double-clicking the shortcut icon on the desktop or from the “Start” menu. The main window of the FOQUS displays. Click the “Settings” icon and make sure the “SimSinter Home” is set to the full path of the folder (not a file inside the folder) where SinterConfigGUI.exe is located. Click the “DRM-Builder” icon and the embedded D-RM Builder frame displays as shown in Figure 7.
15. The embedded D-RM Builder widget is similar to a classical Microsoft Windows application with menus and a toolbar in the top part of the window, a gray area for displaying the input and output variables (initially blank), a white background client area in the middle, and a status bar at the bottom. The client area is used to display the messages. Browse the individual menus to see the submenus or commands. Some of the submenus and commands are shaded/unavailable. They are enabled later when the status of the D-RM building process is updated. Initially a blank project exists in the FOQUS framework and a user can start to work on the blank project by choosing a high-fidelity model (ACM model) configured in Steps 1-14 through the “Setup” menu. The “File” menu contains “New”, “Open”, “Close”, “Save”, “Save As”, and “Export” submenus. Add a new case to the main window by issuing the **File 🡪 New** command or clicking on the toolbar icon “new”. If the current project is not saved, the D-RM Builder will ask the user to save the project. Open an existing case by issuing the **File 🡪 Open** command or clicking the toolbar icon “open”. Again, the D-RM Builder will ask the user to save the project if the current project has been modified. Close the current case by issuing the **File 🡪 Close** command or the toolbar icon “close” and a blank project will be displayed. Save a case by issuing the **File 🡪 Save** command or clicking the toolbar icon “save”. Save a case to a different name by issuing the **File 🡪 Save As…** command. The typical workflow for building and testing a D-RM is from the “Setup” menu, to the “Build” menu, and then finally to the “Post-Process” menu (from left to right). Within each menu, the general workflow is from the top command (submenu) to the bottom command. The “File” menu handles typical file opening, saving, and exporting commands, which could be issued at any time. Depending on the status of the D-RM building process, the user may or may not export certain files under the **File 🡪 Export** command. The toolbar icons are arranged from left to right following the typical D-RM building workflow.
16. To start the D-RM building process for a blank case, select a high-fidelity model first. Select **Setup 🡪 Choose High-Fidelity Model…** or click the toolbar icon “high\_fidelity\_model” to issue the command. A “SimSinter Configuration File” window displays for file browsing and selection. Browse and select the “VdV\_Reactor.json” file created in Step 12. This loads the configuration file into the D-RM Builder. A text message confirming the file selection and loading displays in the client area of the D-RM Builder window.
17. The SinterConfigGUI allows a user to select the dynamic input and output variables. Further configuration among the selected input and output variables are required to setup the D-RM Builder project. To configure the selected ACM input variables, navigate to the **Setup 🡪 Configure Input Variables…** command or click the toolbar icon “config\_input”. The “Input Variable Dialog” window displays as shown in Figure 8 (drm\_input\_variable\_dlg).

Figure 8: Input Variable Dialog Window

The input variables are listed in the “Input Variable List” list box. Click each of the variables to see the description, unit, and default ranges (±10%) used for conducting training/validation. The “Name” and “Unit” text boxes are for read-only. If the input variable is dimensionless, a space is displayed in the “Unit” text box. The user can edit the description as desired. This description is used for the labeling of future plots. For each input, define whether or not it is included in the D-RM. If excluded, the input variable is assumed to be constant and does not change with time. This can be useful for cases where the user wants to use an input as an unmodeled/unknown disturbance or where an input is a fixed parameter. In this example, the second input (V\_Reactor) represents the volume of the reactor and is never going to change in the course of the dynamic runs. Therefore, specify the input as a time-invariant input by clearing the “Varies With Time” check box (make sure the other two inputs are selected). The total number of variables with their “Varies With Time” check box selected is displayed in the read-only “Number of Time Dependent Inputs” text box. Another specification, the “Use Ramp To Replace Step Change” option, is not used in this example (keep this check box cleared for all three inputs). This option is used and discussed in the BFB example. Also notice that when the “Varies With Time” check box is cleared, the “Lower Limit”, “Upper Limit”, and “Ramp Rate” text boxes and the “Use Ramp To Replace Step Change” check box are disabled. When the “Varies With Time” check box is selected, the user can change the lower and upper limits of the variable to desired values. However, the lower limit cannot be larger than the steady-state value displayed at the read-only “Steady State” text box. Likewise, the upper limit cannot be smaller than the steady-state value. If this rule is broken, a warning message dialog box displays when the user clicks “OK” and the user cannot exit the dialog box unless the user clicks “Cancel”, which cancels all of the values and options that the user has entered on the Input Variable Dialog window. For this example, leave the default values of the lower and upper limits for the first two variables unchanged.

Depending on the time scale of the system response, the user may need to revise the sampling time interval, which is the time interval that is used in the discrete-time D-RM. The sampling time interval should be comparable with the time constants of the output variables in response to the change of input variables. This is discussed in other tutorials in this manual. The read-only “Solver Minimum Time Step” text box contains the minimum integration time step used by the ACM solver. The sampling time interval should be higher than the solver minimum time step. Leave the default sampling time interval of 0.01 hours unchanged. Click “OK” to accept the new configured values and close the “Input Variable Dialog” window. The messages in the client area confirm that the input variables are properly specified for the D-RM. Two arrows connected to a black box are plotted in the upper half of the D-RM Builder window, representing the two input variables.

1. Configure the output variables by issuing the **Setup 🡪 Configure Output Variables…** command or clicking the toolbar icon “config\_output”. The “Output Variable Dialog” window as shown in Figure 9 (drm\_output\_variable\_dlg) displays.

There are two output variables for the reactor. Select each output in the “Output Variable List” and then confirm the “Included in DRM” check box is selected. This means both output variables are predicted by the D-RM. Typically, include all of the output variables that change with time when any of the input variables changes. Remember that the user can select the variables of interest inside SinterConfigGUI. This window provides the user with another chance to eliminate any output variables that are not important. Click “OK” to accept the default values. A confirmation message displays in the client area of the main window. Two arrows connected to a the right side of black box are plotted in the upper half of the D-RM Builder window, representing the two output variables.

1. Once input and output variables are configured, the user is ready to prepare a training sequence, a series of step changes of input variables with defined holding durations that are simulated by the high-fidelity model. To prepare the training sequence, select the **Setup 🡪 Prepare Training Sequence…** command or click the toolbar icon “training\_sequence”. The “Step Change Sequence Dialog” window displays as shown in Figure 10.

Figure 10: Training Sequence Dialog Window

The sampling time interval and solver minimum time step are displayed in the first two read-only text boxes for reference. The D-RM Builder uses the Latin Hypercube Sampling (LHS) method to fill the steady-state input space for input variables. A sequence of step changes of the input variables, one at a time, is obtained by moving from one steady-state point to the next steady-state point in a set of LHS. For each LHS point set, specify the number of steady-state points to be sampled. For this example, enter 5 in the “Number of LHS Points” text box. Within each LHS point set, the holding time between the step changes or the duration of the step change is the same. The durations for different LHS point sets should be varied such that different frequencies in a range can be excited. Increasing the number of LHS point sets makes the step change sequence longer and the number of training data points larger, which generally leads to a more accurate D-RM. It takes a longer CPU time to perform the ACM dynamic model simulation. For this example, enter 3 in the “Number of LHS Sets” text box and then press “Enter”. The number of items in the list box in the “For Each LHS Set” section is updated. Three items are listed for this example. Confirm that the “Include Reverse Step Changes” check box is selected. This means, the order of the step changes are reversed after the forward path from the first steady-state point to the last is completed, which makes the length of the step change sequence doubled. For each LHS set, a holding duration is assigned. The duration is measured as the number of sampling time intervals. Click the “LHS Set 1” item in the “For Each LHS Set” list box and then enter 5 in the “Duration of Step Change” text box. Since the time interval is 0.01 hour in this case, the holding duration of 5 intervals corresponds to 0.05 hour of holding time. Select the “LHS Set 2” item in the list box and then enter 10 in the “Duration of Step Change” text box. Select the “LHS Set 3” item in the list box and then enter 15 in the “Duration of Step Change” text box. It is recommended to enter different durations for different LHS sets such that it covers a range of frequencies of interest. Click “OK” to accept the inputs. A confirmation message displays in the client area of the main window.

1. Prepare an input change sequence for validation by issuing the **Setup 🡪 Prepare Validation Sequence…** command or clicking the toolbar icon “validation\_sequence”. The “Validation Sequence Dialog” window displays as shown in Figure 11.

Figure 11: Validation Sequence Dialog Window

The validation sequence is used as inputs for the high-fidelity model to simulate another set of response to validate the D-RM to be generated. This command can be issued before or after the D-RM is generated. In this example, create the step change sequence before the D-RM model is generated. Enter 4 in the “Number of LHS Points” text box. Note: The number of LHS points is not necessarily the same as the value used for training. Enter 2 in the “Number of LHS Sets” text box. Clear the “Include Reverse Step Changes” check box. Usually the sequence for validation is shorter than that for training to reduce the CPU time required to perform the high-fidelity model simulation. Therefore, use fewer LHS sets and exclude the reverse step change. Select the “LHS Set 1” item in the list box and then enter 8 for the duration of the first LHS set. Then select the “LHS Set 2” item in the list box and then enter 12 for the duration of the second LHS set. Note: The duration value for validation should be in the range of the training durations specified in the previous step. Click “OK” to accept the specifications. A confirmation text message displays in the client area of the main window.

1. Save the file by issuing the **File 🡪 Save** or **File 🡪 Save As…** command or clicking the toolbar icon “save”. The “Save As” dialog window displays. Enter “case1” in the “File Name” text box and then click “OK”. A JSON file, “case1.json”, is written to the current working directory. The file contains the inputs that have been entered. Copy the case file to any folder on the computer, open it via the D-RM Builder, and continue the D-RM building work there. If the user saves the case after completing the high-fidelity model simulations, the results of the simulations are also saved in the case file. Likewise, if the user saves the case after a D-RM is generated, the data for the D-RM is also saved in the case file. The D-RM Builder also keeps track of the status of the building process and knows what stage the user is in. When the user opens an existing case file, the D-RM Builder brings the user to the stage that they left off at when the case was saved.
2. Launch the ACM high-fidelity model simulation for training by issuing the **Build 🡪 Perform Training Simulation** command or clicking the toolbar icon “run\_training”. This creates temporary files and a folder in the current working directory and brings up an ACM window. The ACM simulation starts immediately. The ACM iteration message can be viewed inside the ACM window if the ACM’s “Simulation Message” window is turned on. The mouse icon is switched from normal (an arrow) to busy (rotating circle). Wait without closing the ACM window. After the high-fidelity model simulation is completed and the ACM window is closed, a message box window displays confirming the successfully completion of the high-fidelity model simulation. The mouse icon becomes a normal arrow icon. Note: The D-RM Builder calls ConsoleSinter.exe to perform dynamic ACM simulation and, therefore, correct path for the “SimSinter Home” entry of the FOQUS “Settings” needs to be configured before using the D-RM Builder.
3. Launch the ACM high-fidelity model simulation for validation by issuing the **Build 🡪 Perform Validation Simulation** command or clicking the toolbar icon “run\_validation”. Note: The background color of toolbar buttons for training are red and those for validation are green. Wait until the ACM simulation is completed and the message box confirming the successful completion of the simulation displays.
4. Select DABNet as the D-RM model type by issuing the **Build 🡪 D-RM Model Type** **🡪** **DABNet** command and then confirming the “DABNet” pop-up submenu is checked as shown in Figure 13.

Figure 13: DABNet Submenu for Selecting a D-RM Model Type

Since the Decoupled A-B Net (DABNet) model is generated as the model type for this example, confirm the DABNet option is selected, which is the default option.

1. Build the DABNet model by issuing the **Build 🡪 Generate Reduced Model…** command or clicking the toolbar “build\_drm”. The “DABNet DRM Parameter Dialog” window displays as shown in Figure 14.

Figure 14: DABNet DRM Parameter Dialog Window

There is an “Output Variable List” list box and an “Input Variable List” list box in the upper left corner of the window. Each output variable has its own DABNet model. An array of the DABNet model is generated if multiple output variables exist. In this case, there are two output variables and, therefore, two DABNet models are generated. The model parameters for each DABNet model include number of Laguerre states and estimated pole value for each input, and the number of neurons in the hidden layer of the feed-forward neural network. Only one hidden layer is permitted in the current version of the D-RM Builder. Select an output variable in the “Output Variable List” list box and select the number of hidden neurons. Then for each input variable corresponding to a selected output or input/output pair, specify the number of Laguerre states and the pole value. The D-RM Builder enables the user to define a fixed pole value for each input/output pair or to ask the software to optimize the pole values for the individual inputs. Regarding the neural network training, there are two options available. The first option, the default option, is the back propagation (BP) method. The second option is Interior Point Optimization (IPOPT). The D-RM building process involves the generation of discrete-time process matrices A and B based Laguerre series and mapping the state space to the model outputs through a neural network, followed by a balanced realization of A and B matrices and mapping of balanced state space to the outputs through another neural network. Therefore, two neural network trainings are required for each output variable. Specify the training option and maximum number of training iterations for both Laguerre and balanced neural networks. The initial weights of ANN connections can also slightly affect the results of balanced realization. For this tutorial, use the default values for the model parameters and options by clicking “OK” to accept the defaults. The D-RM Builder starts the model generation process with multiple text messages displaying in the FOQUS console window. Read the messages and pay attention to the relative training errors reported, especially for the balanced model. A training error of less than (0.01%) indicates a good selection of DABNet model parameters. In this case the relative error of the DABNet model for the first output () is approximately (based on the balanced model) and that for the second output () is approximately . Note: The user may get different numbers depending on the version of ACM that is being used and the operating system the D-RM Builder is installed on.

1. Save the case by issuing the **File 🡪 Save** command or clicking the toolbar icon “save”. Since the dynamic reduced model has been generated, it is time to save the case setup, high-fidelity model simulation results, and the data of the generated D-RM to the case file.
2. To find how good the D-RM model prediction is compared to the high-fidelity model prediction, use the commands under the “Post-Process” menu. First confirm the “Use Balanced Model for Prediction” option under the “Post-Process” menu is selected. Usually this option is selected to examine the accuracy of the model. If this option is cleared, the unbalanced Laguerre models are used for D-RM predictions. Usually the unbalanced model predictions are not as accurate as the balanced model predictions, especially for the training sequence.
3. Examine the response for the training data. To predict the training response using the generated D-RM, issue the **Post-Process 🡪 Predict Training Response** command or click the toolbar icon “predict\_train”. The D-RM Builder calculates the response to the training sequence of the input changes based on the generated D-RM. This command should be completed very quickly since the calculations for the D-RM based responses of the output variables are very fast. A text message displays in the main window indicating the completion of the calculation.
4. With both ACM and D-RM predictions for the response of the training sequence available, issue the **Post-Process 🡪 Plot Training Responses…** command or click the toolbar icon “plot\_train” to visualize the results. The “Result Plotting Dialog” window as shown in Figure 15 displays.

Figure 15: Result Plotting Dialog Window

The Result Plotting Dialog window lists the input and output variables for the user to plot. Note: The variable names listed with an asterisk (“\*”) indicate the input or output variables that are varied (not fixed) in the reduced model. Click the items in the list boxes to select and deselect the variables. The four buttons below the two list boxes enable the user to select or deselect all of the input or output variables. Here the user can select the first and the third input variables and the first two output variables as shown in Figure 15. The D-RM Builder always plots the responses predicted by the generated D-RM and those by the ACM model in the first row of the figures. The user can also plot the relative errors, input step changes, and correlation points by selecting the corresponding check boxes in the lower part of the window. Accept the defaults with all of the check boxes selected and then click “OK”. A Matplotlib window named “Figure 1” displays in a separate window as shown in Figure 16.

Figure 16: Plots of Input and Output Data as Functions of Time for Training Data

Maximize the plot window to see the plots of the input and output variables clearly. The plots in the top row are the curves of the output variables versus time. If the user looks carefully, there are two curves plotted for each output variable. The one in black is the high-fidelity model (ACM model) simulation result and the other in red is the D-RM prediction. The red curve seems to sit on the top of the black one unless the user zooms in to a selected region. The second row contains the normalized error plots showing the relative errors of the D-RM predictions compared to the ACM simulation results. Notice that the relative errors are very small in this case, indicating that the generated D-RM fits the ACM model results very well. The third row contains the plots of step changes of individual input variables, one input variable on each plot. If the user looks carefully, the step changes of two inputs do not happen at the same time. Actually the two inputs take turns to make step changes, one at a time. The bottom row plots the points of the D-RM model predictions versus the ACM model predictions (plant data). If a D-RM model prediction is exactly the same as the ACM model prediction, the point sits exactly on the 45 degree line if the scales of the two axes are identical. The legends of the bottom row figures also show the R2 values of the correlation. As seen in Figure 16, the R2 (coefficient of determination) values for the two outputs are very close to 1, indicating that the D-RM is quite accurate. Note: R2 is always less than 1. Generally, the higher the R2 value, the more accurate the generated D-RM. Close the plot window by clicking the “x” at the upper right corner. If needed, issue the **Post-Process 🡪 Plot Training Responses…** command again with only one output variable selected at a time to visualize the plot in a single column. Likewise, clear some of the check boxes in the Result Plotting Dialog window shown in Figure 15 to remove some of the rows of figures so the user can see the larger figures of interest.

1. Examine the response for the validation data. Usually the generated D-RM can predict the response of the training sequence very close to that from the high-fidelity model simulation since the reduced model is trained using the training data. A good D-RM should also be able to predict the response of the validation data quite well. To predict the validation response using the generated D-RM, issue the **Post-Process 🡪 Predict Validation Response** command or click the toolbar icon “predict\_valid”.
2. Issue the **Post-Process 🡪 Plot Validation Responses** command or click the toolbar icon “plot\_valid”. The “Result Plotting Dialog” window as shown in Figure 15 displays. Select the first and the third input variables and the first two output variables and click “OK”. A Matplotlib window named “Figure 1” displays in a separate window as shown in Figure 17.

Figure 17: Plots of Input and Output Data as Functions of Time for Validation Data

Notice that the red curves are still on the top of the black curves for both the and outputs, indicating a quite accurate D-RM generated by the D-RM Builder. Close the plot window by clicking the “x” at the upper right corner.

1. Save the file by clicking the toolbar icon “save”. The case file contains the data predicted by the D-RM generated for both the training and validation sequences.
2. Perform the Uncertainty Quantification (UQ) analysis on the generated D-RM, using the ACM predictions as the plant data. Note: The UQ commands are available only when the DABNet model option is selected. For the UQ analysis, it is assumed that the process has certain noise in terms of standard deviation of state-space variables and the measurements of output variables also have noise in terms of standard deviations. During the UQ analysis, the measured output values are obtained from the ACM prediction with the noise term added based on the standard deviation provided by the user. To setup the noise levels of the state and output variables, issue the **UQ 🡪 Specify Noise…** command or click the toolbar icon “specify\_noise”. The “Noise Specification Dialog” window as shown in Figure 18 displays.

Figure 18: Process and Measurement Uncertainty Dialog Window

Specify the noise as the fraction of the standard deviation of the state variables in the training sequence. Note: Since the state variables could be either positive or negative in the normal operating conditions, the standard deviation of the state variables calculated in the training data is used as the basis for the noise level of the state variables. In the current version, the ratio of the noise to the standard deviation of the state variable is assumed to be identical for any state variables, which is not necessarily true for a real-world process. The identification of process noise has not been built in to the D-RM Builder. The user can specify a single number for the noise level. For this tutorial, use the default value of 0.02 in the “Fraction of Standard Deviation” text box in the “Process Noise for All State Variables” section. This means that the noise defined as the standard deviation of each state-space variable is 2% of the standard deviation of the variable itself found in the training sequence. The measurement noises can be specified separately, one for each output variable. These noises are related to the sensitivity of the sensors and the electronics used to measure the output variables. For this tutorial, select the first output variable “1. Cb\_Product” in the “Output List” and then enter 0.05 in the “Fraction of Standard Deviation” text box. Then, select the second variable “2. Ca\_Product” and then enter 0.05 as shown in Figure 18. Click “OK” to accept the noise specifications.

1. Indicate to the D-RM Builder to run the UQ analysis on the generated DABNet model following the validation step-change sequence. The algorithm of Unscented Kalman Filter (UKF) is used to predict the state variables step by step using the D-RM and update the state and output variables using the measurement data, which is calculated by adding the random noise to the ACM predictions. The mean values of the state and output variables are called “filtered” values, which are the weighted averages by considering both the D-RM predictions and measurements. The Unscented Transform (UT) calculations are performed to calculate the means and covariances for the non-linear functions involved. Multiple samples known as the sigma points around the means are used to evaluate the non-linear functions in the UT calculation. The UT approach is proved in the literature to be much faster than Monte Carlo sampling with reasonable accuracy. The covariance matrices are also updated step by step. The elements in the covariance matrices tend to reach their asymptotic values after a certain period of time along the step-change sequence. To run the UQ analysis, issue the **UQ 🡪 Analyze…** command. The “Result Plotting Dialog” window as shown in Figure 19 displays, asking the user to specify the options for plotting the results of the UQ analysis.

Figure 19: Result Plotting Dialog Window

Since the volume of the reactor is constant, the user can select only the first and the third inputs in the “Select Input Variables” section on the window. Select both output variables in the “Select Output Variables” section. This time clear the “Plot Input Step Changes” check box if the user is not interested in the input versus time plots. Click “OK”, the window closes, and the UQ calculation begins. The calculation usually takes a while to complete since UKF algorithm is quite time consuming. After the calculation is completed, the results are plotted in a pop-up window as show in Figure 20.

Figure 20: Plots Showing the Results of UQ Analysis for the Validation Sequence

Adjust the size of the plot window before clicking “OK” on the message window, which closes both the plot and the message windows. The first row of the plot window contains the two output variables (concentrations of Species A and B) as functions of time. The black curves show the measured values which are calculated based on the ACM predictions and the randomly imposed measurement noise. The red curves are the filtered mean output values. The blue and green dotted curves show the error ranges (+/- the standard deviation from the calculated covariance matrix). The relative errors and correlation plots are similar to those shown in Figure 17 except that both the D-RM and UKF results are plotted. Since the generated D-RM is very accurate, the standard deviations of the output variables are dominated by the noise of the measurements, which is 5% of the standard deviation of the output variable in the training data.

1. Save the case by clicking the toolbar icon “save”. Export the covariance matrices at the end of the validation sequence by issuing the **File 🡪 Export 🡪 Covariance Matrices…** command. The “Save Covariance Matrices” dialog window displays. Click “Save” to accept the default name for the file. The exported matrices are written in “csv” format.
2. Export the generated D-RM to a MATLAB script file by issuing the **File 🡪 Export 🡪 D-RM As Matlab Script File…** command or clicking the toolbar icon “save\_drm\_matlab”. The “Save D-RM As Matlab Script” dialog window displays. Use the default file name “case1.m” and then click “OK” to save the D-RM to the “case1.m” file. This file is the main output file of the D-RM Builder and can be used by a MATLAB programmer for dynamic simulations. The usage of this file is given in another tutorial in this manual.
3. Save the log messages that have been displayed in the client area of the main window by selecting the **File 🡪 Export 🡪 Messages to Log File…** command. The “Save Message Log” dialog window displays. Use the default file name “case1.log” to save the log file. The log file contains the information regarding the training errors and the optimized parameters during the reduced model building process, if applicable. The information can be examined later when comparing to a new case with new user inputs.
4. Save the actual input and output sequence data to a comma separated value (.csv) file by issuing the **File 🡪 Export 🡪 Training Data…** command and the **File 🡪 Export 🡪 Validation Data…** command for training data and validation data, respectively. The data contains the step changes of inputs versus time and the responses predicted by the high-fidelity model. If the D-RM predictions are available, they are also included in the exported file.
5. Close the main window of the D-RM Builder to complete the model generation process.