



Procedure for the Simulation of NOx Emissions Produced by Combustion Turbines

SIM-00001-Nox

Revision: 1

Status: Issued

July 2016

CONFIDENTIAL

© Copyright 2015 SimGenics LLC

Proprietary Class 2

CHANGE HISTORY

SimGenics-Simulation LLC

Suite B210, 25½ Road, Grand Junction, Colorado, 81505, United States of America

Tel: +1 (970) 639 2498, Fax: +1 (817) 736 0427

e-mail: mgregg@simgenics.com or flaubscher@simgenics.com



Configuration Control			
Created by:	Franli Laubscher		
Original creation Date:	01 July 2016		
Revision History			
Revision	Date	Author	Description of Changes
A	12-July-2016	Franli Laubscher	Draft issue
B	12-July-2016	Francois Laubscher	Review of Document
1	20-July-2016	Franli Laubscher	Updates

Authorization			
Action	Designate / Signature	Position	Date
Prepared	_____		
Reviewer	_____		
Approver	_____	Management Representative	

Original Signed Copy at Configuration Management Centre (CMC)



Contents

1. PURPOSE	5
2. SCOPE	5
3. REFERENCES	5
4. DEFINITIONS, TERMS, ACRONYMS AND ABBREVIATIONS	5
5. PROCEDURE	6
5.1 ADDING THE AMMONIA	6
5.2 CHANGING THE REACTION RATES FOR CORRECT NO _x EMISSIONS IN THE FLAME	7
5.2.1 ADJUSTING THE REACTION RATE VALUE	7
5.2.2 ADJUSTING THE REACTION RATE FACTOR	8
5.3 FUNCTION GENERATOR OF THE FOGGER	10
5.4 THE CHEMICAL REACTION USING AMMONIA TO REDUCE NITROGEN OXIDES	11
5.5 CALCULATING OMEGA (Ω)	14
6. VERIFICATION AND VALIDATION	15
6.1 TEST ANALYSIS AT STEADY STATE	15
6.2 TEST ANALYSIS AT STARTUP	16
6.3 TEST ANALYSIS AT SHUTDOWN	19
7. APPENDIX	21

Figures

Figure 5-1: Mixed Fluids Window	7
Figure 5-2: NO _x Reaction in Flame Model	8
Figure 5-3: Function Generator for the Reaction Rate Factor	9
Figure 5-4: NO Mass Fraction at Turbine Exit	9
Figure 5-5: Fogger Illustration	10
Figure 5-6: Function Generator of the Fogger	11
Figure 5-7: The Balanced Chemical Reaction to Reduce NO _x Emissions [3]	11
Figure 5-8: Mass Fractions from Turbine Exit and Ammonia Injection	12
Figure 5-9: Mole Rate for Turbine Exit and Ammonia Injection	12
Figure 5-10: Mole Rate of the Reaction	12
Figure 5-11: Added Efficiency to the Reaction Mole Rate	13
Figure 5-12: Total Mole Rate	13
Figure 5-13: Conversion from Mole Rate to Mass Flow Rate	13
Figure 5-14: Final Mass Fractions	14
Figure 5-15: Total Mass Fraction	14
Figure 5-16: NO _x and Ammonia Slip ppm Calculations	14
Figure 5-17: Water Injection and Gas Supply Mass Flow Rates	14
Figure 5-18: Calculation of Omega	15



Figure 6-1: NOx Emissions of the Long Beach Simulator at Startup.....	16
Figure 6-2: NOx Emissions of the Test Simulator at Startup	17
Figure 6-3: Omega Value for Long Beach - and the Test Simulator at Startup (Sensor Dynamics not Included).....	18
Figure 6-4: Ammonia Slip for Long Beach - and the Test Simulator at Startup.....	18
Figure 6-5: NOx Emissions of the Long Beach Simulator at Shutdown	19
Figure 6-6: NOx Emissions of the Test Simulator at Shutdown.....	20
Figure 6-7: Omega Value for Long Beach - and the Test Simulator at Shutdown	20
Figure 6-8: Ammonia Slip for Long Beach – and the Test Simulator at Shutdown	21

Tables

Table 6-1: Steady State NOx Emissions Analysis	15
--	----

Abbreviations and Acronyms

Abbreviations	Description
NOx	Nitrogen Oxides
SCR	Selective Catalytic Reduction
Ω	Omega
ppm	Parts per Million



1. PURPOSE

This document describes the formation and simulation of NOx emissions produced by combustion turbines, using the SimuPact Flame Model.

2. SCOPE

NOx emissions are formed by the combustion of nitrogen and oxygen in furnace flames, and rapidly increase when the flames reach temperatures above 1800°F [4]. These emissions are a big cause for concern as they can combine with other pollutants in the atmosphere to form O₃, or ground level ozone, as well as nitric acid, which is harmful to the environment.

In order to limit the amount of emission gases discharged, aqueous ammonia is diluted which, through a process called selective catalytic reduction (SCR), will break down the NO molecules to form by-products such as water and nitrogen gases.

3. REFERENCES

Reference	Title	Rev/Date
[1] NOx and SOx	Chemistry of NOx Formation	03/14/2016
[2] NRGLB	Long Beach Open Cycle Combustion Turbine Simulation	SAT Version
[3] NOx and SOx	NOx – Sources and Control Methods	NOx.ppt
[4] Alentecinc	The formation of NOx	
[5] Wikipedia	Ammonium Hydroxide	07/11/2016
[6] Wikipedia	Selective Catalytic Reduction	07/01/2016
[7] Wikipedia	Parts-per Notation	06/05/2016

4. DEFINITIONS, TERMS, ACRONYMS AND ABBREVIATIONS

Thermal NOx (nitrogen oxides)

A generic term used to describe mostly NO (nitric oxide) and NO₂ (nitrogen dioxide) that form when nitrogen and oxygen combusts at high temperatures. NO contributes to approximately 90-95% of the NOx emissions released into the atmosphere.

Fogger

A device that injects water vapor into the inlet air mixture, thus lowering the flame temperature and decreasing the NOx emissions.



Aqueous Ammonia	A solution of ammonia in water [5]. The fraction of ammonia needed for the reaction will depend on the efficiency of the catalyst bed.
Selective Catalytic Reduction (SCR)	A method of converting the NO emissions, with the support of a catalyst bed and the aqueous ammonia, into diatomic nitrogen (N ₂) and water (H ₂ O) [6].
Ammonia Slip	The unreacted ammonia that passes through the SCR bed [2].
Omega (Ω)	The ratio of the mass flow of the water injection flowing into the flame over the mass flow of the gas supply entering the flame [2].
Parts per million (ppm)	This is a unit that describes the concentration of small values with dimensionless quantities, e.g. mole fractions or mass fractions [7].

5. PROCEDURE

The following procedures can be followed after the entire flow diagram is setup with all the nodes and boundary conditions defined. To ensure the operations explained below will be successful, the first step is to confirm that the diagram is able to solve and produce answers in steady state.

5.1 ADDING THE AMMONIA

Depending on the quality of the inlet air, the injected water, and the supplied gas, the top five gases present in the mixture after combustion will be varying amounts of CO₂, H₂O, N₂, NO, and O₂. Although ammonia (NH₃) is not a product of the combustion process, this compound will have to be added to the list of mixed gases in order for one to be able to change the mass fraction on another node in the future. To add NH₃ to the list, complete the following steps:

- Under *Charts/Lookup Tables*, expand the *Flow Solver* and choose *Materials and Fluids*. For the purpose of this example, *Mixed Fluids* were used. Expand *Des Moines* and double click on *Mixed Gases*.
- A window, as shown in **Figure 5-1** on the next page, will appear. Under the *Pure Gas Mixture* tab, double click on *Matheson – NH₃*, and it will shift over to the window on the right. Click OK. Ammonia has now been added to all the nodes used in your diagram.

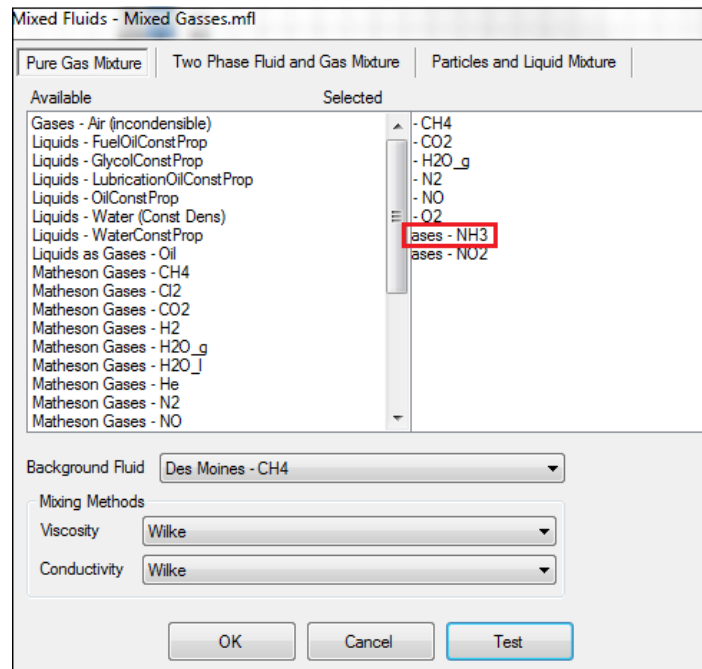


Figure 5-1: Mixed Fluids Window

5.2 CHANGING THE REACTION RATES FOR CORRECT NO_x EMISSIONS IN THE FLAME

In the flame model, there are four major chemical reactions that occur when combustion takes place. The quantity of the products formed after each reaction will vary according to the amounts that enter the combustion chamber. In order for the correct amount of NO_x emissions to be released, as specified by the plant data, the *reaction rate value* together with the *reaction rate factor* need to be adjusted accordingly.

5.2.1 ADJUSTING THE REACTION RATE VALUE

- Click on the flame model and scroll down to *Reaction Chain* (as circled in red). Below the reaction chain data, there will be four balanced chemical reactions. Scroll down until the reaction (also circled in red) as displayed in **Figure 5-2** on the next page, is found.



Flame	
Heat Balance	
dQ_flame_total	4.71879E-07 kW
NumT_flame_flameIterations	1
bBurnerIgnite	Yes
pReactionChain	Click here to edit...
[0]	
[1]	
[2]	
[3]	
Constants	
Reaction Data	O2 + N2 = 2NO Reaction Data (Reaction Data)
Inputs	
bIgnite	No
bAllowIgnition	No
Reaction Rate Limits	
eSpecify_Max_ReactionRate	Yes_as_value
dMax_ReactionRate_factor	1
dMax_ReactionRate_value	0.06135 mol/s
eSpecify_ReactionRate_factor	Yes
dReactionRate_factor	0.938943

Figure 5-2: NOx Reaction in Flame Model

- As mentioned before, both the chemical reaction's *reaction rate value* and *reaction rate factor* will have to be adjusted. The reaction rate value will limit the amount of NO moles that are produced every second. As seen in **Figure 5-2** above, the value (circled in blue) was changed until there were no more than 13.5 ppm NOx emissions present at the turbine exit.

5.2.2 ADJUSTING THE REACTION RATE FACTOR

The flame's temperature will increase if either the fogger, or the water injection, or both are to be switched off. It will be very difficult to change the *reaction rate value* manually every time there is a fluctuation in the temperature. Therefore, the *reaction rate factor* will act as a multiplier to the *reaction rate value*, and will be changed by means of a function generator. Follow the steps below to pair the correct *reaction rate factor* with the flame.

- Insert a function generator, from the *DCS Library*.
- Using a *Data Transfer Link*, connect the temperature of the flame to the input of the function generator.
- The function generator will interpolate linearly between the number of X-values and Y-values that are inserted. On the function generator listed below, **Figure 5-3**, there are six points inserted. The X-values correspond to the flame temperature (°C is used in this example), and the Y-values correspond to the *reaction rate factor*. The X-values should be inserted according to what the flame temperature is, if the fogger and water injection are switched on or shut off respectively. The corresponding Y-values should be changed in order to reflect what the plant data indicates the NOx emissions should be at these various temperatures.



General	
Identifier	CD-Function Generator-1982
Solving	<input checked="" type="checkbox"/>
Description	
Connectable Outputs	
Output	0.938943
Slope	0.00167188
Configuration	
Low Limit	0.4
High Limit	7
Limit to Range	Yes
Filter Time	5
Execution Number	0
Inputs	
Number of Points	6
X values	
[0]	500
[1]	700
[2]	952
[3]	984
[4]	1120
[5]	1151
Y values	
[0]	0.443
[1]	0.621
[2]	0.9386
[3]	0.9921
[4]	5.7562
[5]	6.2663
Connectable Inputs	
Input	952.205
Disable	0

Figure 5-3: Function Generator for the Reaction Rate Factor

- Set the high and low limits so that the Y-values fall in between these limitations (as depicted in blue in **Figure 5-3** above).
- Use a *Data Transfer Link* to connect output of the function generator to the *reaction rate factor* of the flame.
- To see how the quantity of the products change as the *reaction rate value* and the *reaction rate factor* change, the mass fractions at the turbine exit can be examined. **Figure 5-4** below, indicates that the mass fraction of NO can be converted to ppm if multiplied by 1×10^6 in order to equal 13.5 ppm.

Mass Fraction	
Mass fractions	
CH4	0
CO2	0.0542063
H2O_g	0.14427
N2	0.680848
NO	1.35008E-05
O2	0.120662
NH3	0
NO2	0

Figure 5-4: NO Mass Fraction at Turbine Exit



5.3 FUNCTION GENERATOR OF THE FOGGER

The fogger's influence on the temperature of the flame is relatively small, but it is still an important component that must be included in the design. In order to maximize the fogger's influence follow the steps listed below:

- Insert a function generator, from the *DCS Library*.
- Using a *Data Transfer Link*, connect the opening of the valve to the input of the function generator as shown in **Figure 5-5**.
- In this instance, there are 3 points inserted on the function generator. The X-values will correspond to the opening of the valve and the Y-values will correspond to the amount of heat lost (AU) in the node. The air inlet entering the node is 23.5°C, which the fogger has to reduce to 17.5°C if the valve is fully open. Insert different Y-values until this desired temperature is reached at the node. See **Figure 5-6**.

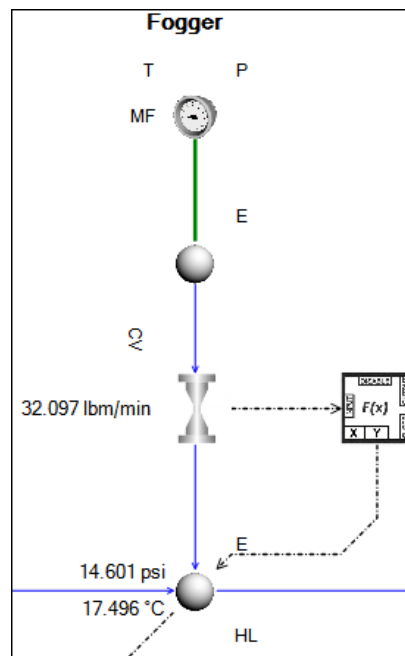


Figure 5-5: Fogger Illustration



General	
Identifier	01-NOX-Function Generator - 33
Solving	<input checked="" type="checkbox"/>
Description	
Connectable Outputs	
Output	116
Slope	132
Configuration	
Low Limit	0
High Limit	150
Limit to Range	No
Filter Time	5
Execution Number	0
Inputs	
Number of Points	3
<input checked="" type="checkbox"/> X values	
[0]	0
[1]	0.5
[2]	1
<input checked="" type="checkbox"/> Y values	
[0]	0
[1]	50
[2]	116
Connectable Inputs	
Input	1
Disable	0

Figure 5-6: Function Generator of the Fogger

- Remember to set the high and low limits for the Y-values to ensure the function generator will operate optimally. The limits are shown in blue in **Figure 5-6** above.
- The ambient temperature on the node is also added here as 40°F to ensure that there will be a heat loss occurring to the atmosphere.
- Using a *Data Transfer Link*, connect the output of the function generator to the heat leak (AU) of the node as shown in **Figure 5-5** on the previous page.

5.4 THE CHEMICAL REACTION USING AMMONIA TO REDUCE NITROGEN OXIDES

The chemical reaction, as presented in **Figure 5-7**, involving the nitrogen oxides and diluted aqueous ammonia will be illustrated in a separate script (see Appendix for the completed code) where the final mass fractions will be calculated and transferred to a node representative of the SCR bed.

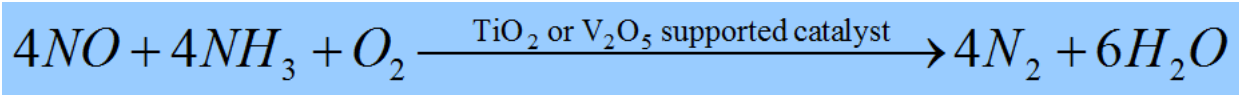


Figure 5-7: The Balanced Chemical Reaction to Reduce NO_x Emissions [3]

The following instructions will be a guide to calculating the correct NO_x emissions and Ammonia Slip.

- Insert a Script under the *Scripting* tab in *Components*.
- Create input properties for each of the mass fractions coming from the turbine exit node and the ammonia injection node respectively. Use the *Data Transfer Link* to connect the various mass fractions to these properties. See **Figure 5-8** below.



- Create two more input properties, also shown in **Figure 5-8**, to account for the mass flow rate at the exit of the turbine and the mass flow rate of the diluted ammonia. Use the *Data Transfer Tool* again to connect these properties. REMEMBER: To ensure all calculations are accurate, check that the units of the mass flow rates are in **kg/s** before connecting to the input properties.

Inputs (Mass Fractions)	
Mass_Fraction_CO2_Turbine	0.0542063
Mass_Fraction_H2Og_Turbine	0.14427
Mass_Fraction_N2_Turbine	0.680848
Mass_Fraction_NO_Turbine	1.35008E-05
Mass_Fraction_O2_Turbine	0.120662
Mass_Fraction_NH3_Turbine	0
Mass_Fraction_CO2_Ammonia	0.0510036
Mass_Fraction_H2Og_Ammonia	0.183604
Mass_Fraction_N2_Ammonia	0.640621
Mass_Fraction_NO_Ammonia	1.27031E-05
Mass_Fraction_O2_Ammonia	0.113533
Mass_Fraction_NH3_Ammonia	0.011226
Inputs [kg/s]	
MassFlow_Turbine	246.813
MassFlow_Ammonia	0.269239

Figure 5-8: Mass Fractions from Turbine Exit and Ammonia Injection

- Compute the mole rates [**mol/s**] for each individual gas present at the exit of the turbine and for the gases present in the diluted ammonia. To compute the mole rate, the mass fraction is multiplied by the mass flow rate, then divided by the molar mass of each gas. In order for the units to cancel out, it must be multiplied by a 1000. An example of the calculated mole rate, as it is written in the script, is expressed in **Figure 5-9** below.

```
//Mole Rate at Turbine Exit          n = [mol/s]
n_CO2_T = (Mass_Fraction_CO2_T*MassFlow_Turb*1000.0)/(44.01);

//Mole Rate at Ammonia Injection
n_CO2_A = (Mass_Fraction_CO2_A*MassFlow_Amm*1000.0)/(44.01);
```

Figure 5-9: Mole Rate for Turbine Exit and Ammonia Injection

- It will now be necessary to compute the mole rate [**mol/s**] of the reaction. Since the ammonia is diluted, it will act as the limiting reactant. For every mole of ammonia used, products will form in relation to what their coefficient, in the balanced chemical equation, is. Refer back to **Figure 5-7** as a guide to the formation of products and dissolution of reactants. **Figure 5-10** shows the reaction for every gas present in the chemical equation.

```
//Mole Rate of Reaction          n = [mol/s]
//4NO + 4NH3 + O2 ----> 4N2 + 6H2O
n_CO2_R = 0.0;
n_H2Og_R = (6.0/4.0)*n_NH3_Turbine + (6.0/4.0)*(n_NH3_Ammonia);
n_N2_R = n_NH3_Turbine + (n_NH3_Ammonia);
n_NO_R = n_NH3_Turbine + (n_NH3_Ammonia);
n_O2_R = (1.0/4.0)*(n_NH3_Turbine) + (1.0/4.0)*(n_NH3_Ammonia);
n_NH3_R = n_NH3_Turbine + (n_NH3_Ammonia);
```

Figure 5-10: Mole Rate of the Reaction



- It is important to notice that the ammonia will not react at 100% efficiency. There will be ammonia exiting the SCR bed, which is defined as the ammonia slip. Therefore, add the efficiency as an input property, and multiply it to the mole rate of the reaction as seen in **Figure 5-11**. The value for the efficiency can now be changed until the desired quantity of NOx emissions are produced.

```
//Mole Rate of Reaction                                n = [mol/s]
//4NO + 4NH3 + O2 ---> 4N2 + 6H2O
Ammonia_Fraction = Ammonia_Efficiency/100.0;
n_CO2_R = 0.0;
n_H2Og_R = (6.0/4.0)*n_NH3_Turbine + (6.0/4.0)*(n_NH3_Ammonia*Ammonia_Fraction);
n_N2_R = n_NH3_Turbine + (n_NH3_Ammonia*Ammonia_Fraction);
n_NO_R = n_NH3_Turbine + (n_NH3_Ammonia*Ammonia_Fraction);
n_O2_R = (1.0/4.0)*(n_NH3_Turbine) + (1.0/4.0)*(n_NH3_Ammonia*Ammonia_Fraction);
n_NH3_R = n_NH3_Turbine + (n_NH3_Ammonia*Ammonia_Fraction);
```

Figure 5-11: Added Efficiency to the Reaction Mole Rate

- The total mole rate [mol/s] can now be computed by adding the mole rate for each gas originating from the turbine exit and the mole rate for each gas originating from the ammonia injection. The mole rate of the reaction will be added or subtracted, depending whether products have been formed, or reactants have been used. The equation for each gas is demonstrated in **Figure 5-12** below.

```
//Total Mole Rate from Reaction                        [mol/s]
n_CO2_Total = n_CO2_Turbine + n_CO2_Ammonia + n_CO2_R;
n_H2Og_Total = n_H2Og_Turbine + n_H2Og_Ammonia + n_H2Og_R;
n_N2_Total = n_N2_Turbine + n_N2_Ammonia + n_N2_R;
n_NO_Total = n_NO_Turbine + n_NO_Ammonia - n_NO_R;
n_O2_Total = n_O2_Turbine + n_O2_Ammonia - n_O2_R;
n_NH3_Total = n_NH3_Turbine + n_NH3_Ammonia - n_NH3_R;
```

Figure 5-12: Total Mole Rate

- The total mole rate [mol/s] for every gas, now needs to be converted back to mass flow rate [kg/s] in order to be displayed as a mass fraction at the SCR bed. This is achieved by multiplying each mole rate with the gases' individual molar mass, and then dividing by a 1000 so that the units will be correct. The conversion from mole rate to mass flow rate is shown in **Figure 5-13**.

```
MassFlow_CO2 = (n_CO2_Total*44.01)/1000.0;
MassFlow_H2Og = (n_H2Og_Total*18.01528)/1000.0;
MassFlow_N2 = (n_N2_Total*28.0134)/1000.0;
MassFlow_NO = (n_NO_Total*30)/1000.0;
MassFlow_O2 = (n_O2_Total*31.998)/1000.0;
MassFlow_NH3 = (n_NH3_Total*17.03052)/1000.0;
```

Figure 5-13: Conversion from Mole Rate to Mass Flow Rate

- Compute the total mass flow rate [kg/s] by adding the mass flow rate at the exit of the turbine and the mass flow rate of the diluted ammonia together.
- The final mass fractions can now be calculated by taking the individual mass flow rate of every gas and dividing it by the total mass flow rate. Examine **Figure 5-14** on the next page to see the



calculations. REMEMBER: To prevent the denominator from being undefined when the total mass flow rate is zero, add a very small number (e.g. 1×10^{-6}) to the total mass flow rate before dividing.

```
Total_MassFlow = MassFlow_Turbine + MassFlow_Amonia;  
mf_CO2_out = MassFlow_CO2 / (Total_MassFlow + 0.0000001); //mf = Mass_Fraction  
mf_N2_out = MassFlow_N2 / (Total_MassFlow + 0.0000001);  
mf_NO_out = MassFlow_NO / (Total_MassFlow + 0.0000001);  
mf_O2_out = MassFlow_O2 / (Total_MassFlow + 0.0000001);  
mf_NH3_out = MassFlow_NH3 / (Total_MassFlow + 0.0000001);  
mf_H2Og_out = 1 - mf_CO2_out - mf_N2_out - mf_NO_out - mf_O2_out - mf_NH3_out;
```

Figure 5-14: Final Mass Fractions

- Before the mass fractions can be linked to the SCR bed, check that they add up to one (1). See **Figure 5-15** below.

```
Total_Mass_Frac = mf_CO2_out + mf_N2_out + mf_NO_out + mf_O2_out + mf_NH3_out + mf_H2Og_out;
```

Figure 5-15: Total Mass Fraction

- Finally, if the total mass fraction adds up to one, create output properties in the script in order to display the individual mass fractions for each gas to the outside world.
- Link the individual mass fractions for each gas to the boundary condition of the SCR node using the *Data Transfer Link*. The quantity of the NO_x emissions can now vary if the efficiency at which the ammonia reacts is changed.
- Calculate the NO_x emissions and ammonia slip, in ppm, by simply multiplying the NO and NH₃ mass fractions at the SCR bed by 1×10^6 respectively, as done in **Figure 5-16**.

```
//Calculations of parts per million [ppm]  
NOx_Amm = mf_NO_out * (1000000.0);  
Amm_Slip = mf_NH3_out * (1000000.0);
```

Figure 5-16: NO_x and Ammonia Slip ppm Calculations

5.5 CALCULATING OMEGA (Ω)

Omega is calculated as the ratio of the mass flow rate of the water injection flowing into the flame over the mass flow rate of the gas supply entering the flame. The following steps will require more data to be added to the existing script so that only a small calculation is needed to determine Omega.

- Create two new input properties in the script and connect them, using the *Data Transfer Link*, to the mass flow rate of the water injection and to the mass flow rate of the gas supply.
REMEMBER: Since Omega is a ratio, the units need to cancel out. Thus be sure to check that both properties are in the same unit. An example of the input properties are shown in **Figure 5-17**.

MassFlow_H2O_Injection	9.07983
MassFlow_Gas_Supply	5.17637

Figure 5-17: Water Injection and Gas Supply Mass Flow Rates



- Calculate Omega by dividing the mass flow rate of the water injection by the mass flow rate of the gas supply as in **Figure 5-18**. Remember to add the small value to the denominator in case the mass flow rate of the gas supply should equal zero.

```
//Calculation of Omega  
//Omega = Water Injection Mass Flow / Gas Supply Mass Flow  
Omega = MassFlow_H2O / (MassFlow_Gas_Supply +0.0000001);
```

Figure 5-18: Calculation of Omega

- Lastly, create an output property in the script in order to make Omega visible to the outside world.

6. VERIFICATION AND VALIDATION

To ensure that this new method for determining the NO_x emissions, as described in the previous sections, is accurate enough to use on simulators in the future, it has been compared to the NRG Long Beach simulator. The Long Beach simulator makes use of empirical methods to determine the NO_x emissions, as opposed to this new method that uses the gases' mass fractions and flow rates. An analysis of the emissions were done at startup, shutdown, and while the simulators were running at steady state.

6.1 TEST ANALYSIS AT STEADY STATE

At steady state (65 MW), the Long Beach simulator will produce the lowest quantity of NO_x emissions. At the turbine exit, the emissions should equal to 13.5 ppm. When ammonia has been added to the system, the NO_x emissions will reduce to 2.3 ppm. The following table, **Table 6-1**, compares the influence of the foggers, and the influence of the ammonia injection, on the presence of the NO_x emissions. The water injection will always be switched on.

Table 6-1: Steady State NO_x Emissions Analysis

	NRG Long Beach	New Method
Fogger ON Ammonia ON	2.3 ppm	2.299 ppm
Fogger ON Ammonia OFF	13.5 ppm	13.5 ppm
Fogger OFF Ammonia OFF	14.7 ppm	14.35 ppm
Fogger OFF Ammonia ON	2.4 ppm	2.91 ppm



	NRG Long Beach	New Method
Omega	1.76	1.75
Ammonia Slip with Fogger ON	0.1 ppm	5.874 ppm

It can be observed from **Table 6-1**, that the new method for determining the NO_x emissions, is very close to the values produced on the Long Beach simulator. Thus, it is possible that the new method can be considered for determining NO_x emissions on other simulators. The ammonia slip is the only property that shows a significant difference between the two methods. The power plant data [2] indicates that the ammonia slip needs to be around 5 ppm, when the NO_x emissions are 2 ppm. **It was noticed that the slip never changed on the Long Beach simulator, which proves that this value is incorrect.** Thus, it can be concluded that the ammonia slip determined from the new method would provide the more accurate number.

6.2 TEST ANALYSIS AT STARTUP

When starting up a simulator, it is expected that the NO_x emissions will have an increase in concentration, and then gradually reduce as steady state is reached. A study of the startup was done on both simulators in order to compare the accuracy of the new method to the values of the Long Beach simulator. **Figure 6-1** and **Figure 6-2** below indicates the differences and similarities in the two methods.

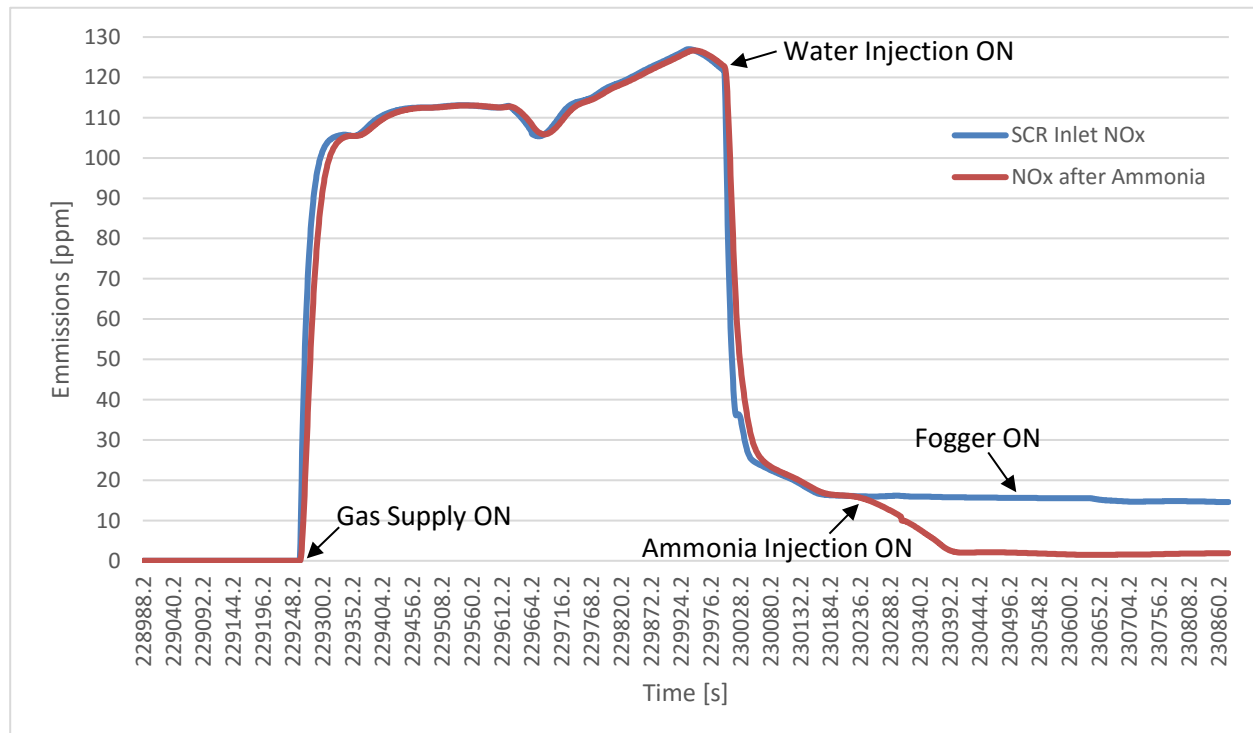


Figure 6-1: NO_x Emissions of the Long Beach Simulator at Startup

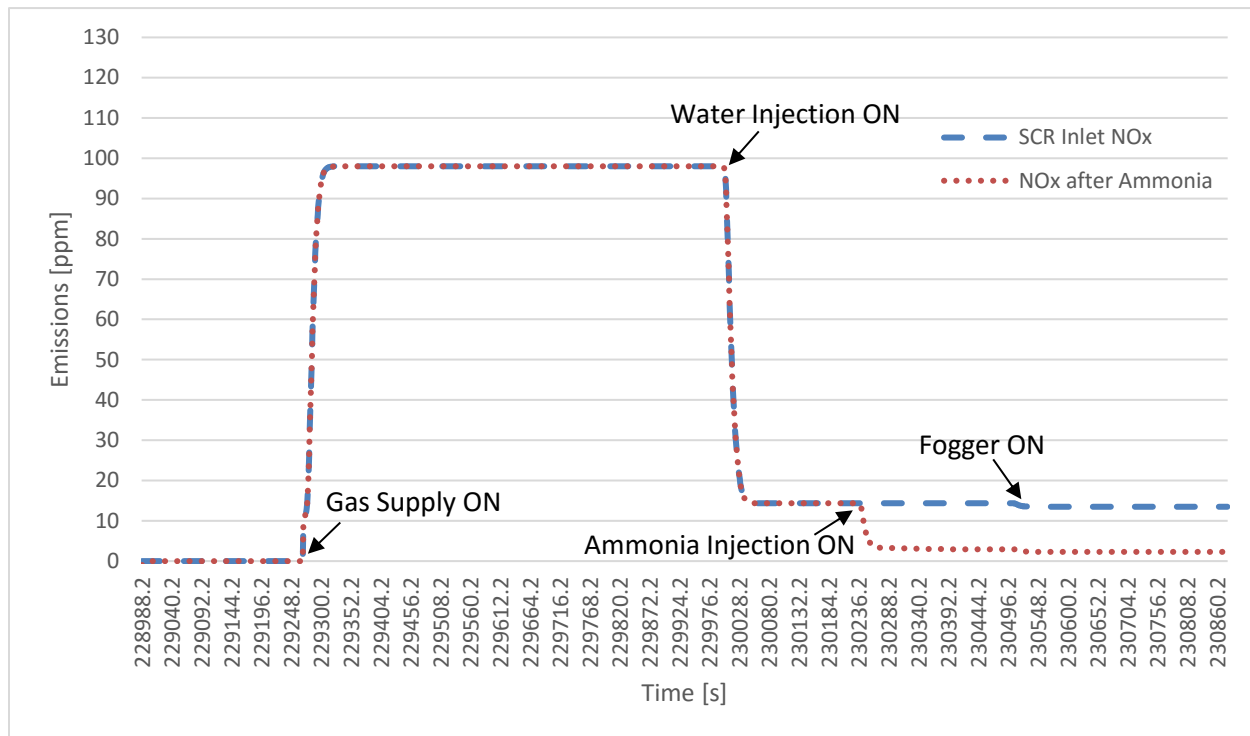


Figure 6-2: NOx Emissions of the Test Simulator at Startup

NOTE:

Sensor Dynamics was not included in the Test Simulator.

From the figures above, it can be observed that the NOx emissions of the test simulator do not reach the same maximum concentration than that of the NOx emissions at the Long Beach simulator. At the Long Beach simulator, the emissions are much higher because more gas is initially injected than is needed, in order to get the shaft speed up to 3600 rpm. The test simulator was tested at constant shaft speed and constant gas supply. Had the test simulator been tested at varying shaft speeds and varying gas supply, the results would have been similar to the values obtained at Long Beach. This is true because the final concentrations of the NOx emissions, as the simulator approached steady state, proved to be exactly the same.

As mentioned before, Omega fluctuates when the gas supply - and water injection mass flow rates' fluctuate. From **Figure 6-3**, on the next page, it can be seen that the Long Beach simulator's Omega value, increased and decreased significantly before it reached a stable value. This oscillation in the ratio occurred because the simulator attempted to increase the shaft speed up to 3600 rpm as fast as possible, which caused water injection and gas supply to be uncontrolled. There was no fluctuation in the test simulator's Omega, since the gas supply and water injection reached a constant value once the valves were opened. Apart from the oscillation, both simulators reached the same ratio as soon as the shaft speed reached a constant state.

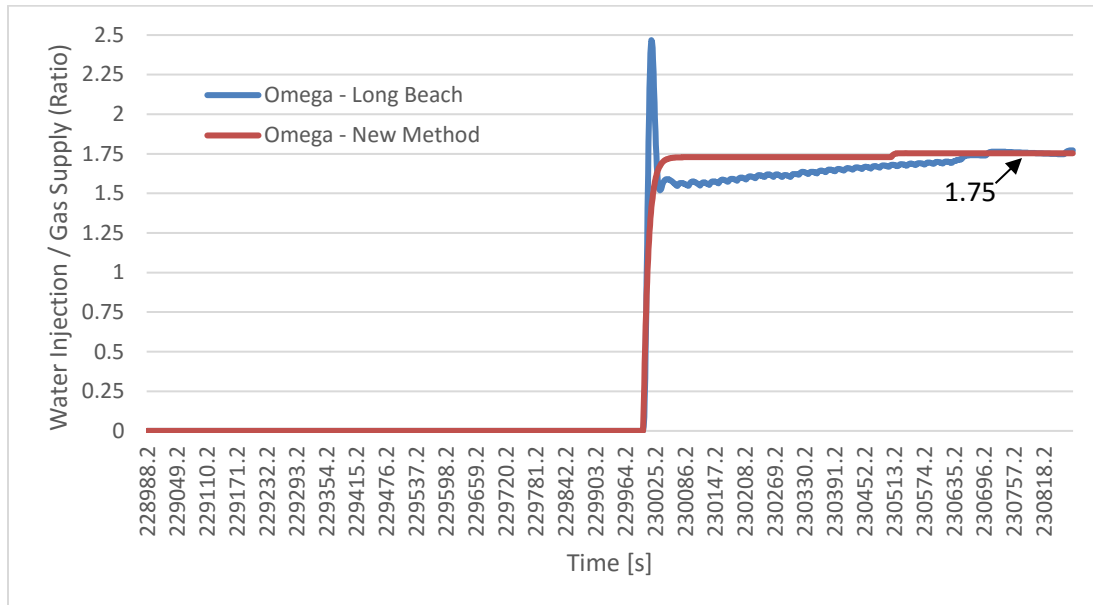


Figure 6-3: Omega Value for Long Beach - and the Test Simulator at Startup (Sensor Dynamics not Included)

It make sense for the ammonia slip to be present only when there is ammonia being injected into the system. From **Figure 6-4** below, it can be observed that the ammonia slip at Long Beach stays at a constant rate of 0.105 ppm throughout the entire startup, even when there is no ammonia present. This is incorrect. As mentioned before, the ammonia slip needs to be around 5 ppm when the simulator is at steady state, which close to the value obtained by the test simulator.

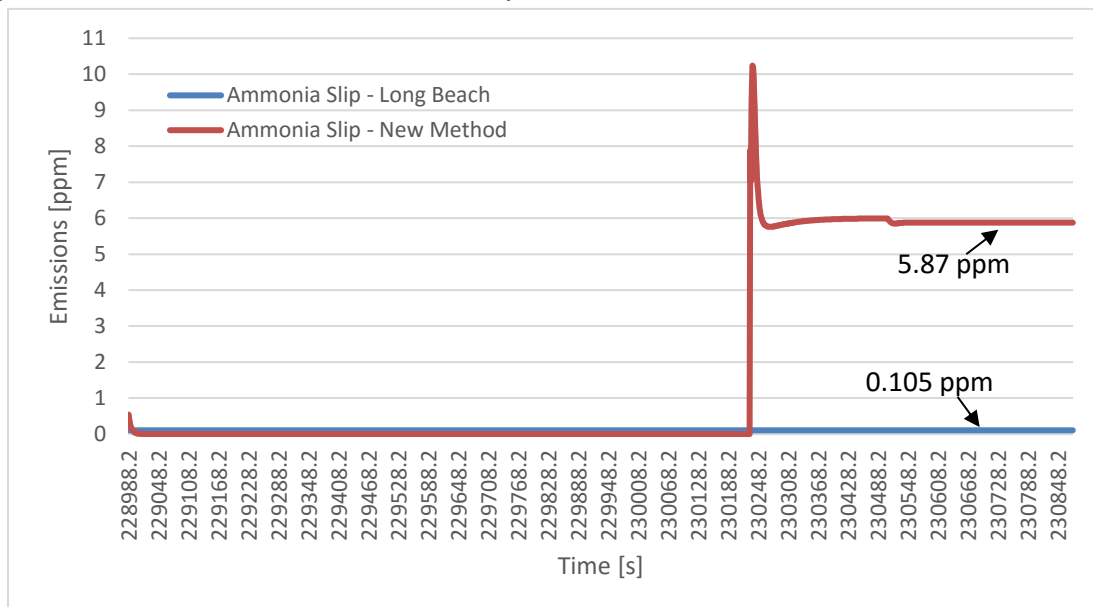


Figure 6-4: Ammonia Slip for Long Beach - and the Test Simulator at Startup



6.3 TEST ANALYSIS AT SHUTDOWN

The NO_x emissions released by the simulators, ranging from steady state to shutdown, can be examined in this section. The emissions will again increase in concentration as the ammonia injection, foggers, and water injection are switched off, but will ultimately decrease to zero once there is no more gas supplied to the combustion chamber. Look at **Figure 6-5** below, and **Figure 6-6** on the next page to compare the results between the Long Beach simulator and the test simulator.

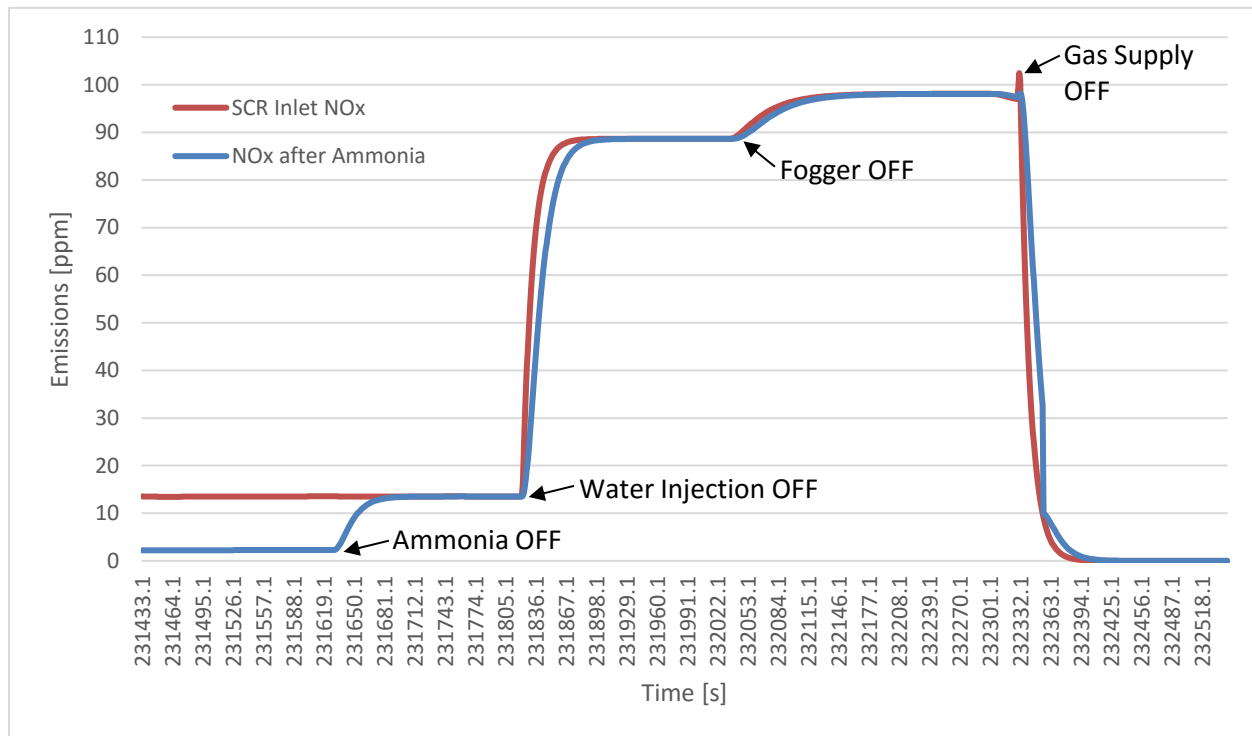


Figure 6-5: NO_x Emissions of the Long Beach Simulator at Shutdown

From these figures, it can be observed that the NO_x emissions on both simulators are very similar to each other when the ammonia injection, water injection, and foggers are switched off at the indicated time intervals. The concentration of the NO_x emissions never exceed 98 ppm on either simulator, and after the gas supply is switched off, both simulators decrease to 0 ppm. Since both simulators yielded the same results, it proves that the test simulator (containing the new method) is functioning correctly.

NOTE:

Sensor dynamics was not included in the Test Simulator graphs on the next page.

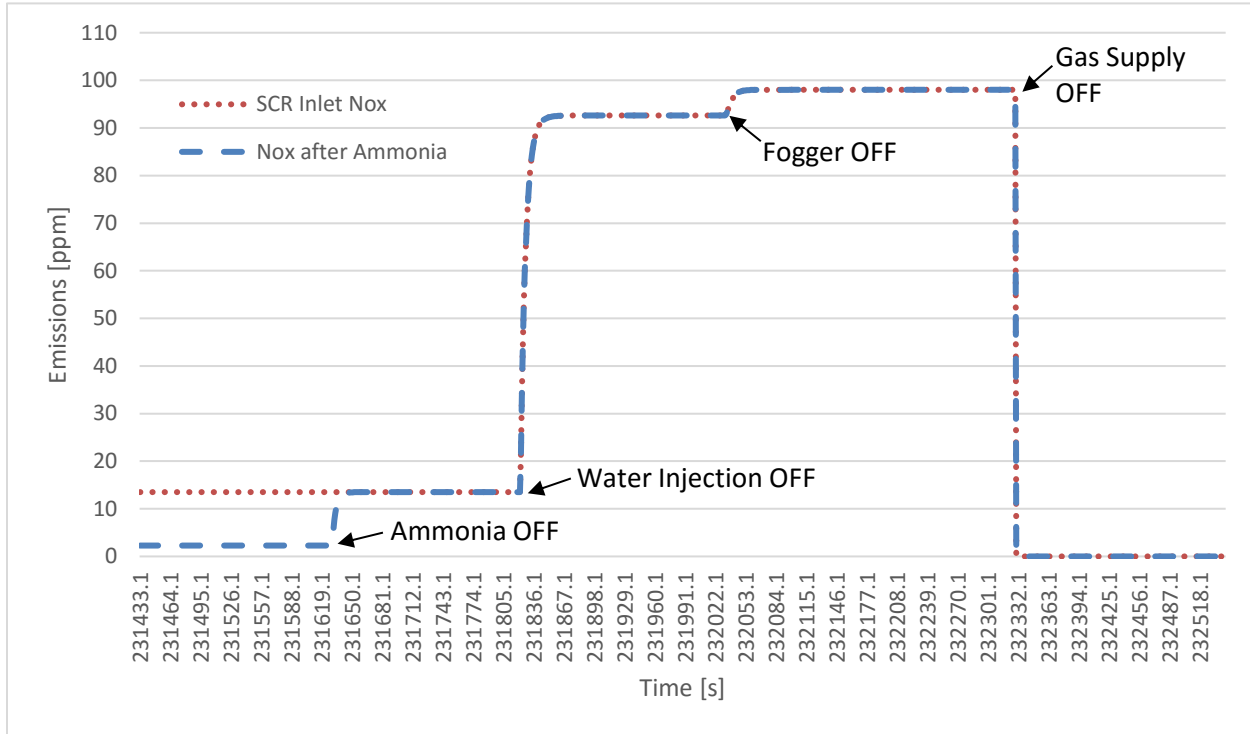


Figure 6-6: NOx Emissions of the Test Simulator at Shutdown

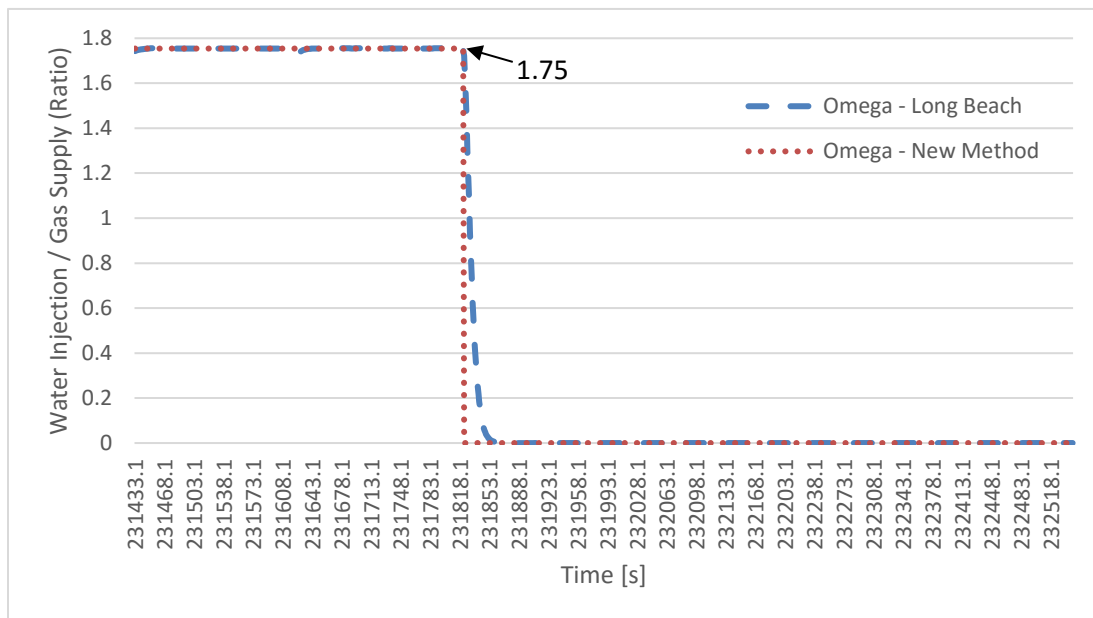


Figure 6-7: Omega Value for Long Beach - and the Test Simulator at Shutdown



The Omega value on both simulators were also examined at shutdown. It can be seen from **Figure 6-7** on the previous page, that Omega was immediately decreased to zero, on both simulators, when the water injection was turned off. From the figure, it seems that the Long Beach simulator takes a little longer to reach zero; this is only because there might be time constants inserted into the simulator preventing the calculation from happening too fast and making the simulator unstable.

The ammonia slip for each simulator was also calculated starting at steady state and ending when both simulators were completely shut down. As observed in **Figure 6-8** below, the ammonia slip at the Long Beach simulator was kept at a constant rate of 0.105 ppm, even after the ammonia injection had been shut off. This is proven to be incorrect by the test simulator, which demonstrates that the ammonia slip decreases from 5.87 ppm to 0 ppm at the moment the ammonia injection is turned off.

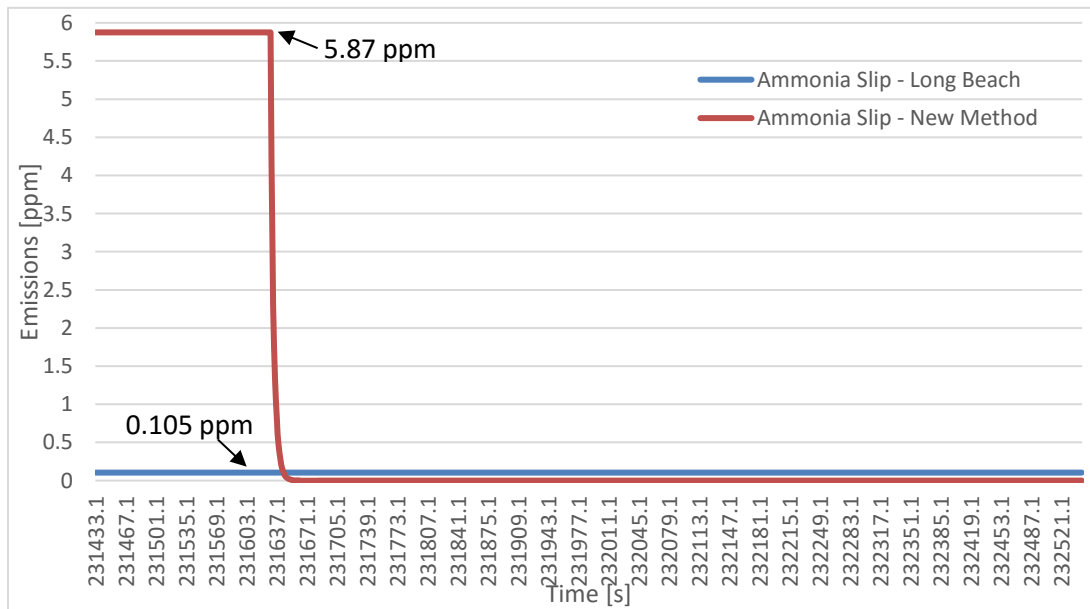


Figure 6-8: Ammonia Slip for Long Beach – and the Test Simulator at Shutdown

NOTE:

Sensor Dynamics was not included in the Test Simulator.

7. APPENDIX

The completed script, containing the code for solving the chemical equation using mass fractions and mass flow rates, as explained in **Section 5.4** above, is shown here to serve as an example for solving other chemical reactions.

Procedure for the Simulation of NOx
Emissions Produced by Combustion
Turbines



SIM-00001-Nox

```
1  //script using directives
2  //css_ref IPS.Core.dll;
3  //css_ref IPS.PluginInterface.dll;
4  using System;
5  using IPS.Properties;
6  using IPS.Scripting;
7
8  //script must be derived from IComponentScript
9  public class Script: IPS.Scripting.IComponentScript
10 {
11     //Inputs
12     IPS.Properties.Double _Mass_Fraction_CO2_Turbine;           //Mass Fractions
13     IPS.Properties.Double _Mass_Fraction_H2Og_Turbine;         //T = At turbine exit
14     IPS.Properties.Double _Mass_Fraction_N2_Turbine;
15     IPS.Properties.Double _Mass_Fraction_NO_Turbine;
16     IPS.Properties.Double _Mass_Fraction_O2_Turbine;
17     IPS.Properties.Double _Mass_Fraction_NH3_Turbine;
18
19     IPS.Properties.Double _Mass_Fraction_CO2_Ammonia;           //A = Ammonia Injection
20     IPS.Properties.Double _Mass_Fraction_H2Og_Ammonia;
21     IPS.Properties.Double _Mass_Fraction_N2_Ammonia;
22     IPS.Properties.Double _Mass_Fraction_NO_Ammonia;
23     IPS.Properties.Double _Mass_Fraction_O2_Ammonia;
24     IPS.Properties.Double _Mass_Fraction_NH3_Ammonia;
25
26     IPS.Properties.Double _MassFlow_Turbine;    //[kg/s]
27     IPS.Properties.Double _MassFlow_Ammonia;
28     IPS.Properties.Double _MassFlow_H2O_Injection;
29     IPS.Properties.Double _MassFlow_Gas_Supply;
30     IPS.Properties.Double _Ammonia_Efficiency;  //[%]
31
32     //Variables
33     IPS.Properties.Double _n_CO2_T;            //[mol/s]
34     IPS.Properties.Double _n_H2Og_T;
35     IPS.Properties.Double _n_N2_T;
36     IPS.Properties.Double _n_NO_T;
37     IPS.Properties.Double _n_O2_T;
38     IPS.Properties.Double _n_NH3_T;
39
40     IPS.Properties.Double _n_CO2_A;            //[mol/s]
41     IPS.Properties.Double _n_H2Og_A;
42     IPS.Properties.Double _n_N2_A;
43     IPS.Properties.Double _n_NO_A;
44     IPS.Properties.Double _n_O2_A;
45     IPS.Properties.Double _n_NH3_A;
46
47     IPS.Properties.Double _n_CO2_R;            //[mol/s]
48     IPS.Properties.Double _n_H2Og_R;
49     IPS.Properties.Double _n_N2_R;
50     IPS.Properties.Double _n_NO_R;
51     IPS.Properties.Double _n_O2_R;
52     IPS.Properties.Double _n_NH3_R;
53
54     IPS.Properties.Double _n_CO2_Total;        //[mol/s]
55     IPS.Properties.Double _n_H2Og_Total;
56     IPS.Properties.Double _n_N2_Total;
57     IPS.Properties.Double _n_NO_Total;
58     IPS.Properties.Double _n_O2_Total;
59     IPS.Properties.Double _n_NH3_Total;
```

Procedure for the Simulation of NOx
Emissions Produced by Combustion
Turbines



SIM-00001-Nox

```
60
61     IPS.Properties.Double _MassFlow_CO2;           //[kg/s]
62     IPS.Properties.Double _MassFlow_H2Og;
63     IPS.Properties.Double _MassFlow_N2;
64     IPS.Properties.Double _MassFlow_NO;
65     IPS.Properties.Double _MassFlow_O2;
66     IPS.Properties.Double _MassFlow_NH3;
67
68     IPS.Properties.Double _Ammonia_Fraction;
69
70     //Outputs
71     IPS.Properties.Double _Mass_Fraction_CO2_Out;   //Mass Fractions
72     IPS.Properties.Double _Mass_Fraction_H2Og_Out;
73     IPS.Properties.Double _Mass_Fraction_N2_Out;
74     IPS.Properties.Double _Mass_Fraction_NO_Out;
75     IPS.Properties.Double _Mass_Fraction_O2_Out;
76     IPS.Properties.Double _Mass_Fraction_NH3_Out;
77
78     IPS.Properties.Double _Total_MassFlow;
79     IPS.Properties.Double _Total_Mass_Fraction;
80     IPS.Properties.Double _NOx_at_SCR_Inlet;
81     IPS.Properties.Double _NOx_after_Ammonia;
82     IPS.Properties.Double _Ammonia_Slip;
83     IPS.Properties.Double _Omega;
84
85     //do pre simulation initialisation here
86     public override void Initialise()
87     {
88     }
89
90     //do post simulation cleanup here
91     public override void Cleanup()
92     {
93     }
94
95     //script main execution function - called every cycle
96     public override void Execute(double Time)
97     {
98         //Local Variables
99         double mf_CO2_T, mf_H2Og_T, mf_N2_T, mf_NO_T, mf_O2_T, mf_NH3_T; //Inputs
100        double mf_CO2_A, mf_H2Og_A, mf_N2_A, mf_NO_A, mf_O2_A, mf_NH3_A; //Inputs
101        double MF_Turb, MF_Amm, Amm_Eff, MF_H2O, MF_GS; //Inputs
102        double n_CO2_T, n_H2Og_T, n_N2_T, n_NO_T, n_O2_T, n_NH3_T; //Variables
103        double n_CO2_A, n_H2Og_A, n_N2_A, n_NO_A, n_O2_A, n_NH3_A; //Variables
104        double n_CO2_R, n_H2Og_R, n_N2_R, n_NO_R, n_O2_R, n_NH3_R; //Variables
105        double n_CO2_Total, n_H2Og_Total, n_N2_Total, n_NO_Total, n_O2_Total, n_NH3_Total; //Variables
106        double MF_CO2, MF_H2Og, MF_N2, MF_NO, MF_O2, MF_NH3, Amm_Frac; //Variables
107        double mf_CO2_out, mf_H2Og_out, mf_N2_out, mf_NO_out, mf_O2_out, mf_NH3_out; //Outputs
108        double Total_MF, Total_Mass_Frac, NOx_SCR, NOx_Amm, Amm_Slip, Omega; //Outputs
109
110        //Inputs
111        mf_CO2_T = _Mass_Fraction_CO2_Turbine.Value;
112        mf_H2Og_T = _Mass_Fraction_H2Og_Turbine.Value;
113        mf_N2_T = _Mass_Fraction_N2_Turbine.Value;
114        mf_NO_T = _Mass_Fraction_NO_Turbine.Value;
115        mf_O2_T = _Mass_Fraction_O2_Turbine.Value;
116        mf_NH3_T = _Mass_Fraction_NH3_Turbine.Value;
117
118        mf_CO2_A = _Mass_Fraction_CO2_Ammonia.Value;
```


Procedure for the Simulation of NOx
Emissions Produced by Combustion
Turbines

SIM-00001-Nox



```
119 mf_H2Og_A = _Mass_Fraction_H2Og_Ammonia.Value;
120 mf_N2_A = _Mass_Fraction_N2_Ammonia.Value;
121 mf_NO_A = _Mass_Fraction_NO_Ammonia.Value;
122 mf_O2_A = _Mass_Fraction_O2_Ammonia.Value;
123 mf_NH3_A = _Mass_Fraction_NH3_Ammonia.Value;
124
125 MF_Turb = _MassFlow_Turbine.Value;
126 MF_Amm = _MassFlow_Ammonia.Value;
127 Amm_Eff = _Ammonia_Efficiency.Value;
128 MF_H2O = _MassFlow_H2O_Injection.Value;
129 MF_GS = _MassFlow_Gas_Supply.Value;
130
131 //Calculates the mass fractions of the remaining products after ammonia had been added
132
133 //Mole Rate at Turbine Exit n = [mol/s]
134 n_CO2_T = (mf_CO2_T*MF_Turb*1000.0)/(44.01);
135 n_H2Og_T = (mf_H2Og_T*MF_Turb*1000.0)/(18.01528);
136 n_N2_T = (mf_N2_T*MF_Turb*1000.0)/(28.0134);
137 n_NO_T = (mf_NO_T*MF_Turb*1000.0)/(30.0);
138 n_O2_T = (mf_O2_T*MF_Turb*1000.0)/(31.998);
139 n_NH3_T = (mf_NH3_T*MF_Turb*1000.0)/(17.03052);
140
141 //Mole Rate at Ammonia Injection
142 n_CO2_A = (mf_CO2_A*MF_Amm*1000.0)/(44.01);
143 n_H2Og_A = (mf_H2Og_A*MF_Amm*1000.0)/(18.01528);
144 n_N2_A = (mf_N2_A*MF_Amm*1000.0)/(28.0134);
145 n_NO_A = (mf_NO_A*MF_Amm*1000.0)/(30.0);
146 n_O2_A = (mf_O2_A*MF_Amm*1000.0)/(31.998);
147 n_NH3_A = (mf_NH3_A*MF_Amm*1000.0)/(17.03052);
148
149 //Mole Rate of Reaction n = [mol/s]
150 //4NO + 4NH3 + O2 ---> 4N2 + 6H2O
151 Amm_Frac = Amm_Eff/100.0;
152 n_CO2_R = 0.0;
153 n_H2Og_R = (6.0/4.0)*n_NH3_T + (6.0/4.0)*(n_NH3_A*Amm_Frac); //Pos
154 n_N2_R = n_NH3_T + (n_NH3_A*Amm_Frac); //Pos
155 n_NO_R = n_NH3_T + (n_NH3_A*Amm_Frac); //Neg
156 n_O2_R = (1.0/4.0)*(n_NH3_T) + (1.0/4.0)*(n_NH3_A*Amm_Frac); //Neg
157 n_NH3_R = n_NH3_T + (n_NH3_A*Amm_Frac); //Neg
158
159 //Total Mole Rate from Reaction
160 n_CO2_Total = n_CO2_T + n_CO2_A + n_CO2_R; // [mol/s]
161 n_H2Og_Total = n_H2Og_T + n_H2Og_A + n_H2Og_R;
162 n_N2_Total = n_N2_T + n_N2_A + n_N2_R;
163 n_NO_Total = n_NO_T + n_NO_A - n_NO_R;
164 n_O2_Total = n_O2_T + n_O2_A - n_O2_R;
165 n_NH3_Total = n_NH3_T + n_NH3_A - n_NH3_R;
166
167 MF_CO2 = (n_CO2_Total*44.01)/1000.0; //MF = Mass Flow [kg/s]
168 MF_H2Og = (n_H2Og_Total*18.01528)/1000.0;
169 MF_N2 = (n_N2_Total*28.0134)/1000.0;
170 MF_NO = (n_NO_Total*30)/1000.0;
171 MF_O2 = (n_O2_Total*31.998)/1000.0;
172 MF_NH3 = (n_NH3_Total*17.03052)/1000.0;
173
174 Total_MF = MF_Turb + MF_Amm;
175 mf_CO2_out = MF_CO2/(Total_MF +0.0000001); //mf = Mass_Fraction
176 mf_N2_out = MF_N2/(Total_MF +0.0000001);
177 mf_NO_out = MF_NO/(Total_MF +0.0000001);
```


Procedure for the Simulation of NOx
Emissions Produced by Combustion
Turbines



SIM-00001-Nox

```
178 mf_O2_out = MF_O2 / (Total_MF + 0.0000001);
179 mf_NH3_out = MF_NH3 / (Total_MF + 0.0000001);
180 mf_H2Og_out = 1 - mf_CO2_out - mf_N2_out - mf_NO_out - mf_O2_out - mf_NH3_out;
181
182 Total_Mass_Frac = mf_CO2_out + mf_N2_out + mf_NO_out + mf_O2_out + mf_NH3_out + mf_H2Og_out;
183
184 //Calculations of parts per million
185 NOx_SCR = mf_NO_T * (1000000.0); // [ppm]
186 NOx_Amm = mf_NO_out * (1000000.0);
187 Amm_Slip = mf_NH3_out * (1000000.0);
188
189 //Calculations of parts per million [ppm]
190 NOx_Amm = mf_NO_out * (1000000.0);
191 Amm_Slip = mf_NH3_out * (1000000.0);
192
193 //Calculation of Omega
194 //Omega = Water Injection Mass Flow / Gas Supply Mass Flow
195 Omega = MF_H2O / (MF_GS + 0.0000001);
196
197 //Global Variables
198 _MassFlow_CO2.Value = MF_CO2;
199 _MassFlow_H2Og.Value = MF_H2Og;
200 _MassFlow_N2.Value = MF_N2;
201 _MassFlow_NO.Value = MF_NO;
202 _MassFlow_O2.Value = MF_O2;
203 _MassFlow_NH3.Value = MF_NH3;
204
205 _n_CO2_T.Value = n_CO2_T;
206 _n_H2Og_T.Value = n_H2Og_T;
207 _n_N2_T.Value = n_N2_T;
208 _n_NO_T.Value = n_NO_T;
209 _n_O2_T.Value = n_O2_T;
210 _n_NH3_T.Value = n_NH3_T;
211
212 _n_CO2_A.Value = n_CO2_A;
213 _n_H2Og_A.Value = n_H2Og_A;
214 _n_N2_A.Value = n_N2_A;
215 _n_NO_A.Value = n_NO_A;
216 _n_O2_A.Value = n_O2_A;
217 _n_NH3_A.Value = n_NH3_A;
218
219 _n_CO2_R.Value = n_CO2_R;
220 _n_H2Og_R.Value = n_H2Og_R;
221 _n_N2_R.Value = n_N2_R;
222 _n_NO_R.Value = n_NO_R;
223 _n_O2_R.Value = n_O2_R;
224 _n_NH3_R.Value = n_NH3_R;
225
226 _n_CO2_Total.Value = n_CO2_Total;
227 _n_H2Og_Total.Value = n_H2Og_Total;
228 _n_N2_Total.Value = n_N2_Total;
229 _n_NO_Total.Value = n_NO_Total;
230 _n_O2_Total.Value = n_O2_Total;
231 _n_NH3_Total.Value = n_NH3_Total;
232
233 _Ammonia_Fraction.Value = Amm_Frac;
234
235 //Outputs
236 _Mass_Fraction_CO2_Out.Value = mf_CO2_out;
```

Procedure for the Simulation of NOx
Emissions Produced by Combustion
Turbines



SIM-00001-Nox

```
237     _Mass_Fraction_H2Og_Out.Value = mf_H2Og_out;
238     _Mass_Fraction_N2_Out.Value = mf_N2_out;
239     _Mass_Fraction_NO_Out.Value = mf_NO_out;
240     _Mass_Fraction_O2_Out.Value = mf_O2_out;
241     _Mass_Fraction_NH3_Out.Value = mf_NH3_out;
242
243     _Total_MassFlow.Value = Total_MF;
244     _Total_Mass_Fraction.Value = Total_Mass_Frac;
245     _NOx_at_SCR_Inlet.Value = NOx_SCR;
246     _NOx_after_Ammonia.Value = NOx_Amm;
247     _Ammonia_Slip.Value = Amm_Slip;
248     _Omega.Value = Omega;
249
250 }
251
252 //any processing you want to do before steady state
253 public override void ExecuteBeforeSteadyState()
254 {
255     Execute(0.0);
256 }
257 //any processing you want to do while solving steady state
258 public override void ExecuteSteadyState()
259 {
260     Execute(0.0);
261 }
262
263 //any processing you want to do after steady state
264 public override void ExecuteAfterSteadyState()
265 {
266     Execute(0.0);
267 }
268
269 //constructor initialises parameters
270 public Script()
271 {
272     //Inputs
273     _Mass_Fraction_CO2_Turbine = new IPS.Properties.Double();
274     _Mass_Fraction_CO2_Turbine.Value = 0.0;
275     _Mass_Fraction_H2Og_Turbine = new IPS.Properties.Double();
276     _Mass_Fraction_H2Og_Turbine.Value = 0.0;
277     _Mass_Fraction_N2_Turbine = new IPS.Properties.Double();
278     _Mass_Fraction_N2_Turbine.Value = 0.0;
279     _Mass_Fraction_NO_Turbine = new IPS.Properties.Double();
280     _Mass_Fraction_NO_Turbine.Value = 0.0;
281     _Mass_Fraction_O2_Turbine = new IPS.Properties.Double();
282     _Mass_Fraction_O2_Turbine.Value = 0.0;
283     _Mass_Fraction_NH3_Turbine = new IPS.Properties.Double();
284     _Mass_Fraction_NH3_Turbine.Value = 0.0;
285
286     _Mass_Fraction_CO2_Ammonia = new IPS.Properties.Double();
287     _Mass_Fraction_CO2_Ammonia.Value = 0.0;
288     _Mass_Fraction_H2Og_Ammonia = new IPS.Properties.Double();
289     _Mass_Fraction_H2Og_Ammonia.Value = 0.0;
290     _Mass_Fraction_N2_Ammonia = new IPS.Properties.Double();
291     _Mass_Fraction_N2_Ammonia.Value = 0.0;
292     _Mass_Fraction_NO_Ammonia = new IPS.Properties.Double();
293     _Mass_Fraction_NO_Ammonia.Value = 0.0;
294     _Mass_Fraction_O2_Ammonia = new IPS.Properties.Double();
295     _Mass_Fraction_O2_Ammonia.Value = 0.0;
```

Procedure for the Simulation of NOx
Emissions Produced by Combustion
Turbines



SIM-00001-Nox

```
296      _Mass_Fraction_NH3_Ammonia = new IPS.Properties.Double();
297      _Mass_Fraction_NH3_Ammonia.Value = 0.0;
298
299      _MassFlow_Turbine = new IPS.Properties.Double();
300      _MassFlow_Turbine.Value = 0.0;
301      _MassFlow_Ammonia = new IPS.Properties.Double();
302      _MassFlow_Ammonia.Value = 0.0;
303      _Ammonia_Efficiency = new IPS.Properties.Double();
304      _Ammonia_Efficiency.Value = 0.0;
305      _MassFlow_H2O_Injection = new IPS.Properties.Double();
306      _MassFlow_H2O_Injection.Value = 0.0;
307      _MassFlow_Gas_Supply = new IPS.Properties.Double();
308      _MassFlow_Gas_Supply.Value = 0.0;
309
310      //Variables
311      _n_CO2_T = new IPS.Properties.Double();
312      _n_CO2_T.Value = 0.0;
313      _n_H2Og_T = new IPS.Properties.Double();
314      _n_H2Og_T.Value = 0.0;
315      _n_N2_T = new IPS.Properties.Double();
316      _n_N2_T.Value = 0.0;
317      _n_NO_T = new IPS.Properties.Double();
318      _n_NO_T.Value = 0.0;
319      _n_O2_T = new IPS.Properties.Double();
320      _n_O2_T.Value = 0.0;
321      _n_NH3_T = new IPS.Properties.Double();
322      _n_NH3_T.Value = 0.0;
323
324      _n_CO2_A = new IPS.Properties.Double();
325      _n_CO2_A.Value = 0.0;
326      _n_H2Og_A = new IPS.Properties.Double();
327      _n_H2Og_A.Value = 0.0;
328      _n_N2_A = new IPS.Properties.Double();
329      _n_N2_A.Value = 0.0;
330      _n_NO_A = new IPS.Properties.Double();
331      _n_NO_A.Value = 0.0;
332      _n_O2_A = new IPS.Properties.Double();
333      _n_O2_A.Value = 0.0;
334      _n_NH3_A = new IPS.Properties.Double();
335      _n_NH3_A.Value = 0.0;
336
337      _n_CO2_R = new IPS.Properties.Double();
338      _n_CO2_R.Value = 0.0;
339      _n_H2Og_R = new IPS.Properties.Double();
340      _n_H2Og_R.Value = 0.0;
341      _n_N2_R = new IPS.Properties.Double();
342      _n_N2_R.Value = 0.0;
343      _n_NO_R = new IPS.Properties.Double();
344      _n_NO_R.Value = 0.0;
345      _n_O2_R = new IPS.Properties.Double();
346      _n_O2_R.Value = 0.0;
347      _n_NH3_R = new IPS.Properties.Double();
348      _n_NH3_R.Value = 0.0;
349
350      _n_CO2_Total = new IPS.Properties.Double();
351      _n_CO2_Total.Value = 0.0;
352      _n_H2Og_Total = new IPS.Properties.Double();
353      _n_H2Og_Total.Value = 0.0;
354      _n_N2_Total = new IPS.Properties.Double();
```



```
355     _n_N2_Total.Value = 0.0;
356     _n_NO_Total = new IPS.Properties.Double();
357     _n_NO_Total.Value = 0.0;
358     _n_O2_Total = new IPS.Properties.Double();
359     _n_O2_Total.Value = 0.0;
360     _n_NH3_Total = new IPS.Properties.Double();
361     _n_NH3_Total.Value = 0.0;
362
363     _MassFlow_CO2 = new IPS.Properties.Double();
364     _MassFlow_CO2.Value = 0;
365     _MassFlow_H2Og = new IPS.Properties.Double();
366     _MassFlow_H2Og.Value = 0;
367     _MassFlow_N2 = new IPS.Properties.Double();
368     _MassFlow_N2.Value = 0;
369     _MassFlow_NO = new IPS.Properties.Double();
370     _MassFlow_NO.Value = 0;
371     _MassFlow_O2 = new IPS.Properties.Double();
372     _MassFlow_O2.Value = 0;
373     _MassFlow_NH3 = new IPS.Properties.Double();
374     _MassFlow_NH3.Value = 0;
375
376     _Ammonia_Fraction = new IPS.Properties.Double();
377     _Ammonia_Fraction.Value = 0;
378
379     //Outputs
380     _Mass_Fraction_CO2_Out = new IPS.Properties.Double();
381     _Mass_Fraction_CO2_Out.Value = 0.0;
382     _Mass_Fraction_H2Og_Out = new IPS.Properties.Double();
383     _Mass_Fraction_H2Og_Out.Value = 0.0;
384     _Mass_Fraction_N2_Out = new IPS.Properties.Double();
385     _Mass_Fraction_N2_Out.Value = 0.0;
386     _Mass_Fraction_NO_Out = new IPS.Properties.Double();
387     _Mass_Fraction_NO_Out.Value = 0.0;
388     _Mass_Fraction_O2_Out = new IPS.Properties.Double();
389     _Mass_Fraction_O2_Out.Value = 0.0;
390     _Mass_Fraction_NH3_Out = new IPS.Properties.Double();
391     _Mass_Fraction_NH3_Out.Value = 0.0;
392
393     _Total_MassFlow = new IPS.Properties.Double();
394     _Total_MassFlow.Value = 0.0;
395     _Total_Mass_Fraction = new IPS.Properties.Double();
396     _Total_Mass_Fraction.Value = 0.0;
397     _NOx_at_SCR_Inlet = new IPS.Properties.Double();
398     _NOx_at_SCR_Inlet.Value = 0.0;
399     _NOx_after_Ammonia = new IPS.Properties.Double();
400     _NOx_after_Ammonia.Value = 0.0;
401     _Ammonia_Slip = new IPS.Properties.Double();
402     _Ammonia_Slip.Value = 0.0;
403     _Omega = new IPS.Properties.Double();
404     _Omega.Value = 0.0;
405
406 }
407
408 //property declarations to make
409 //parameters visible to outside world
410
411 //Inputs
412 //-----
413 [PropertyUsage(UseProperty.DYNAMIC)]
```



```
414 [GridCategory(new String[]{"Inputs [%]"})]
415 [GridOrder(100)]
416 public IPS.Properties.Double Ammonia_Efficiency
417 {
418     get
419     {
420         return _Ammonia_Efficiency;
421     }
422 }
423 [PropertyUsage(UseProperty.DYNAMIC)]
424 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
425 [GridOrder(101)]
426 public IPS.Properties.Double Mass_Fraction_CO2_Turbine
427 {
428     get
429     {
430         return _Mass_Fraction_CO2_Turbine;
431     }
432 }
433 [PropertyUsage(UseProperty.DYNAMIC)]
434 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
435 [GridOrder(102)]
436 public IPS.Properties.Double Mass_Fraction_H2Og_Turbine
437 {
438     get
439     {
440         return _Mass_Fraction_H2Og_Turbine;
441     }
442 }
443 [PropertyUsage(UseProperty.DYNAMIC)]
444 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
445 [GridOrder(103)]
446 public IPS.Properties.Double Mass_Fraction_N2_Turbine
447 {
448     get
449     {
450         return _Mass_Fraction_N2_Turbine;
451     }
452 }
453 [PropertyUsage(UseProperty.DYNAMIC)]
454 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
455 [GridOrder(104)]
456 public IPS.Properties.Double Mass_Fraction_NO_Turbine
457 {
458     get
459     {
460         return _Mass_Fraction_NO_Turbine;
461     }
462 }
463 [PropertyUsage(UseProperty.DYNAMIC)]
464 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
465 [GridOrder(105)]
466 public IPS.Properties.Double Mass_Fraction_O2_Turbine
467 {
468     get
469     {
470         return _Mass_Fraction_O2_Turbine;
471     }
472 }
```



```
473 [PropertyUsage(UseProperty.DYNAMIC)]
474 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
475 [GridOrder(106)]
476 public IPS.Properties.Double Mass_Fraction_NH3_Turbine
477 {
478     get
479     {
480         return _Mass_Fraction_NH3_Turbine;
481     }
482 }
483 [PropertyUsage(UseProperty.DYNAMIC)]
484 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
485 [GridOrder(110)]
486 public IPS.Properties.Double Mass_Fraction_CO2_Ammonia
487 {
488     get
489     {
490         return _Mass_Fraction_CO2_Ammonia;
491     }
492 }
493
494 [PropertyUsage(UseProperty.DYNAMIC)]
495 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
496 [GridOrder(111)]
497 public IPS.Properties.Double Mass_Fraction_H2Og_Ammonia
498 {
499     get
500     {
501         return _Mass_Fraction_H2Og_Ammonia;
502     }
503 }
504
505 [PropertyUsage(UseProperty.DYNAMIC)]
506 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
507 [GridOrder(112)]
508 public IPS.Properties.Double Mass_Fraction_N2_Ammonia
509 {
510     get
511     {
512         return _Mass_Fraction_N2_Ammonia;
513     }
514 }
515
516 [PropertyUsage(UseProperty.DYNAMIC)]
517 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
518 [GridOrder(113)]
519 public IPS.Properties.Double Mass_Fraction_NO_Ammonia
520 {
521     get
522     {
523         return _Mass_Fraction_NO_Ammonia;
524     }
525 }
526
527 [PropertyUsage(UseProperty.DYNAMIC)]
528 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
529 [GridOrder(114)]
530 public IPS.Properties.Double Mass_Fraction_O2_Ammonia
531 {
```




```
532         get
533     {
534         return _Mass_Fraction_O2_Ammonia;
535     }
536 }
537
538 [PropertyUsage(UseProperty.DYNAMIC)]
539 [GridCategory(new String[]{"Inputs (Mass Fractions)"})]
540 [GridOrder(115)]
541 public IPS.Properties.Double Mass_Fraction_NH3_Ammonia
542 {
543     get
544     {
545         return _Mass_Fraction_NH3_Ammonia;
546     }
547 }
548 [PropertyUsage(UseProperty.DYNAMIC)]
549 [GridCategory(new String[]{"Inputs [kg/s]"})]
550 [GridOrder(120)]
551 public IPS.Properties.Double MassFlow_Turbine
552 {
553     get
554     {
555         return _MassFlow_Turbine;
556     }
557 }
558 [PropertyUsage(UseProperty.DYNAMIC)]
559 [GridCategory(new String[]{"Inputs [kg/s]"})]
560 [GridOrder(121)]
561 public IPS.Properties.Double MassFlow_Ammonia
562 {
563     get
564     {
565         return _MassFlow_Ammonia;
566     }
567 }
568 [PropertyUsage(UseProperty.DYNAMIC)]
569 [GridCategory(new String[]{"Inputs [kg/s]"})]
570 [GridOrder(122)]
571 public IPS.Properties.Double MassFlow_H2O_Injection
572 {
573     get
574     {
575         return _MassFlow_H2O_Injection;
576     }
577 }
578 [PropertyUsage(UseProperty.DYNAMIC)]
579 [GridCategory(new String[]{"Inputs [kg/s]"})]
580 [GridOrder(123)]
581 public IPS.Properties.Double MassFlow_Gas_Supply
582 {
583     get
584     {
585         return _MassFlow_Gas_Supply;
586     }
587 }
588 //Variables
589 //-----
590 [PropertyUsage(UseProperty.DYNAMIC)]
```



```
591 [GridCategory(new String[]{"Variables"})]
592 [GridOrder(200)]
593 public IPS.Properties.Double Ammonia_Fraction
594 {
595     get
596     {
597         return _Ammonia_Fraction;
598     }
599 }
600 [PropertyUsage(UseProperty.DYNAMIC)]
601 [GridCategory(new String[]{"Variables"})]
602 [GridOrder(201)]
603 public IPS.Properties.Double n_CO2_T
604 {
605     get
606     {
607         return _n_CO2_T;
608     }
609 }
610 [PropertyUsage(UseProperty.DYNAMIC)]
611 [GridCategory(new String[]{"Variables"})]
612 [GridOrder(202)]
613 public IPS.Properties.Double n_H2Og_T
614 {
615     get
616     {
617         return _n_H2Og_T;
618     }
619 }
620 [PropertyUsage(UseProperty.DYNAMIC)]
621 [GridCategory(new String[]{"Variables"})]
622 [GridOrder(203)]
623 public IPS.Properties.Double n_N2_T
624 {
625     get
626     {
627         return _n_N2_T;
628     }
629 }
630 [PropertyUsage(UseProperty.DYNAMIC)]
631 [GridCategory(new String[]{"Variables"})]
632 [GridOrder(204)]
633 public IPS.Properties.Double n_NO_T
634 {
635     get
636     {
637         return _n_NO_T;
638     }
639 }
640 [PropertyUsage(UseProperty.DYNAMIC)]
641 [GridCategory(new String[]{"Variables"})]
642 [GridOrder(205)]
643 public IPS.Properties.Double n_O2_T
644 {
645     get
646     {
647         return _n_O2_T;
648     }
649 }
```




```
650 [PropertyUsage(UseProperty.DYNAMIC)]
651 [GridCategory(new String[]{"Variables"})]
652 [GridOrder(206)]
653 public IPS.Properties.Double n_NH3_T
654 {
655     get
656     {
657         return _n_NH3_T;
658     }
659 }
660 [PropertyUsage(UseProperty.DYNAMIC)]
661 [GridCategory(new String[]{"Variables"})]
662 [GridOrder(211)]
663 public IPS.Properties.Double n_CO2_A
664 {
665     get
666     {
667         return _n_CO2_A;
668     }
669 }
670 [PropertyUsage(UseProperty.DYNAMIC)]
671 [GridCategory(new String[]{"Variables"})]
672 [GridOrder(212)]
673 public IPS.Properties.Double n_H2Og_A
674 {
675     get
676     {
677         return _n_H2Og_A;
678     }
679 }
680 [PropertyUsage(UseProperty.DYNAMIC)]
681 [GridCategory(new String[]{"Variables"})]
682 [GridOrder(213)]
683 public IPS.Properties.Double n_N2_A
684 {
685     get
686     {
687         return _n_N2_A;
688     }
689 }
690 [PropertyUsage(UseProperty.DYNAMIC)]
691 [GridCategory(new String[]{"Variables"})]
692 [GridOrder(214)]
693 public IPS.Properties.Double n_NO_A
694 {
695     get
696     {
697         return _n_NO_A;
698     }
699 }
700 [PropertyUsage(UseProperty.DYNAMIC)]
701 [GridCategory(new String[]{"Variables"})]
702 [GridOrder(215)]
703 public IPS.Properties.Double n_O2_A
704 {
705     get
706     {
707         return _n_O2_A;
708     }
709 }
```



```
709     }
710     [PropertyUsage(UseProperty.DYNAMIC)]
711     [GridCategory(new String[] {"Variables"})]
712     [GridOrder(216)]
713     public IPS.Properties.Double n_NH3_A
714     {
715         get
716         {
717             return _n_NH3_A;
718         }
719     }
720     [PropertyUsage(UseProperty.DYNAMIC)]
721     [GridCategory(new String[] {"Variables"})]
722     [GridOrder(221)]
723     public IPS.Properties.Double n_CO2_R
724     {
725         get
726         {
727             return _n_CO2_R;
728         }
729     }
730     [PropertyUsage(UseProperty.DYNAMIC)]
731     [GridCategory(new String[] {"Variables"})]
732     [GridOrder(222)]
733     public IPS.Properties.Double n_H2Og_R
734     {
735         get
736         {
737             return _n_H2Og_R;
738         }
739     }
740     [PropertyUsage(UseProperty.DYNAMIC)]
741     [GridCategory(new String[] {"Variables"})]
742     [GridOrder(223)]
743     public IPS.Properties.Double n_N2_R
744     {
745         get
746         {
747             return _n_N2_R;
748         }
749     }
750     [PropertyUsage(UseProperty.DYNAMIC)]
751     [GridCategory(new String[] {"Variables"})]
752     [GridOrder(224)]
753     public IPS.Properties.Double n_NO_R
754     {
755         get
756         {
757             return _n_NO_R;
758         }
759     }
760     [PropertyUsage(UseProperty.DYNAMIC)]
761     [GridCategory(new String[] {"Variables"})]
762     [GridOrder(225)]
763     public IPS.Properties.Double n_O2_R
764     {
765         get
766         {
767             return _n_O2_R;
```



```
768         }
769     }
770     [PropertyUsage (UseProperty.DYNAMIC)]
771     [GridCategory(new String[]{"Variables"})]
772     [GridOrder(226)]
773     public IPS.Properties.Double n_NH3_R
774     {
775         get
776         {
777             return _n_NH3_R;
778         }
779     }
780     [PropertyUsage (UseProperty.DYNAMIC)]
781     [GridCategory(new String[]{"Variables"})]
782     [GridOrder(231)]
783     public IPS.Properties.Double n_CO2_Total
784     {
785         get
786         {
787             return _n_CO2_Total;
788         }
789     }
790     [PropertyUsage (UseProperty.DYNAMIC)]
791     [GridCategory(new String[]{"Variables"})]
792     [GridOrder(232)]
793     public IPS.Properties.Double n_H2Og_Total
794     {
795         get
796         {
797             return _n_H2Og_Total;
798         }
799     }
800     [PropertyUsage (UseProperty.DYNAMIC)]
801     [GridCategory(new String[]{"Variables"})]
802     [GridOrder(233)]
803     public IPS.Properties.Double n_N2_Total
804     {
805         get
806         {
807             return _n_N2_Total;
808         }
809     }
810     [PropertyUsage (UseProperty.DYNAMIC)]
811     [GridCategory(new String[]{"Variables"})]
812     [GridOrder(234)]
813     public IPS.Properties.Double n_NO_Total
814     {
815         get
816         {
817             return _n_NO_Total;
818         }
819     }
820     [PropertyUsage (UseProperty.DYNAMIC)]
821     [GridCategory(new String[]{"Variables"})]
822     [GridOrder(235)]
823     public IPS.Properties.Double n_O2_Total
824     {
825         get
826         {
```



```
827         return _n_O2_Total;
828     }
829 }
830 [PropertyUsage(UseProperty.DYNAMIC)]
831 [GridCategory(new String[]{"Variables"})]
832 [GridOrder(236)]
833 public IPS.Properties.Double n_NH3_Total
834 {
835     get
836     {
837         return _n_NH3_Total;
838     }
839 }
840 [PropertyUsage(UseProperty.DYNAMIC)]
841 [GridCategory(new String[]{"Variables"})]
842 [GridOrder(241)]
843 public IPS.Properties.Double MassFlow_CO2
844 {
845     get
846     {
847         return _MassFlow_CO2;
848     }
849 }
850 [PropertyUsage(UseProperty.DYNAMIC)]
851 [GridCategory(new String[]{"Variables"})]
852 [GridOrder(242)]
853 public IPS.Properties.Double MassFlow_H2Og
854 {
855     get
856     {
857         return _MassFlow_H2Og;
858     }
859 }
860 [PropertyUsage(UseProperty.DYNAMIC)]
861 [GridCategory(new String[]{"Variables"})]
862 [GridOrder(243)]
863 public IPS.Properties.Double MassFlow_N2
864 {
865     get
866     {
867         return _MassFlow_N2;
868     }
869 }
870 [PropertyUsage(UseProperty.DYNAMIC)]
871 [GridCategory(new String[]{"Variables"})]
872 [GridOrder(244)]
873 public IPS.Properties.Double MassFlow_NO
874 {
875     get
876     {
877         return _MassFlow_NO;
878     }
879 }
880 [PropertyUsage(UseProperty.DYNAMIC)]
881 [GridCategory(new String[]{"Variables"})]
882 [GridOrder(245)]
883 public IPS.Properties.Double MassFlow_O2
884 {
885     get
```



```
886 {  
887     return _MassFlow_O2;  
888 }  
889 }  
890 [PropertyUsage(UseProperty.DYNAMIC)]  
891 [GridCategory(new String[]{"Variables"})]  
892 [GridOrder(246)]  
893 public IPS.Properties.Double MassFlow_NH3  
894 {  
895     get  
896     {  
897         return _MassFlow_NH3;  
898     }  
899 }  
900 //Outputs  
901 //-----  
902 [PropertyUsage(UseProperty.DYNAMIC)]  
903 [GridCategory(new String[]{"Outputs (Mass Fractions)"})]  
904 [GridOrder(310)]  
905 public IPS.Properties.Double Mass_Fraction_CO2_Out  
906 {  
907     get  
908     {  
909         return _Mass_Fraction_CO2_Out;  
910     }  
911 }  
912 [PropertyUsage(UseProperty.DYNAMIC)]  
913 [GridCategory(new String[]{"Outputs (Mass Fractions)"})]  
914 [GridOrder(311)]  
915 public IPS.Properties.Double Mass_Fraction_H2Og_Out  
916 {  
917     get  
918     {  
919         return _Mass_Fraction_H2Og_Out;  
920     }  
921 }  
922 [PropertyUsage(UseProperty.DYNAMIC)]  
923 [GridCategory(new String[]{"Outputs (Mass Fractions)"})]  
924 [GridOrder(312)]  
925 public IPS.Properties.Double Mass_Fraction_N2_Out  
926 {  
927     get  
928     {  
929         return _Mass_Fraction_N2_Out;  
930     }  
931 }  
932 [PropertyUsage(UseProperty.DYNAMIC)]  
933 [GridCategory(new String[]{"Outputs (Mass Fractions)"})]  
934 [GridOrder(313)]  
935 public IPS.Properties.Double Mass_Fraction_NO_Out  
936 {  
937     get  
938     {  
939         return _Mass_Fraction_NO_Out;  
940     }  
941 }  
942 [PropertyUsage(UseProperty.DYNAMIC)]  
943 [GridCategory(new String[]{"Outputs (Mass Fractions)"})]  
944 [GridOrder(314)]
```



SIM-00001-Nox

```
945     public IPS.Properties.Double Mass_Fraction_O2_Out
946     {
947         get
948         {
949             return _Mass_Fraction_O2_Out;
950         }
951     }
952     [PropertyUsage (UseProperty.DYNAMIC)]
953     [GridCategory(new String[]{"Outputs (Mass Fractions)"})]
954     [GridOrder(315)]
955     public IPS.Properties.Double Mass_Fraction_NH3_Out
956     {
957         get
958         {
959             return _Mass_Fraction_NH3_Out;
960         }
961     }
962     [PropertyUsage (UseProperty.DYNAMIC)]
963     [GridCategory(new String[]{"Outputs (Mass Fractions)"})]
964     [GridOrder(320)]
965     public IPS.Properties.Double Total_Mass_Fraction
966     {
967         get
968         {
969             return _Total_Mass_Fraction;
970         }
971     }
972     [PropertyUsage (UseProperty.DYNAMIC)]
973     [GridCategory(new String[]{"Outputs [kg/s]"})]
974     [GridOrder(330)]
975     public IPS.Properties.Double Total_MassFlow
976     {
977         get
978         {
979             return _Total_MassFlow;
980         }
981     }
982     [PropertyUsage (UseProperty.DYNAMIC)]
983     [GridCategory(new String[]{"Outputs [ppm]"})]
984     [GridOrder(340)]
985     public IPS.Properties.Double NOx_at_SCR_Inlet
986     {
987         get
988         {
989             return _NOx_at_SCR_Inlet;
990         }
991     }
992     [PropertyUsage (UseProperty.DYNAMIC)]
993     [GridCategory(new String[]{"Outputs [ppm]"})]
994     [GridOrder(350)]
995     public IPS.Properties.Double NOx_after_Ammonia
996     {
997         get
998         {
999             return _NOx_after_Ammonia;
1000     }
1001     }
1002     [PropertyUsage (UseProperty.DYNAMIC)]
1003     [GridCategory(new String[]{"Outputs [ppm]"})]
```



SIM-00001-Nox

```
1004 [GridOrder(360)]
1005 public IPS.Properties.Double Ammonia_Slip
1006 {
1007     get
1008     {
1009         return _Ammonia_Slip;
1010     }
1011 }
1012 [PropertyUsage(UseProperty.DYNAMIC)]
1013 [GridCategory(new String[]{"Outputs (Ratio)"})]
1014 [GridOrder(370)]
1015 public IPS.Properties.Double Omega
1016 {
1017     get
1018     {
1019         return _Omega;
1020     }
1021 }
1022 }
```