ATR Model Functions README

This document describes the interpolation functions contained in `ATR_model_functions.pkl`. These functions provide instantaneous values of the ATR plant's most important parameters, based on oxygen flowrate (F_O2) as the independent variable. The transient behavior of the plant is not accounted for, so the validity of the present functions is only guaranteed when using an oxygen profile with smooth, gradual changes.

Quick Start

```python
import pickle
import numpy as np
import scipy.interpolate

# Load the ATR model functions
with open('ATR_model_functions.pkl', 'rb') as f:
    plant_model = pickle.load(f)

# Extract individual functions
H01_Q_func = plant_model['H01_Q_func']
F_bio_func = plant_model['F_bio_func']

# Use the functions
O2_val = 15.0  # kmol/hr (within valid range: 7.125 - 23.75 kmol/hr)
heat_input = H01_Q_func(O2_val)
bio_flowrate = F_bio_func(O2_val)
```

Independent Variable

All functions use oxygen molar flowrate (F_O2) as the independent variable

- **Units:** kmol/hr
- **Valid Range:** 7.125 - 23.75 kmol/hr

Available Functions

Input Stream Functions

Function Name	Variable	Description	Units
`F_bio_func`	F_bio	Biogas molar flowrate	kmol/hr
`Fm_bio_func`	Fm_bio	Biogas mass flowrate	kg/hr
`F_steam_func`	F_steam	Steam molar flowrate	kmol/hr

Function Name	Variable	Description	Units
Fm_steam_func	Fm_steam	Steam mass flowrate	kg/hr
Fm_O2_func	Fm_O2	Oxygen mass flowrate	kg/hr

Output Stream Functions

Function Name	Variable	Description	Units
F_H2_func	F_H2	Hydrogen molar flowrate	kmol/hr
Fm_H2_func	Fm_H2	Hydrogen mass flowrate	kg/hr
F_water_func	F_water	Water molar flowrate	kmol/hr
Fm_water_func	Fm_water	Water mass flowrate	kg/hr
F_offgas_func	F_offgas	Off-gas molar flowrate	kmol/hr
Fm_offgas_func	Fm_offgas	Off-gas mass flowrate	kg/hr

Off-gas Composition Functions

Function Name	Variable	Description	Units
xCO2_offgas_func	xCO2_offgas	CO_2 mole fraction in offgas	dimensionless (0-1)
xH2_offgas_func	xH2_offgas	H_2 mole fraction in offgas	dimensionless (0-1)
xCH4_offgas_func	xCH4_offgas	CH_4 mole fraction in offgas	dimensionless (0-1)

Heater Functions

Heater Duties

Function Name	Variable	Description	Units
H01_Q_func	H01_Q	H01 heater duty	kW
H02_Q_func	H02_Q	H02 heater duty	kW
H04_Q_func	H04_Q	H04 heater duty	kW

Heater Temperatures

Function Name	Variable	Description	Units
Tout_H01_func	Tout_H01	H01 heater outlet temperature	°C
Tout_H02_func	Tout_H02	H02 heater outlet temperature	°C
Tout_H04_func	Tout_H04	H04 heater outlet temperature	°C
Tin_H01_func	Tin_H01	H01 heater inlet temperature	°C
Tin_H02_func	Tin_H02	H02 heater inlet temperature	°C
Tin_H04_func	Tin_H04	H04 heater inlet temperature	°C

Cooler Functions

Cooler Duties

Function Name	Variable	Description	Units
H05_Q_func	H05_Q	H05 cooler duty	kW
H08_Q_func	H08_Q	H08 cooler duty	kW
H09_Q_func	H09_Q	H09 cooler duty	kW

Cooler Temperatures

Function Name	Variable	Description	Units
Tout_H05_func	Tout_H05	H05 cooler outlet temperature	°C
Tout_H08_func	Tout_H08	H08 cooler outlet temperature	°C
Tout_H09_func	Tout_H09	H09 cooler outlet temperature	°C
Tin_H05_func	Tin_H05	H05 cooler inlet temperature	°C
Tin_H08_func	Tin_H08	H08 cooler inlet temperature	°C
Tin_H09_func	Tin_H09	H09 cooler inlet temperature	°C

Energy Functions

Function Name	Variable	Description	Units
W_in_func	W_in	Total work input	kW

Unit Conventions

- **F:** Molar flowrates in kmol/hr
- **Fm:** Mass flowrates in kg/hr
- **Q:** Heat duties in kW (positive for heating, negative for cooling)
- **W:** Work/power in kW
- **x:** Mole fractions (dimensionless, range 0-1)
- **T:** Temperatures in °C

Required Libraries

```python
import pickle       # For loading the functions
import numpy as np  # For array operations (optional)
import scipy.interpolate  # Required by the interpolation functions
```

Usage Notes

1. **Transient Behavior:** These functions do not account for transient plant behavior. They are valid only for smooth, gradual changes in oxygen flowrate.
2. **Input Validation:** Ensure F_O2 values are within the valid range (7.125 - 23.75 kmol/hr) to avoid extrapolation errors.

Example Usage

python

```python
import pickle
import numpy as np

# Load model
with open('ATR_model_functions.pkl', 'rb') as f:
    model = pickle.load(f)

# Define oxygen flowrate range
F_O2_values = np.linspace(7.125, 23.75, 100)

# Calculate corresponding outputs
H2_production = [model['F_H2_func'](F_O2) for F_O2 in F_O2_values]
total_heating = [model['H01_Q_func'](F_O2) +
                 model['H02_Q_func'](F_O2) +
                 model['H04_Q_func'](F_O2) for F_O2 in F_O2_values]
```

Process Flowsheet