TOWARDS A SOFTWARE PROTOTYPE FOR SYNTHESIS OF OPERABLE PROCESS INTENSIFICATION SYSTEMS

A Thesis

by

SHIVAM VEDANT

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Efstratios N. Pistikopoulos
Committee Members,	Mahmoud M. El-Halwagi
	Hongcai Zhou
Head of Department,	Arul Jayaraman

August 2021

Major Subject: Chemical Engineering

ABSTRACT

Process Intensification (PI) has been identified as a promising tool to drastically improve process profitability and energy efficiency, and reduce process waste through the application of innovative schemes and processes. The past few decades have witnessed burgeoning interest in the field of PI towards systematic innovation of chemical processes. Recent efforts by the Process Systems Engineering (PSE) community have focused on: (i) generating novel intensified designs without pre-postulating chemical processes and/or equipment, (ii) flexibility, safety, and controllability analysis of PI systems under uncertain conditions, and (iii) integrated framework for process synthesis, optimization, and operability analysis of PI processes. Despite current advances in commercial software for chemical process modeling and simulation, there lacks a widely used software platform to support PI innovation, synthesis, and operability analysis.

In this work, we present a software prototype for systematic generation of PI process systems with guaranteed safety, flexibility, and control performances. The prototype leverages state-of-the-art PSE strategies and toolkits in an integrated manner for: (i) computer-aided process synthesis and optimization using the Generalized Modular Representation Framework, (ii) operability and safety analysis using flexibility test and quantitative risk analysis, and (iii) explicit model predictive control following the PAROC (PARametric Optimization and Control) Framework. Commercial platforms embedded within the prototype include Python for user interface, GAMS® for process synthesis and optimization, DWSIM® & ASPEN® for process simulation, and gPROMS® & MATLAB® for dynamic analysis, process operability, and process control. The capabilities of the prototype are demonstrated through a case study on the process design and intensification of pentene metathesis reaction/separation systems.

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Efstratios N. Pistikopoulos for his continuous guidance and encouragement. I would like to extend my gratitude to my colleagues at the Multi-parametric Optimization and Control Group for their support and collaboration. I would also like to convey special thanks to faculty and staff at Texas A&M Energy Institute and Artie McFerrin Department of Chemical Engineering.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professors Efstratios N. Pistikopoulos, Mahmoud M. El-Halwagi, and Hongcai Zhou. The frameworks used in the study were adopted from the previous works of the Multi-parametric Optimization and Control group, chiefly from the works of Yuhe Tian, Iosif Pappas, Baris Burnak, and Justin Katz. This work is in collaboration with Yuhe Tian, Mary Rivera Atencio, Moustafa Ali, Dustin Kenefake, and Iosif Pappas. I would like to especially mention the contributions of Mary Rivera and Moustafa Ali towards the development of synthesis suite, model library, and operability suite.

All the other work conducted for the thesis was completed independently by me.

Funding Sources

The funding for project was provided by the Texas A&M Energy Institute, Shell, and RAPID SYNOPSIS Project (DE-EE0007888-09-03, Partner Organizations: Texas A&M University, Georgia Institute of Technology, Auburn University, Shell, Dow Chemical Company, Process Systems Enterprise Limited).

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

1. INTRODUCTION

Process Intensification (PI) has been considered as an effective strategy to develop innovative process alternatives for cheaper, safer, and sustainable design alternatives. PI offers the potential to substantially improve chemical processes by harnessing the synergies existing within a system [1, 2]. The past few decades have seen tremendous growth in the research and application of PI technologies in industries and academia alike [3, 4]. Given the demands of the generation and growing climate change concerns, the need for innovative and contemporary solutions to our problems are in high demand. To meet these challenges, there is a significant need to develop systematic methods and frameworks to generate and analyse novel PI flowsheets .

Commercial software toolkits such as ASCEND®, ASPEN HYSYS®, and Simulink® have provided platforms to address challenges in process design, simulation, and control. Furthermore, driven by several national initiatives such as RAPID [5], the academic community has initiated attempts to develop software prototypes for computer-aided process design, synthesis, and intensification leveraging state-of-the-art process systems engineering approaches [6, 7, 8, 9]. Platforms such as ProCAFD® and ProCACD®, Pyosyn®, and ICAS® provide capable environments to generate the chemical processes. However, the majority of the softwares necessitate the creation of superstructures based on available unit operations for both conventional and intensified process systems. This step may hinder the discovery of innovative and novel designs. Moreover, the available packages largely fail to consider the safety and operability aspects of process design in an integrated manner. Hence, there is a need for software platforms that can approach the design problem from a holistic perspective while allowing PI innovation.

In the recent years, the PSE community has focused on developing methodologies for generation and analysis of PI process systems. In this regard, Generalized Modular Representation Framework (GMF) offers potential to create chemical processes without prior assumptions of unit operations or equipment [10]. It employs phenomena-based building blocks, discussed in section 2.2, to create non-exclusive superstructures for a given synthesis problem. Moreover, it allows the integration of operability analyses during the synthesis stage to ensure flexibility and inherent safety performances of the resulting chemical processes. For the dynamic and control analysis of process systems, the PAROC framework [11] provides powerful model-based approaches. It relies on the execution of high-fidelity dynamic models and explicit multi-parametric model predictive control (mp-MPC) to deliver strategies ensuring nominal operation of processes under dynamic uncertainties.

To combine the various steps in process design, an integrated framework has been adopted from the previous works of the group [12]. The framework is capable of systematically generating intensified process systems with embedded process operability, safety, and explicit model predictive control analysis criterion. Different functionalities of the framework are integrated together by leveraging state-of-the-art software applications for process synthesis, optimization, model & simulation, and control analyses.

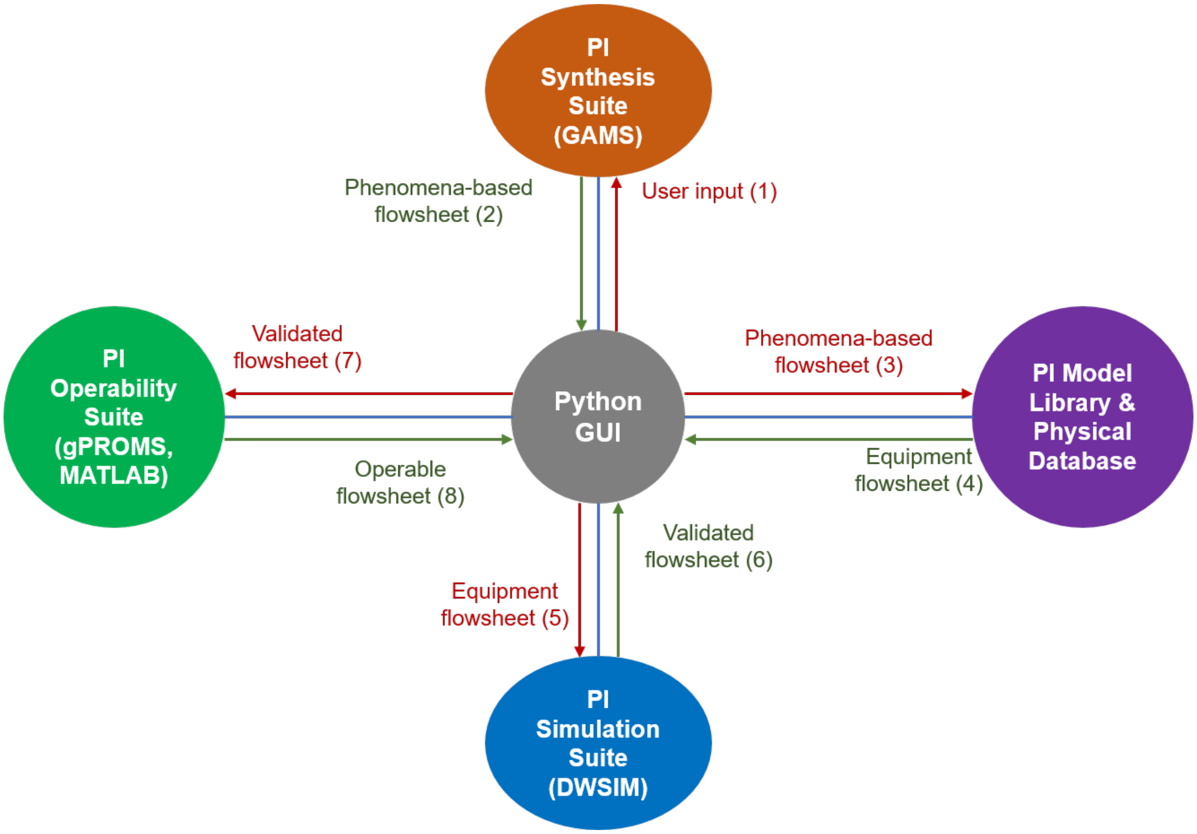

PI Synthesis Suite (GAMS)
Phenomena-based flowsheet (2)
User input (1)
PI Operability Suite (gPROMS, MATLAB)
Validated flowsheet (7)
Python GUI
Phenomena-based flowsheet (3)
PI Model Library & Physical Database
Operable flowsheet (8)
Equipment flowsheet (4)
Validated flowsheet (6)
Equipment flowsheet (5)
PI Simulation Suite (DWSIM)

Figure 1.1: Data flow within the prototype

Figure 1.1 highlights the different suites embedded within the prototype. The user interface (UI) consolidates the suites in a seamless manner to provide an integrated environment inclusive of user input. The input obtained from the UI is conveyed in a sequential manner to different suites along with the output from each design stage to develop the flowsheet. The suites leverage commercially available Process Systems Engineering (PSE) software packages such as, GAMS® for process synthesis and optimization, DWSIM® for steady-state process simulation, ASPEN® and gPROMS® for dynamic modeling, and MATLAB® for control analysis.

Key attributes of the prototype include: (i) providing an integrated platform for synthesis and analysis of PI systems, (ii) facilitating autonomous flow of information between various toolkits required during the designing process, and (iii) allowing a free exchange of information between user and the prototype to deliver critical insights.

The remainder of this thesis is structured as follows. In Chapter 2, components and capabilities of the prototype are elucidated. Next, Chapter 3 demonstrates the working potential of the framework through process design of the pentene metathesis reaction/separation system. Finally, concluding remarks and future directions are discussed in Chapter 4.

1.1 Thesis Objectives

To address the aforementioned challenges, the following objectives were set for the software prototype:

- Leveraging the currently available state-of-the-art PSE software packages;
- Enabling seamless and autonomous transfer of information between the various software platforms;
- Developing an open-source interface for convenient exchange of data with the user.

2. SOFTWARE PROTOTYPE

In this chapter, a detailed description of the software prototype and the individual suites is provided. As shown in Figure 1.1, the prototype consists of five suites: (i) User Interface, (ii) PI Synthesis Suite, (iii) PI Model Library & Physical database, (iv) PI Simulation Suite, and (v) PI Operability Suite.

2.1 User Interface

The user interface (UI) enables convenient exchange of information between user and the prototype, and an automated data transfer within the suites. Figure 1.1 illustrates the data flow chart for the prototype. The UI is built on python to enable the connectivity and transfer of information to various platforms. Through the interface, users can select individual suites for targeted equipment/flowsheet intensification, design or analysis. It can also guide users to navigate between different suites in a step-by-step manner for integrated design with operability, safety, and/or control.

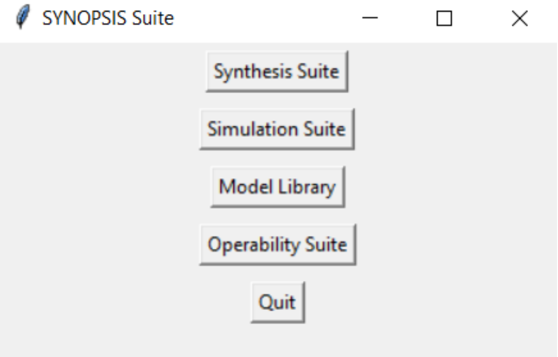


Figure 2.1: Software Prototype initial window

The initial window of the interface, displayed in Figure 2.1, enables access to multiple operational suites for targeted development of chemical flowsheets. The interfaces involved within the

The work has been accepted under the title "Towards a Software Prototype for Synthesis of Operable Process Intensification Systems" in Elsevier as part of the proceedings of the "31st European Symposium on Computer-Aided Process Engineering (ESCAPE-31)" - Appendix A framework, has been created using *tkinter* library available with the python modules. The synthesis window, shown in Figure 2.2 is structurally organized by creating different panels as *objects* of python defined *classes*. The key advantage of defining the panels as such is that the attributes and the appearances of these panels can be accessed and manipulated in real-time. The input panel enables the user to add chemical components, define feed composition, reaction data, feed conditions, and utility cost information. The result panel displays flowsheet results and configurations at different stages. Dedicated panels are established to display superstructure results, comparison between multiple iterations, and execution log data from the synthesis suite. To access the numerous functionalities and execute the desired operations within the prototype, *Button* widgets have been added to the interface. These widgets also enable initiating the designing process, saving the obtained solution, loading an existing file, aborting the calculations, and/or quitting the application on user command.

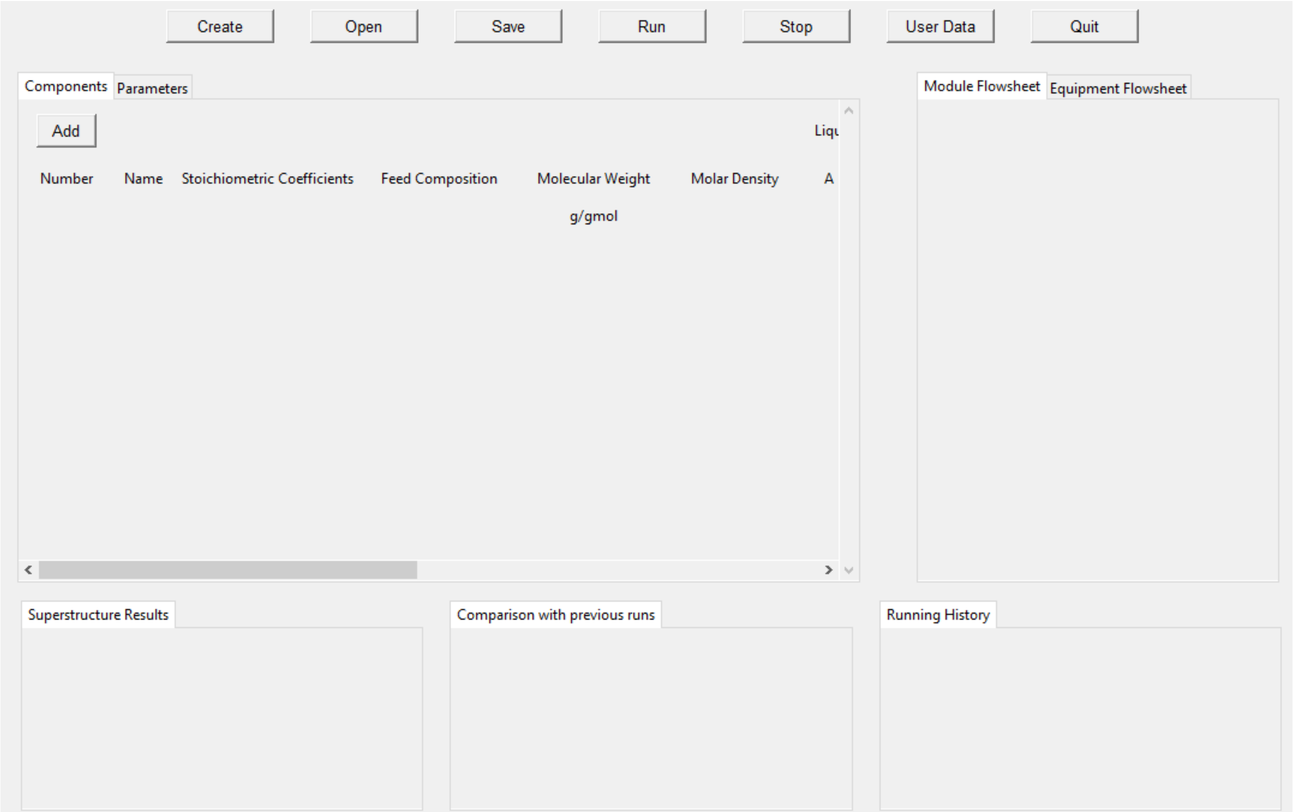
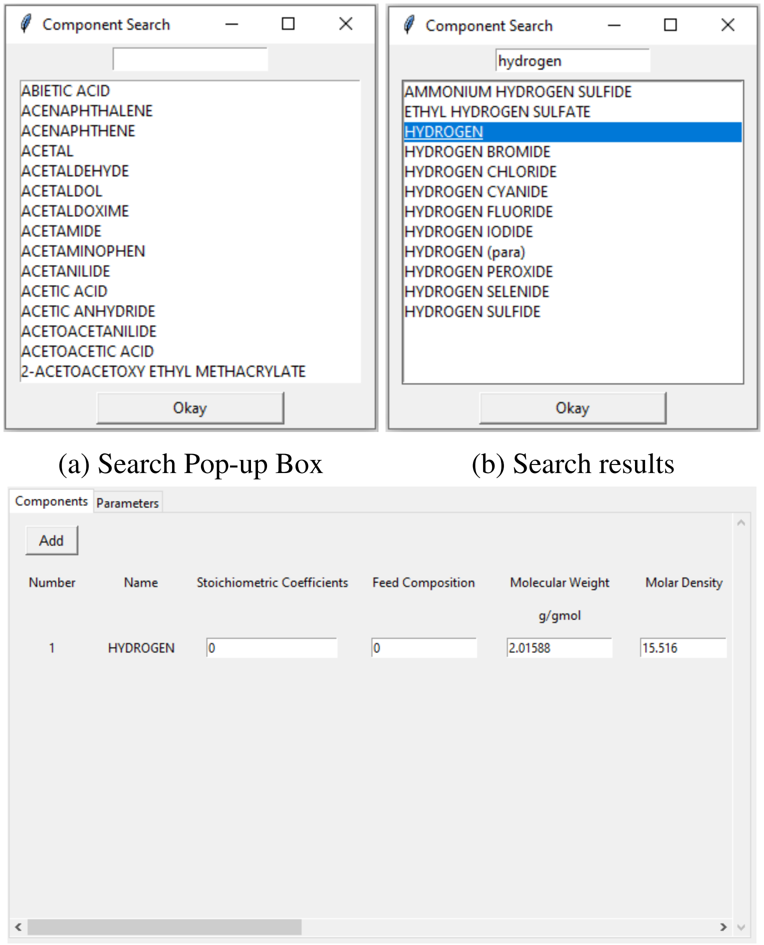


Figure 2.2: User interface for process synthesis

2.1.1 Adding components to the simulation

In order to select chemical compounds for simulation, a component search box has been provided. The box is equipped with the search field and is programmed to cross-reference the entered text against the names of components available with the thermophysical database. Figure 2.3a displays the pop-up search box that appears on pressing the *ADD* button provided in the input panel. Figure 2.3b further illustrates the revised list of compounds containing the word *"hydrogen"* within their name. To confirm the selection, *Okay* button should be pressed. It is imperative to note that only one compound can be added at a time.



ABIETIC ACID
ACENAPHTHALENE
ACENAPHTHENE
ACETAL
ACETALDEHYDE
ACETALDOL
ACETALDOXIME
ACETAMIDE
ACETAMINOPHEN
ACETANILIDE
ACETIC ACID
ACETIC ANHYDRIDE
ACETOACETANILIDE
ACETOACETIC ACID
2-ACETOACETOXY ETHYL METHACRYLATE

(a) Search Pop-up Box

hydrogen

AMMONIUM HYDROGEN SULFIDE
ETHYL HYDROGEN SULFATE
HYDROGEN
HYDROGEN BROMIDE
HYDROGEN CHLORIDE
HYDROGEN CYANIDE
HYDROGEN FLUORIDE
HYDROGEN IODIDE
HYDROGEN (para)
HYDROGEN PEROXIDE
HYDROGEN SELENIDE
HYDROGEN SULFIDE

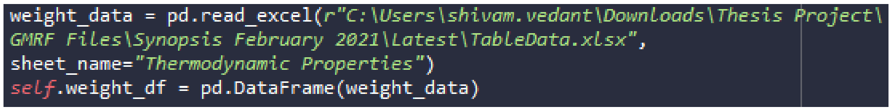
(b) Search results

Number	Name	Stoichiometric Coefficients	Feed Composition	Molecular Weight g/gmol	Molar Density
1	HYDROGEN	0	0	2.01588	15.516

(c) Interface after adding a component

Figure 2.3: Addition of chemical compounds to the simulation

Following the selection of a compound, the interface retrieves information like molecular weight, molar density, liquid & vapor heat capacity coefficients, and Gibbs energy of formation from dedicated python dictionaries containing the imported thermophysical data (in excel format). To import property values from the excel file, *pandas* library is utilized along with its *read_excel* function (Figure 2.4). The retrieved data is displayed on the interface with an option allowing modification of the values to accommodate unit consistencies, as illustrated by Fig. 2.3c.

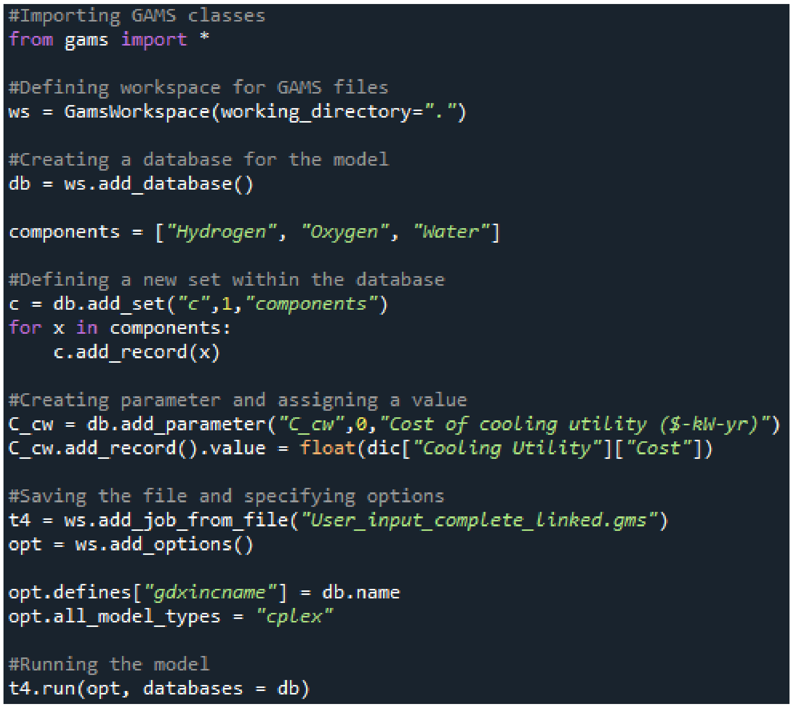


```
weight_data = pd.read_excel(r"C:\Users\shivam.vedant\Downloads\Thesis Project\
GMRF Files\Synopsis February 2021\Latest\TableData.xlsx",
sheet_name="Thermodynamic Properties")
self.weight_df = pd.DataFrame(weight_data)
```

Figure 2.4: Importing thermophysical database from excel via python

2.1.2 User data transfer to synthesis suite

Python Application Programming Interfaces (APIs) are employed to transfer user input from interface to the synthesis suite. APIs can be used for creation and execution of GAMS models, exchange of model input and results, and reconfigure the GAMS solver options. Figure 2.5 provides a brief example of a python script to build and run a GAMS model.



```
#Importing GAMS classes
from gams import *

#Defining workspace for GAMS files
ws = GamsWorkspace(working_directory=".")

#Creating a database for the model
db = ws.add_database()

components = ["Hydrogen", "Oxygen", "Water"]

#Defining a new set within the database
c = db.add_set("c",1,"components")
for x in components:
    c.add_record(x)

#Creating parameter and assigning a value
C_cw = db.add_parameter("C_cw",0,"Cost of cooling utility ($-kW-yr)")
C_cw.add_record().value = float(dic["Cooling Utility"]["Cost"])

#Saving the file and specifying options
t4 = ws.add_job_from_file("User_input_complete_linked.gms")
opt = ws.add_options()

opt.defines["gdxincname"] = db.name
opt.all_model_types = "cplex"

#Running the model
t4.run(opt, databases = db)
```

Figure 2.5: Python script used for creating and executing GAMS model

Initially, a GAMS workspace initialization is required to denote the location on the computer system for accessing and storing GAMS files. Following this, analogous to a GAMS model, a database is created within the python environment to accommodate definitions of required sets, variables, and parameters based on the user input can be added to the database. The python script can now be linked to a GAMS file; however, it is important to note that the name of defined sets and parameters are to be kept the same in both the files. For instance component set "c" and cooling water utility cost parameter "C_cw" defined in python script (Figure 2.5) have the same name as the corresponding set and parameter in the GAMS file (Figure 2.7). Upon completing the input requirements, users can press the provided *Run* button (Figure 2.2). Consequently, the user data would be automatically transferred to the synthesis suite.

2.2 Synthesis Suite

The synthesis suite, built on GAMS®, provide the tools required for the systematic generation of optimal PI flowsheets. GMF methodology is employed to develop novel and innovative flowsheets without pre-postulation of unit operations or equipment. To characterize various chemical processes,the framework employs two phenomenological modules, namely, the pure heat exchange module and the multi-functional mass/heat exchange module [10]. As a result of the optimization problem, interlinked configuration of these modules are obtained which can form the basis for creation of conventional and/or novel unit operations and processes.

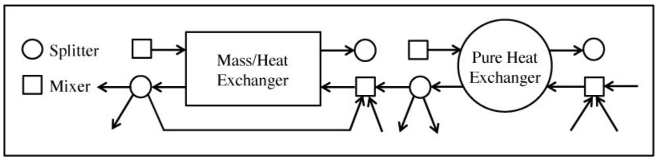
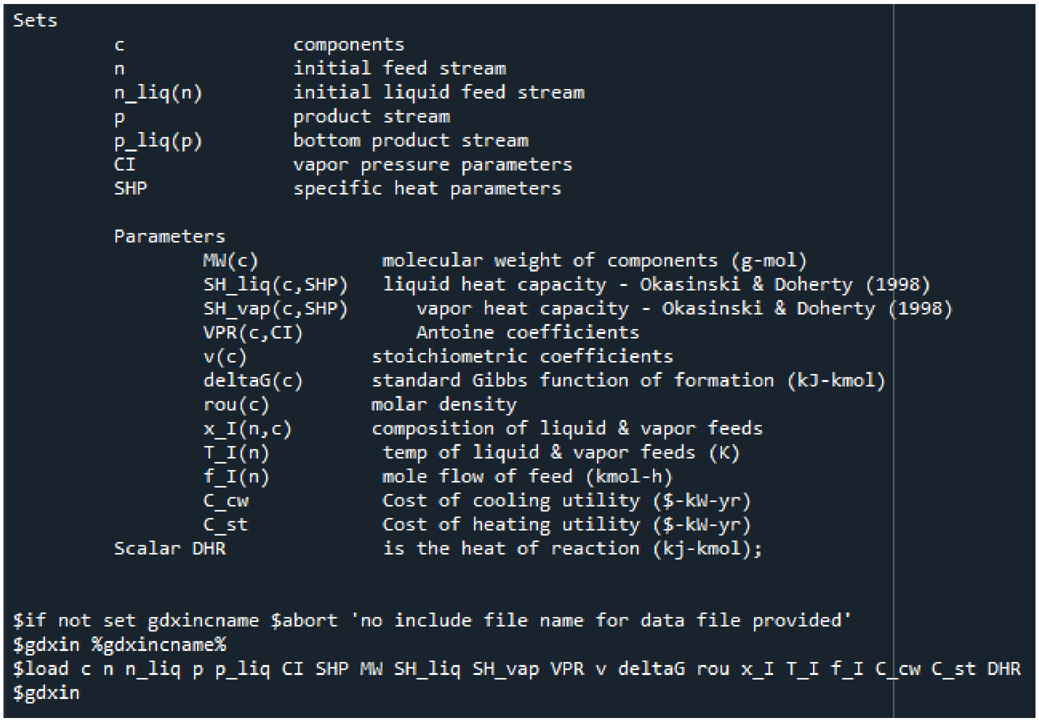


Figure 2.6: GMF building blocks

Upon obtaining the user input, the optimization problem is formulated as a Mixed Integer Non-Linear Programming (MINLP), and solved using the Generalized Benders Decomposition (GBD) algorithm.

2.2.1 User input from interface

To accurately capture the user data from the interface, a GAMS input file is created containing the definitions of the required sets, variables and parameters. As illustrated in Figure 2.7, the defined model structure does not include the set members and parameter values. This type of definition allows the python script to overwrite the entries, hence allowing the update in real-time. To import the data from the python environment, *load* command is executed.



```
Sets
         c               components
         n               initial feed stream
         n_liq(n)        initial liquid feed stream
         p               product stream
         p_liq(p)        bottom product stream
         CI              vapor pressure parameters
         SHP             specific heat parameters

         Parameters
                 MW(c)           molecular weight of components (g-mol)
                 SH_liq(c,SHP)   liquid heat capacity - Okasinski & Doherty (1998)
                 SH_vap(c,SHP)      vapor heat capacity - Okasinski & Doherty (1998)
                 VPR(c,CI)          Antoine coefficients
                 v(c)           stoichiometric coefficients
                 deltaG(c)      standard Gibbs function of formation (kJ-kmol)
                 rou(c)         molar density
                 x_I(n,c)       composition of liquid & vapor feeds
                 T_I(n)          temp of liquid & vapor feeds (K)
                 f_I(n)          mole flow of feed (kmol-h)
                 C_cw            Cost of cooling utility ($-kW-yr)
                 C_st            Cost of heating utility ($-kW-yr)
         Scalar DHR              is the heat of reaction (kj-kmol);


$if not set gdxincname $abort 'no include file name for data file provided'
$gdxin %gdxincname%
$load c n n_liq p p_liq CI SHP MW SH_liq SH_vap VPR v deltaG rou x_I T_I f_I C_cw C_st DHR
$gdxin
```

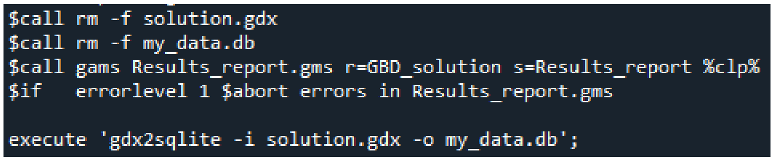
Figure 2.7: GAMS file structure allowing for dynamic updation through Python script

2.2.2 Results from Synthesis Suite

The collated user input is transferred to the GMF framework via an interconnected network of GAMS files. The optimal solution for the problem, generated in a GAMS Data eXchange (gdx) file, is exported to a database file (.db extension). To prevent overlapping of exported solution, results from the previous calculation is deleted in prior to updating the database file with the latest solution by calling *remove* function in GAMS file (Figure 2.8).

The solution set contains optimal values for:

- Objective function
- Continuous variables such as flow rates, temperature, and composition of streams
- Binary variables denoting the selection of GMF modules and/or connections between the selected modules



```
$call rm -f solution.gdx
$call rm -f my_data.db
$call gams Results_report.gms r=GBD_solution s=Results_report %clp%
$if   errorlevel 1 $abort errors in Results_report.gms

execute 'gdx2sqlite -i solution.gdx -o my_data.db';
```

Figure 2.8: Exporting GAMS model results to a database file

The solution set from the database file is interpreted through a python script to create a module-based flowsheet. Figure 2.9 illustrates the optimal configurations obtained as a result of the optimization problem [12] where the values from the database file are read via a series of conditional if/else statements to realize the value of binary variables. The generated flowsheet and superstructure results is displayed in their respective panels on the interface (Figure 3.3).

The solution of the optimization problem can provide key insights into PI process design. For instance the shaded modules indicate presence of a reaction along with the mass transfer between liquid-vapor phase, and feed is suggested to be introduced in two different stages of the column.

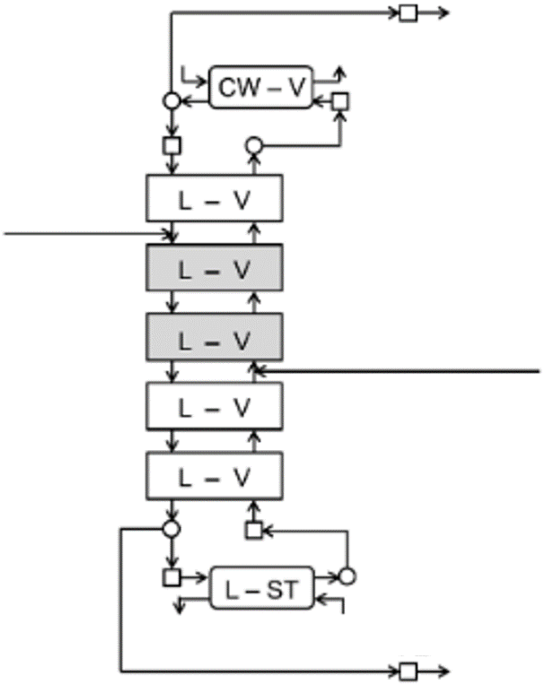
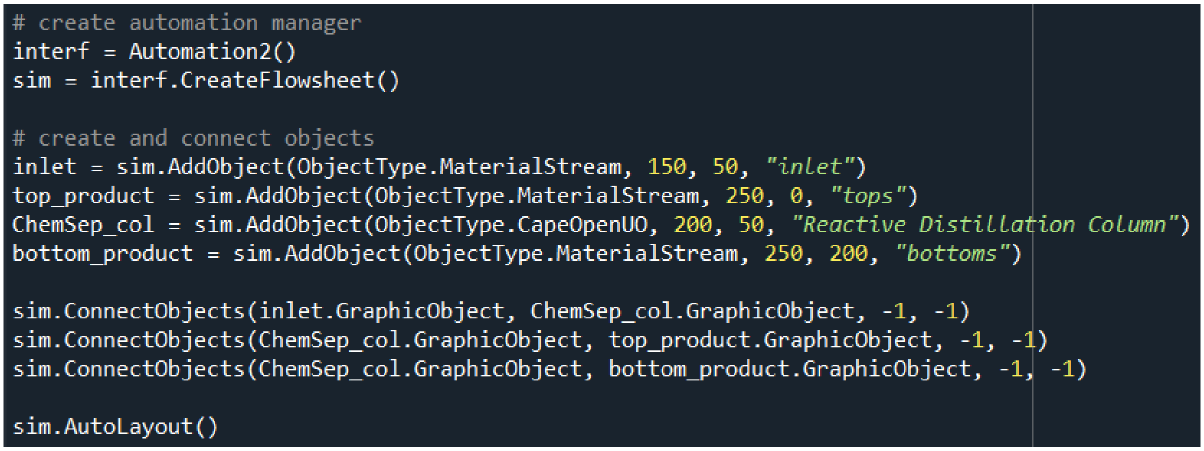


Figure 2.9: Module based representation of the optimal solution

2.3 Simulation Suite

The simulation suite is equipped with tools for validation of equipment based flowsheets under steady-state and/or dynamic mode of operation. Commercial software packages such as DWSIM® (open-source) and ASPEN® provide powerful platforms for the analysis. Following the creation of module based flowsheet (Figure 2.9), users can guide the translation to an equipment-based flowsheet by leveraging the high-fidelity models available with the software platforms. Upon finalizing the selections, the desired flowsheet can be created in the DWSIM® simulation environment using python scripts for validation.

Chemical compounds and simulation objects available with the DWSIM® packages can be added to the simulation using *sim.SelectedCompounds.Add()* and *sim.AddObject()* functions respectively. Subsequently, *sim.ConnectObjects()* function is called to establish linkages amongst the various selected unit operations. Figure 2.10 exemplifies the structure and workings of these functions to create a simulation for a reactive distillation column in Figure 2.11.



```
# create automation manager
interf = Automation2()
sim = interf.CreateFlowsheet()

# create and connect objects
inlet = sim.AddObject(ObjectType.MaterialStream, 150, 50, "inlet")
top_product = sim.AddObject(ObjectType.MaterialStream, 250, 0, "tops")
ChemSep_col = sim.AddObject(ObjectType.CapeOpenUO, 200, 50, "Reactive Distillation Column")
bottom_product = sim.AddObject(ObjectType.MaterialStream, 250, 200, "bottoms")

sim.ConnectObjects(inlet.GraphicObject, ChemSep_col.GraphicObject, -1, -1)
sim.ConnectObjects(ChemSep_col.GraphicObject, top_product.GraphicObject, -1, -1)
sim.ConnectObjects(ChemSep_col.GraphicObject, bottom_product.GraphicObject, -1, -1)

sim.AutoLayout()
```

Figure 2.10: Creating and executing DWSIM simulation via python scripts

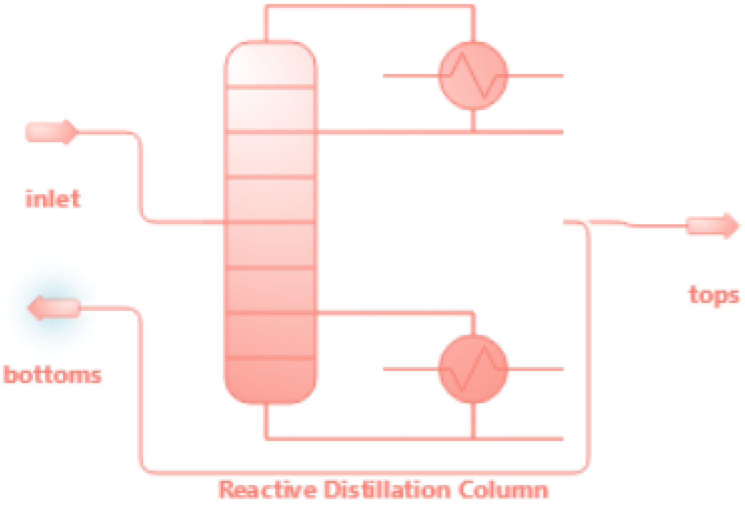


Figure 2.11: DWSIM flowsheet: Reactive distillation column

DWSIM® allows the importation of CAPE-OPEN applications and components to operate in a platform independent manner. Furthermore, the simulation environment is equipped with methodologies to enable inclusion and creation of custom models in order to accommodate for novel/unconventional unit operations. Currently, three alternatives are available to define custom models in DWSIM® (Figure 2.12). It is imperative to note that the custom models will have to be appended to the model library in prior to application.

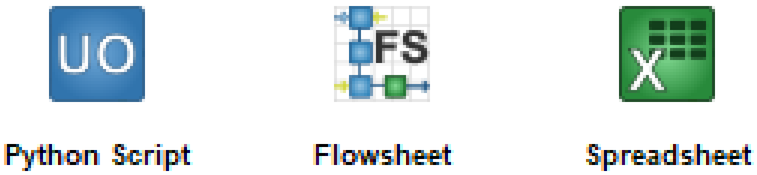


Figure 2.12: Custom modeling options available with DWSIM environment

2.4 PI Model Library & Physical Database

The model library suite consists of the thermophysical database for chemical compounds, and validated short-cut and rigorous equipment models for various intensified reaction and/or separation systems. It brings together various software platforms, such as ASPEN®, gPROMS®, and DWSIM®, to harness the available models and enable their multi-purpose use for process simulation, optimization, and analysis.

2.4.1 Equipment database

Currently, the model library contains ASPEN® and gPROMS® models for reactive distillation and dividing wall column (Figure 2.13). This interface is linked with corresponding excel files created for a convenient exchange of input/output data using ASPEN workbook functionality. Figure 2.14 illustrates the input interface for a reactive distillation column for MTBE production from methanol and isobutylene [13].

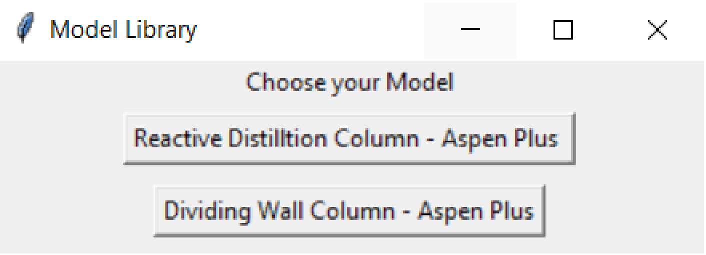
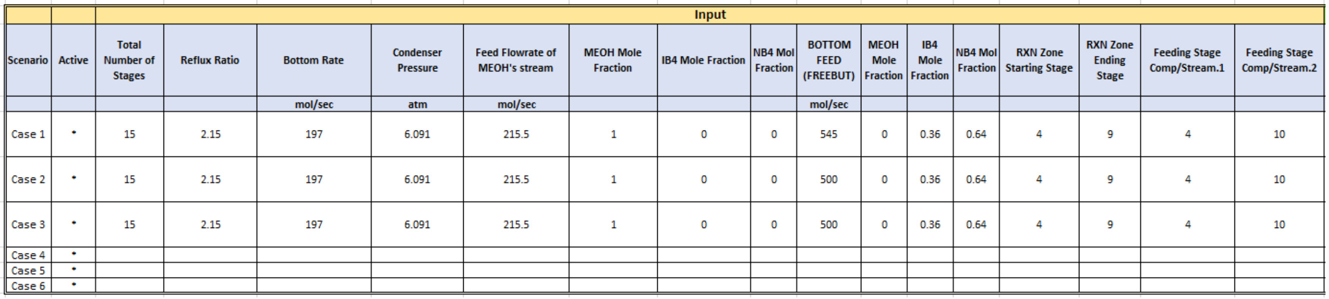
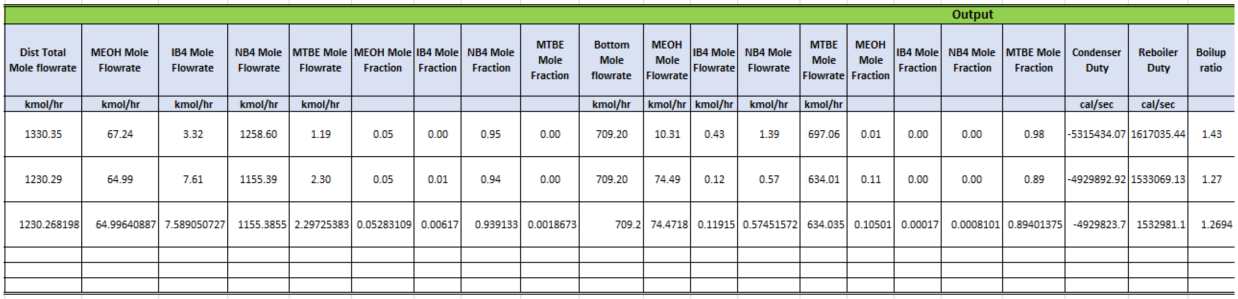

Model Library
Choose your Model
Reactive Distilltion Column - Aspen Plus
Dividing Wall Column - Aspen Plus

Figure 2.13: ASPEN® models available in Model Library



		Input																
Scenario	Active	Total Number of Stages	Reflux Ratio	Bottom Rate	Condenser Pressure	Feed Flowrate of MEOH's stream	MEOH Mole Fraction	IB4 Mole Fraction	NB4 Mol Fraction	BOTTOM FEED (FREEBUT)	MEOH Mole Fraction	IB4 Mole Fraction	NB4 Mol Fraction	RXN Zone Starting Stage	RXN Zone Ending Stage	Feeding Stage Comp/Stream.1	Feeding Stage Comp/Stream.2
				mol/sec	atm	mol/sec				mol/sec							
Case 1	•	15	2.15	197	6.091	215.5	1	0	0	545	0	0.36	0.64	4	9	4	10
Case 2	•	15	2.15	197	6.091	215.5	1	0	0	500	0	0.36	0.64	4	9	4	10
Case 3	•	15	2.15	197	6.091	215.5	1	0	0	500	0	0.36	0.64	4	9	4	10
Case 4	•																
Case 5	•																
Case 6	•																

Figure 2.14: Model Library: Reactive distillation column input



Output																				
Dist Total Mole flowrate	MEOH Mole Flowrate	IB4 Mole Flowrate	NB4 Mole Flowrate	MTBE Mole Flowrate	MEOH Mole Fraction	IB4 Mole Fraction	NB4 Mole Fraction	MTBE Mole Fraction	Bottom Mole flowrate	MEOH Mole Flowrate	IB4 Mole Flowrate	NB4 Mole Flowrate	MTBE Mole Flowrate	MEOH Mole Fraction	IB4 Mole Fraction	NB4 Mole Fraction	MTBE Mole Fraction	Condenser Duty	Reboiler Duty	Boilup ratio
kmol/hr	kmol/hr	kmol/hr	kmol/hr	kmol/hr					kmol/hr	kmol/hr	kmol/hr	kmol/hr	kmol/hr					cal/sec	cal/sec	
1330.35	67.24	3.32	1258.60	1.19	0.05	0.00	0.95	0.00	709.20	10.31	0.43	1.39	697.06	0.01	0.00	0.00	0.98	-5315434.07	1617035.44	1.43
1230.29	64.99	7.61	1155.39	2.30	0.05	0.01	0.94	0.00	709.20	74.49	0.12	0.57	634.01	0.11	0.00	0.00	0.89	-4929892.92	1533069.13	1.27
1230.268198	64.99640887	7.589050727	1155.3855	2.29725383	0.05283109	0.00617	0.939133	0.0018673	709.2	74.4718	0.11915	0.57451572	634.035	0.10501	0.00017	0.0008101	0.89401375	-4929823.7	1532981.1	1.2694

Figure 2.15: Model Library: Reactive distillation column output

Once the input is completed, the simulation can be executed and the results will be shown in the output interface (Figure 2.15). Furthermore, the calculated stage temperature data will be automatically plotted and displayed (Figure 2.16).

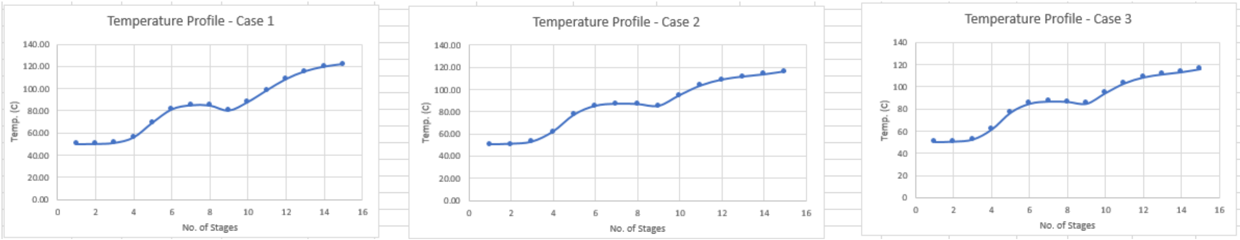
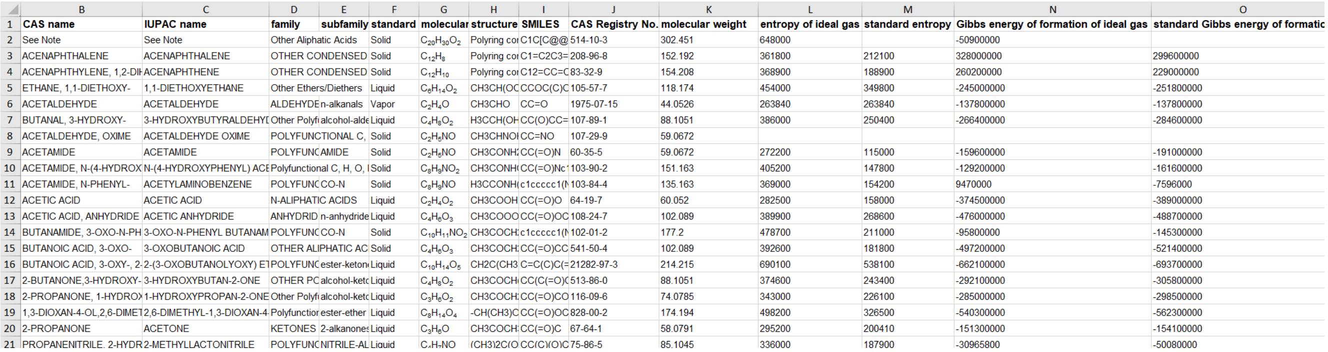


Figure 2.16: Model Library: Reactive distillation column stage temperature plot

Moreover, unit equipment models included in DWSIM® and ChemSep® can be utilized to translate module-based solution into equipment-based alternatives (Section 2.3).

2.4.2 Thermophysical property database

The thermophysical database, developed as a part of DIPPR® (Design Institute for Physical Properties) project, contains values for 34 constant and 15 temperature-dependent properties of chemical compounds. Figure 2.17 shows a subset of compounds and corresponding properties available within the excel database. The data is accessed while defining the thermodynamic and physical properties of chemical compounds. The excel file allows the capability to add a new or missing chemical compound to the database, along with its properties, to be used within the process design.



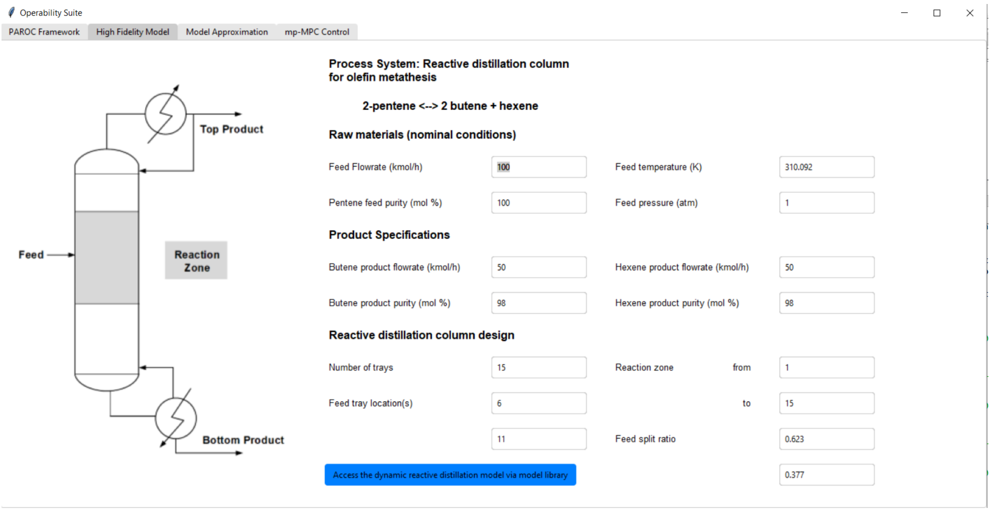
	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	**CAS name**	**IUPAC name**	**family**	**subfamily**	**standard**	**molecular**	**structure**	**SMILES**	**CAS Registry No.**	**molecular weight**	**entropy of ideal gas**	**standard entropy**	**Gibbs energy of formation of ideal gas**	**standard Gibbs energy of formatic**
2	See Note	See Note	Other Aliphatic Acids		Solid	$C_{20}H_{30}O_2$	Polyring cor	C1C[C@@	514-10-3	302.451	648000		-50900000	
3	ACENAPHTHALENE	ACENAPHTHALENE	OTHER CONDENSED		Solid	$C_{12}H_8$	Polyring cor	C1=C2C3=	208-96-8	152.192	361800	212100	328000000	299600000
4	ACENAPHTHYLENE, 1,2-DIH	ACENAPHTHENE	OTHER CONDENSED		Solid	$C_{12}H_{10}$	Polyring cor	C12=CC=C	83-32-9	154.208	368900	188900	260200000	229000000
5	ETHANE, 1,1-DIETHOXY-	1,1-DIETHOXYETHANE	Other Ethers/Diethers		Liquid	$C_6H_{14}O_2$	CH3CH(OC	CCOC(C)C	105-57-7	118.174	454000	349800	-245000000	-251800000
6	ACETALDEHYDE	ACETALDEHYDE	ALDEHYDE	n-alkanals	Vapor	C_2H_4O	CH3CHO	CC=O	1975-07-15	44.0526	263840	263840	-137800000	-137800000
7	BUTANAL, 3-HYDROXY-	3-HYDROXYBUTYRALDEHYD	Other Polyfu	alcohol-alde	Liquid	$C_4H_8O_2$	H3CCH(OH	CC(O)CC=	107-89-1	88.1051	386000	250400	-266400000	-284600000
8	ACETALDEHYDE, OXIME	ACETALDEHYDE OXIME	POLYFUNCTIONAL C,		Solid	C_2H_5NO	CH3CHNOI	CC=NO	107-29-9	59.0672				
9	ACETAMIDE	ACETAMIDE	POLYFUNC	AMIDE	Solid	C_2H_5NO	CH3CONH2	CC(=O)N	60-35-5	59.0672	272200	115000	-159600000	-191000000
10	ACETAMIDE, N-(4-HYDROX	N-(4-HYDROXYPHENYL) ACE	Polyfunctional C, H, O, I		Solid	$C_8H_9NO_2$	CH3CONH(	CC(=O)Nc1	103-90-2	151.163	405200	147800	-129200000	-161600000
11	ACETAMIDE, N-PHENYL-	ACETYLAMINOBENZENE	POLYFUNC	CO-N	Solid	C_8H_9NO	H3CCONH(	c1ccccc1(N	103-84-4	135.163	369000	154200	9470000	-7596000
12	ACETIC ACID	ACETIC ACID	N-ALIPHATIC ACIDS		Liquid	$C_2H_4O_2$	CH3COOH	CC(=O)O	64-19-7	60.052	282500	158000	-374500000	-389000000
13	ACETIC ACID, ANHYDRIDE	ACETIC ANHYDRIDE	ANHYDRID	n-anhydride	Liquid	$C_4H_6O_3$	CH3COOO	CC(=O)OC	108-24-7	102.089	389900	268600	-476000000	-488700000
14	BUTANAMIDE, 3-OXO-N-PH	3-OXO-N-PHENYL BUTANAM	POLYFUNC	CO-N	Solid	$C_{10}H_{11}NO_2$	CH3COCH2	c1ccccc1(N	102-01-2	177.2	478700	211000	-95800000	-145300000
15	BUTANOIC ACID, 3-OXO-	3-OXOBUTANOIC ACID	OTHER ALIPHATIC AC		Solid	$C_4H_6O_3$	CH3COCH2	CC(=O)CC	541-50-4	102.089	392600	181800	-497200000	-521400000
16	BUTANOIC ACID, 3-OXY-, 2-	2-(3-OXOBUTANOLYOXY) ET	POLYFUNC	ester-keton	Liquid	$C_{10}H_{14}O_5$	CH2C(CH3	C=C(C)C(=	21282-97-3	214.215	690100	538100	-662100000	-693700000
17	2-BUTANONE,3-HYDROXY-	3-HYDROXYBUTAN-2-ONE	OTHER PC	alcohol-keto	Liquid	$C_4H_8O_2$	CH3COCH(	CC(C(=O)C	513-86-0	88.1051	374600	243400	-292100000	-305800000
18	2-PROPANONE, 1-HYDROX	1-HYDROXYPROPAN-2-ONE	Other Polyfu	alcohol-keto	Liquid	$C_3H_6O_2$	CH3COCH2	CC(=O)CO	116-09-6	74.0785	343000	226100	-285000000	-298500000
19	1,3-DIOXAN-4-OL,2,6-DIMET	2,6-DIMETHYL-1,3-DIOXAN-4-	Polyfunctior	ester-ether	Liquid	$C_8H_{14}O_4$	-CH(CH3)C	CC(=O)OC	828-00-2	174.194	498200	326500	-540300000	-562300000
20	2-PROPANONE	ACETONE	KETONES	2-alkanones	Liquid	C_3H_6O	CH3COCH3	CC(=O)C	67-64-1	58.0791	295200	200410	-151300000	-154100000
21	PROPANENITRILE, 2-HYDR	2-METHYLLACTONITRILE	POLYFUNC	NITRILE-AL	Liquid	C_4H_7NO	(CH3)2C(O	CC(C)(O)C	75-86-5	85.1045	336000	187900	-30965800	-50080000

Figure 2.17: Thermophysical property database

2.5 PI Operability Suite

The operability suite offers the capability to perform model-based dynamic analysis, flexibility & safety assessment, and delivering control strategies for optimal controller design. High fidelity dynamic models, available in gPROMS®, are imported from the model library to achieve this task.

PI chemical processes are subjected to rigorous flexibility & risk assessments to ensure feasible operation under uncertain operating conditions,and evaluate safety performances of overall chemical process by considering individual equipment failure frequency and consequence severity [2]. The operability analyses can either be implemented during the synthesis stage for an integrated process design, and optimization with guaranteed flexibility and safety performances or via posterior operational analysis for a given PI process design.



Process System: Reactive distillation column for olefin metathesis

2-pentene <--> 2 butene + hexene

Raw materials (nominal conditions)

Feed Flowrate (kmol/h)	100	Feed temperature (K)	310.092
Pentene feed purity (mol %)	100	Feed pressure (atm)	1

Product Specifications

Butene product flowrate (kmol/h)	50	Hexene product flowrate (kmol/h)	50
Butene product purity (mol %)	98	Hexene product purity (mol %)	98

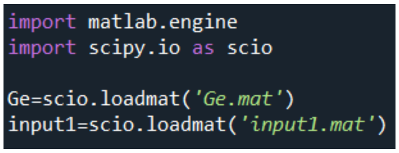
Reactive distillation column design

Number of trays	15	Reaction zone from	1
Feed tray location(s)	6	to	15
	11	Feed split ratio	0.623
			0.377

Figure 2.18: Operability suite: High fidelity model interface

The high fidelity dynamic model for reactive distillation column built on gPROMS® can be accessed through the interface shown in as Figure 2.18. The interface allows the model parameters to be altered before computing the dynamic response of the column.

The PAROC framework is employed for operational optimization, and explicit multi-parametric model predictive control (mp-MPC) of process designs. The framework, built on gPROMS® and MATLAB®, is connected to the interface via the *matlab.engine* and *scipy.io* libraries that enable transfer of information between python and MATLAB® scripts (Figure 2.19).

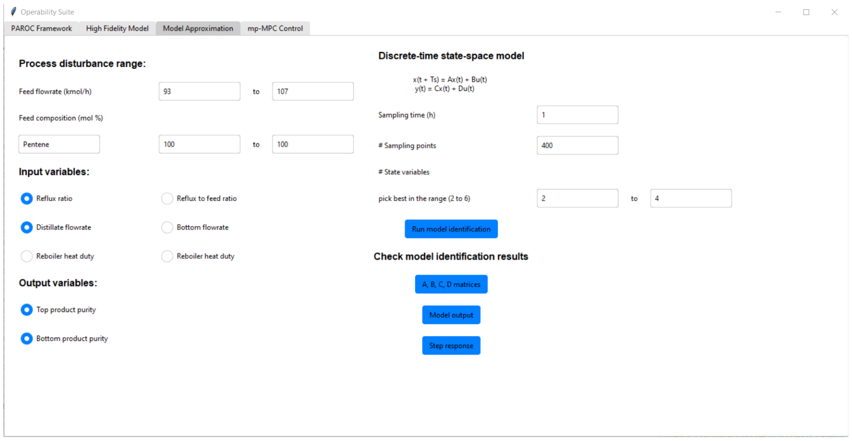


```
import matlab.engine
import scipy.io as scio

Ge=scio.loadmat('Ge.mat')
input1=scio.loadmat('input1.mat')
```

Figure 2.19: Operability suite: Establishing python-MATLAB® link

Model approximation is a crucial step within the PAROC framework prior to optimal controller design. Through system identification and model reduction techniques, the high-fidelity model is approximated to a reduced order linear state-space model that closely matches the output profile of the original model. Multiple estimated models can be obtained by varying the required number of state variables. These models remain available for user to inspect along with their output and dynamic response to a step change through the window shown in Figure 2.20.



Process disturbance range:

Feed flowrate (kmol/h) 93 to 107

Feed composition (mol %)

Pentene 100 to 100

Input variables:

● Reflux ratio ○ Reflux to feed ratio

● Distillate flowrate ○ Bottom flowrate

○ Reboiler heat duty ○ Reboiler heat duty

Output variables:

● Top product purity

● Bottom product purity

Discrete-time state-space model

x(t + Ts) = Ax(t) + Bu(t)
y(t) = Cx(t) + Du(t)

Sampling time (h) 1

Sampling points 400

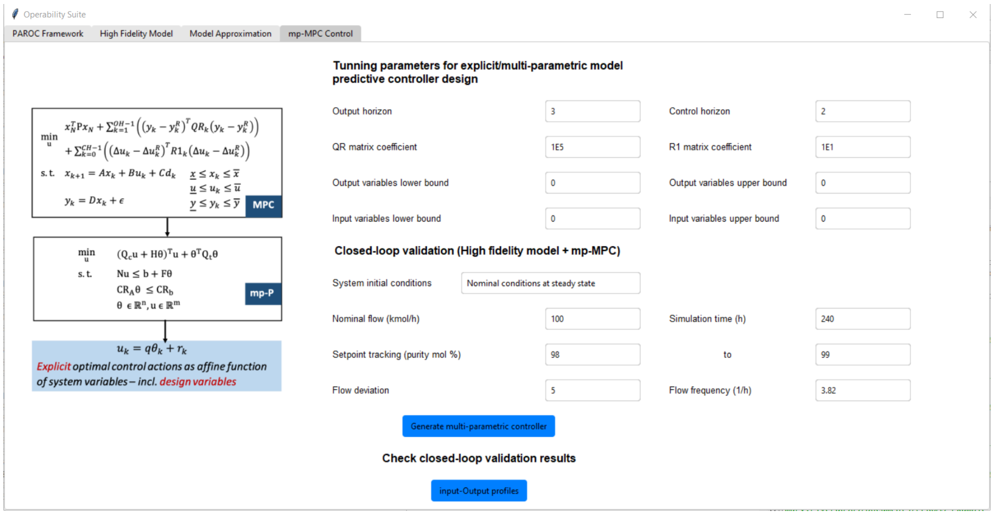
State variables

pick best in the range (2 to 6) 2 to 4

Check model identification results

Figure 2.20: Operability suite: Model Approximation

Following the model reduction, multi-parametric programming can be applied to the explicit model predictive control (MPC) problem. Tuning parameters for the controller design can be redefined and optimal solution is calculated through the provided *"Generate multi-parametric controller"* button (Figure 2.21). Moreover, *"Input-output profiles"* button initiates the validation of obtained solution in conjunction with the high-fidelity model in a closed-loop manner.



$$\min_{u} \; x_N^T P x_N + \sum_{k=1}^{OH-1}\left((y_k - y_k^R)^T QR_k (y_k - y_k^R)\right) + \sum_{k=0}^{CH-1}\left((\Delta u_k - \Delta u_k^R)^T R1_k (\Delta u_k - \Delta u_k^R)\right)$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k + Cd_k \quad \underline{x} \le x_k \le \overline{x}$$

$$\underline{u} \le u_k \le \overline{u}$$

$$y_k = Dx_k + \epsilon \quad \underline{y} \le y_k \le \overline{y}$$

$$\min_{u} \; (Q_c u + H\theta)^T u + \theta^T Q_t \theta$$

$$\text{s.t.} \quad Nu \le b + F\theta$$

$$CR_A\theta \le CR_b$$

$$\theta \in \mathbb{R}^n, u \in \mathbb{R}^m$$

$$u_k = q\theta_k + r_k$$

Explicit optimal control actions as affine function of system variables – incl. design variables

Tunning parameters for explicit/multi-parametric model predictive controller design

Output horizon 3 Control horizon 2

QR matrix coefficient 1E5 R1 matrix coefficient 1E1

Output variables lower bound 0 Output variables upper bound 0

Input variables lower bound 0 Input variables upper bound 0

Closed-loop validation (High fidelity model + mp-MPC)

System initial conditions Nominal conditions at steady state

Nominal flow (kmol/h) 100 Simulation time (h) 240

Setpoint tracking (purity mol %) 98 to 99

Flow deviation 5 Flow frequency (1/h) 3.82

Check closed-loop validation results

Figure 2.21: Operability suite: Explicit mp-MPC design

2.6 Remarks

The combination of these different suites constitutes a large collection of python, GAMS® and MATLAB® code files. For the convenience of access and enabling inter-file transfer of data, these files have been stored at the same location; however, they can be saved at several locations on the computer system.

To manage and enable the automated flow of information, a master file is essential through which all the other files along with their local attributes and variables can be accessed. To demonstrate such a hierarchical structure, consider the case for process synthesis as shown in Figure 2.22.

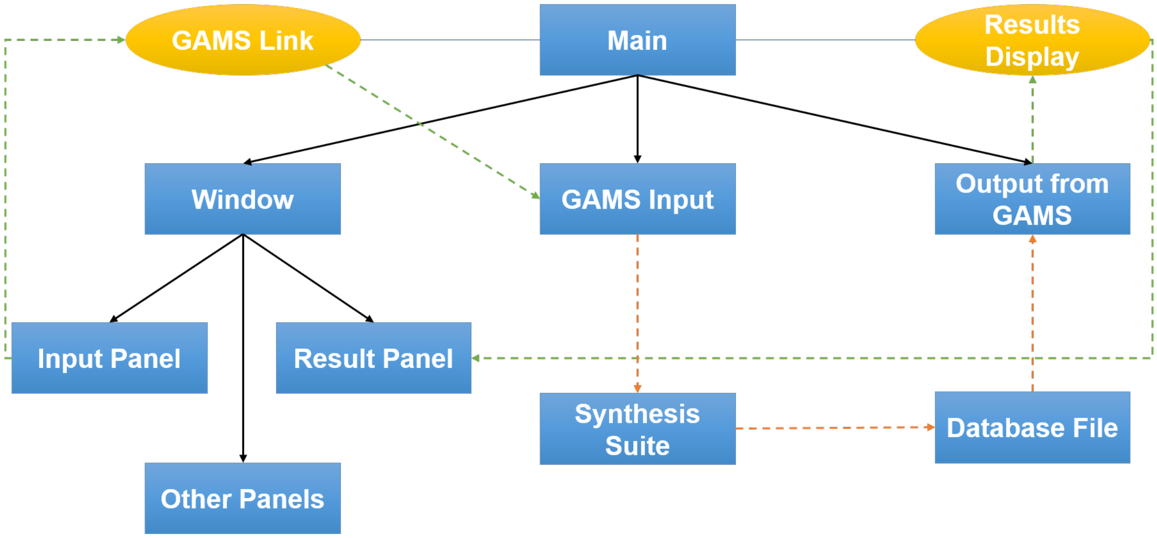


Figure 2.22: Python File Hierarchy

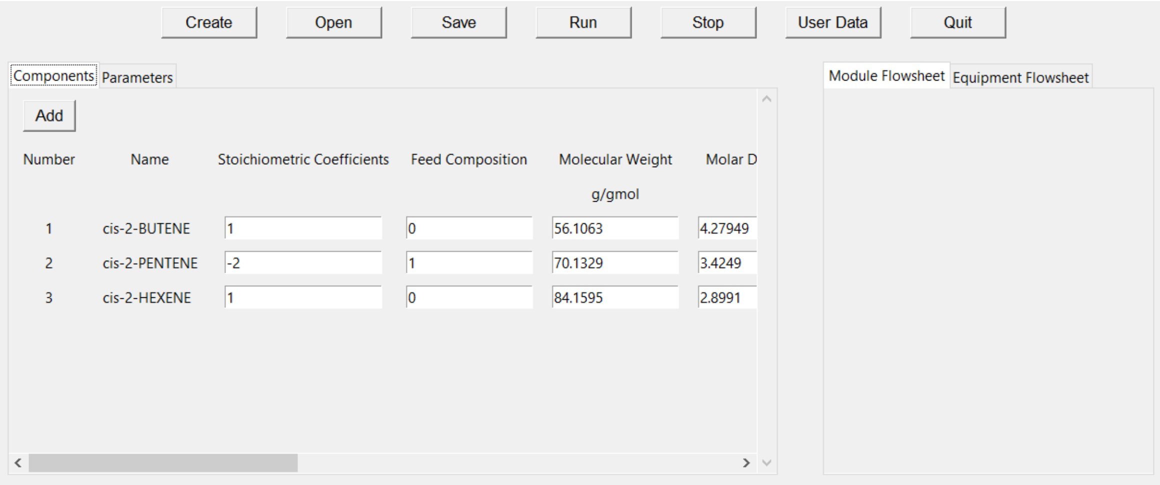
Here the *Main* file acts as a connecting node to enable and control the flow within the prototype. For instance, when user data transfer is initiated, the input panel activates the *GAMS Link* function provided in the master file to access the input file for GAMS (*GAMS input*). Subsequently, the *Results Display* function, available in the master file, is actuated to initiate the interpretation of database file and graphical results are created to be displayed in the results panel. The solid lines ($\longrightarrow$) denote the connection between python files and dashed lines indicate the direction of information flow within interface (green) and synthesis suite (orange)

3. CASE STUDY

To showcase the workings of the prototype we revisit the olefin metathesis problem[14] shown in equation 3.1.

$$2\,C_5H_{10} \rightleftharpoons C_4H_8 + C_6H_{12} \tag{3.1}$$

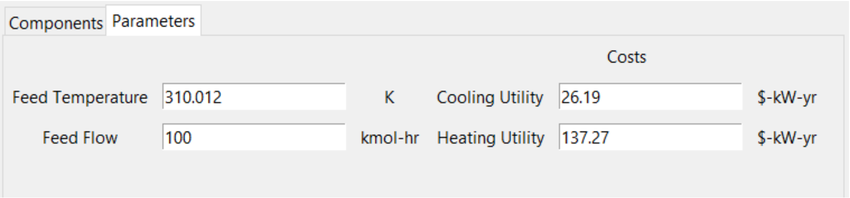
The metathesis reaction is equilibrium-limited and occurs homogeneously in liquid phase at atmospheric pressure. The interactions of components can be described with ideal vapor-liquid equilibrium [15]. In this example, the synthesis task is to produce 50 kmol/h of 98% of butene and 50 kmol/h of 98% of hexene at atmospheric pressure from an inlet stream of 100 kmol/h pure pentene at 310K.



Number	Name	Stoichiometric Coefficients	Feed Composition	Molecular Weight g/gmol	Molar D
1	cis-2-BUTENE	1	0	56.1063	4.27949
2	cis-2-PENTENE	-2	1	70.1329	3.4249
3	cis-2-HEXENE	1	0	84.1595	2.8991

Figure 3.1: Case Study: Chemical components selection

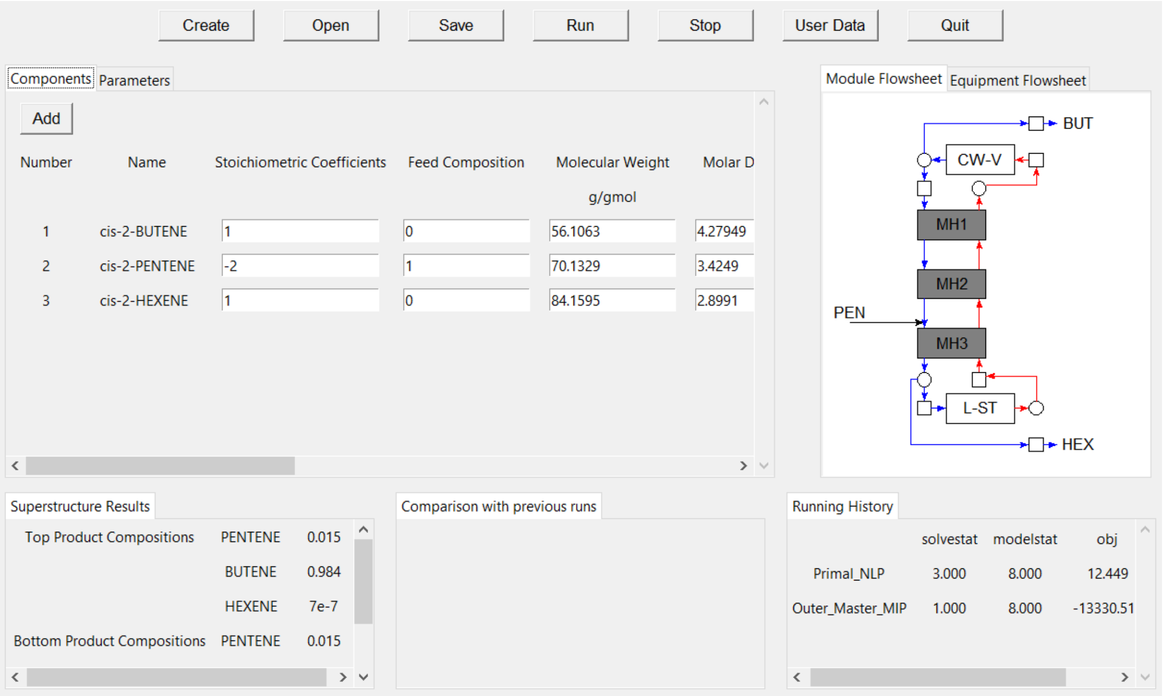
Figure 3.1 illustrates the components added to simulation by following the steps demonstrated in section 2.1.1. Furthermore, stoichiometric coefficients, feed specifications, and utility costs are entered into the corresponding fields (Figure 3.1 & 3.2)



			Costs		
Feed Temperature	310.012	K	Cooling Utility	26.19	$-kW-yr
Feed Flow	100	kmol-hr	Heating Utility	137.27	$-kW-yr

Figure 3.2: Case Study: Parameter definition

After completing the input, *Run* button is clicked to transfer the data to synthesis suite and export results from the database file (2.2). Figure 3.3 shows the optimal module-based flowsheet displayed within the interface.



Number	Name	Stoichiometric Coefficients	Feed Composition	Molecular Weight g/gmol	Molar D
1	cis-2-BUTENE	1	0	56.1063	4.27949
2	cis-2-PENTENE	-2	1	70.1329	3.4249
3	cis-2-HEXENE	1	0	84.1595	2.8991

Top Product Compositions	PENTENE	0.015
	BUTENE	0.984
	HEXENE	7e-7
Bottom Product Compositions	PENTENE	0.015

	solvestat	modelstat	obj
Primal_NLP	3.000	8.000	12.449
Outer_Master_MIP	1.000	8.000	-13330.51

Figure 3.3: Case Study: Optimal synthesis solution

The illustrated configuration in tandem with the superstructure results indicate the intensification of liquid-vapor mass transfer with the metathesis reaction within the intermediate (shaded) modules ans splitting of the inlet feed stream. Conclusively, the collective operation of these modules mimic a reactive distillation column.

Following the methodology laid out in section 2.3, a reactive distillation column is created using *ChemSep*® model available in DWSIM® through the CAPE-OPEN interface. However, the functionality to completely define the parameters of simulation through the python interface is currently in progress. Therefore, steady-state simulation of the reactive distillation column, shown in Figure 3.4, is reconfigured via the DWSIM® and ChemSep® environment.

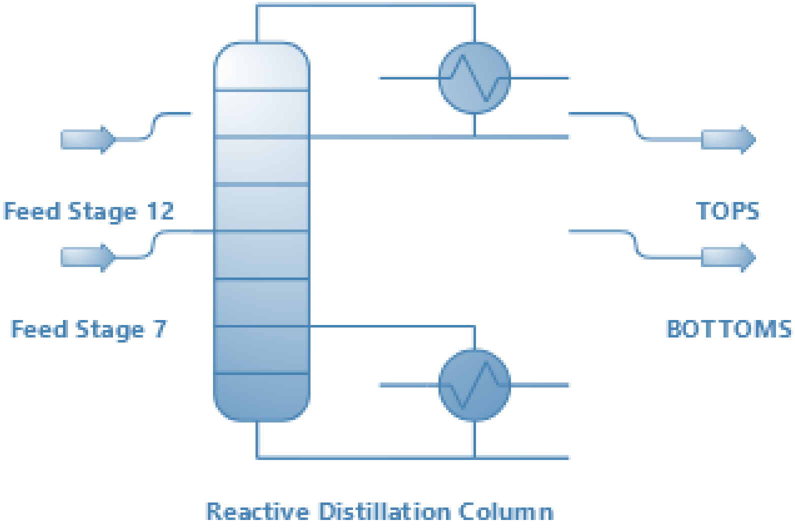
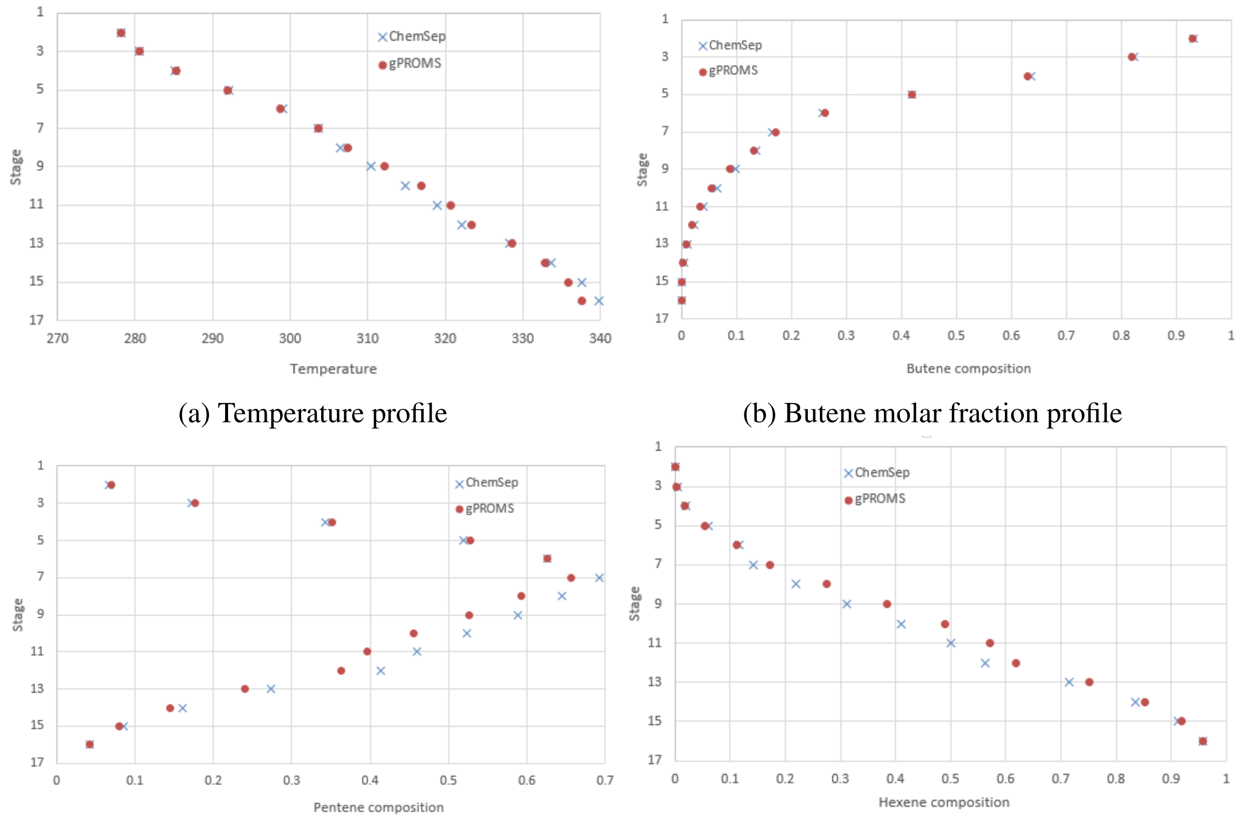


Figure 3.4: Simulation of reactive distillation column

The simulated results were cross-referenced against the solution obtained from a corresponding gPROMS® model for stage temperature and component molar fraction (Figure 3.5), and were found to be within acceptable margin of error. However, differences remain in utilized models for the two separate platforms. The reaction volume used for ChemSep® simulation was 8 m^3 as compared to 2 m^3 for the gPROMS® model.



(a) Temperature profile

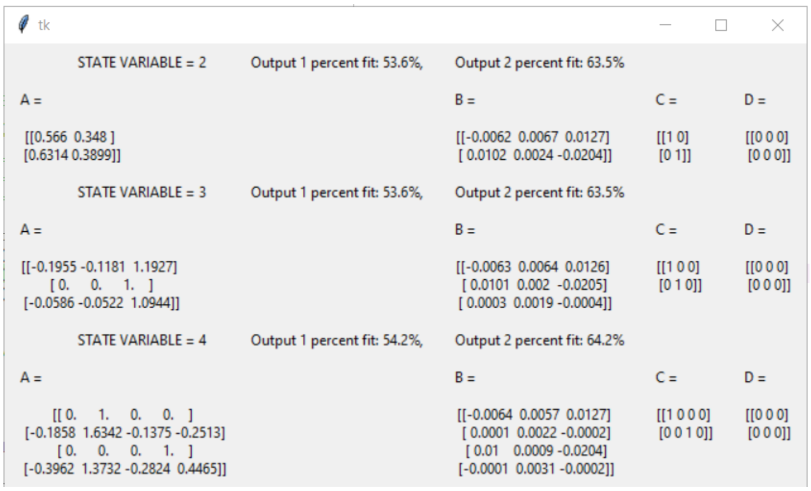
(b) Butene molar fraction profile

(c) Pentene molar fraction profile

(d) Hexene molar fraction profile

Figure 3.5: Comparison of simulation results: DWSIM® vs gPROMS®

Next, approximation techniques were applied on the high-fidelity gPROMS® model to obtain three estimates having two, three, and four state variables respectively. The obtained models along with their output accuracy percentages are displayed in 3.6.



```
        STATE VARIABLE = 2      Output 1 percent fit: 53.6%,    Output 2 percent fit: 63.5%

A =                                                      B =                        C =       D =

[[0.566  0.348 ]                                         [[-0.0062  0.0067  0.0127]  [[1 0]    [[0 0 0]
 [0.6314 0.3899]]                                         [ 0.0102  0.0024 -0.0204]]  [0 1]]   [0 0 0]]

        STATE VARIABLE = 3      Output 1 percent fit: 53.6%,    Output 2 percent fit: 63.5%

A =                                                      B =                        C =       D =

[[-0.1955 -0.1181  1.1927]                               [[-0.0063  0.0064  0.0126]  [[1 0 0]  [[0 0 0]
  [ 0.      0.      1.    ]                               [ 0.0101  0.002  -0.0205]   [0 1 0]]  [0 0 0]]
 [-0.0586 -0.0522  1.0944]]                               [ 0.0003  0.0019 -0.0004]]

        STATE VARIABLE = 4      Output 1 percent fit: 54.2%,    Output 2 percent fit: 64.2%

A =                                                      B =                        C =         D =

     [[ 0.      1.      0.      0.    ]                  [[-0.0064  0.0057  0.0127]  [[1 0 0 0]  [[0 0 0]
 [-0.1858  1.6342 -0.1375 -0.2513]                        [ 0.0001  0.0022 -0.0002]   [0 0 1 0]]  [0 0 0]]
      [ 0.      0.      0.      1.    ]                   [ 0.01    0.0009 -0.0204]
 [-0.3962  1.3732 -0.2824  0.4465]]                       [-0.0001  0.0031 -0.0002]]
```

Figure 3.6: Approximate models computed from high fidelity gPROMS® model

Given the various choices, it is recommended to move forward with the model displaying highest fit percentage and correct step change directions for unit step disturbance (Figure 3.7). In case of identical values and response behaviour, model having the least number of state variables is chosen. Considering the selection criteria the approximated model with 2 state variables was chosen for design of mp-MPC strategies.

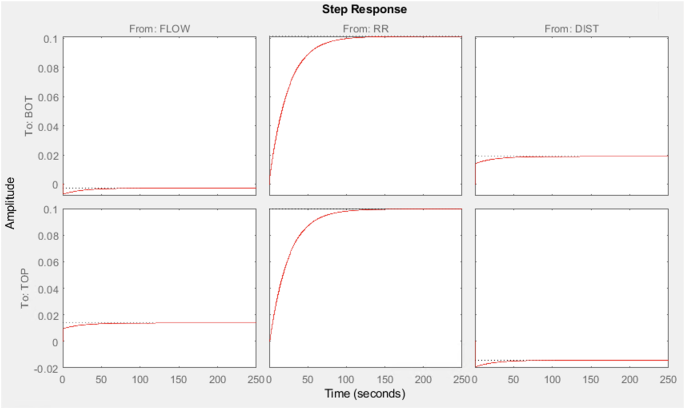


Figure 3.7: Step response for approximate model with 2 state variables

Lastly, to generate the optimal controller strategy, tuning parameters and system requirements (shown in Figure 2.21) were considered. Figure 3.8 displays the controlled output of composition of outlet streams as a result of feed disturbance of (+/-) 5 kmol/h and a disturbance frequency of 3.82/hr.

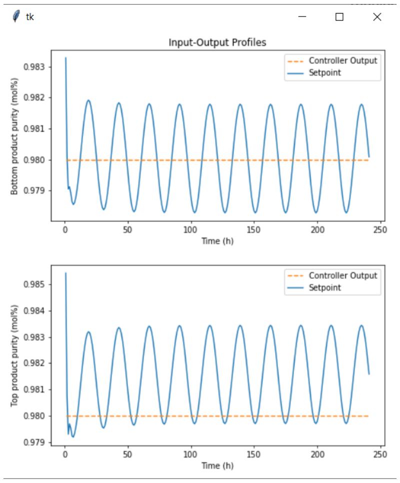


Figure 3.8: Closed loop validation results of the optimal controller

4. CONCLUSIONS AND FUTURE WORK

In this work, the framework and various abilities of the software prototype are highlighted. The prototype is capable of developing and delivering safe and operable PI systems through the application of an integrated design framework. User input can be deployed through the interface which effectively guides and controls the informational flow within different suites, and between the prototype and the user. The synthesis suite offers the potential of superstructure-based synthesis, integration, and intensification for automated design and optimization of PI process systems. Model library is programmed to perform steady-state validation of models available in ASPEN® and DWSIM®. Furthermore, it houses an exhaustive list of chemical compounds with their thermophysical properties to accurately capture the process model. The simulation suite has been developed to initiate the attempt towards a pseudo-automatic development of PI process alternatives. Finally, operability suite has been incorporated within the architecture for dynamic analysis, rigorous flexibility and safety assessment and to deliver optimal explicit mp-MPC strategies for the validated PI process flowsheet. The various functionalities of the prototype are showcased through synthesis, optimization, and control analysis of a pentene-metathesis reaction/separation system.

The ongoing work is targeted towards enhancement of the interface to include multiple feed streams and unit operations. Furthermore, the model library is being expanded to include high-fidelity steady state and dynamic models of various chemical processes for process simulation and model-based analyses. To improve the synthesis suite, strategies are being actively pursued for complete description and definition of selected unit operations via python scripts. Lastly, a link with gPROMS® model builder environment is currently being established to enable importation, modification, and execution of the available high-fidelity models via the interface.

REFERENCES

[1] J. Moulijn, A. Stankiewicz, J. Grievink, and A. Górak, “Process intensification and process systems engineering: A friendly symbiosis,” *Computers & Chemical Engineering*, vol. 32, pp. 3–11, 2008.

[2] Y. Tian, S. E. Demirel, M. F. Hasan, and E. N. Pistikopoulos, “An overview of process systems engineering approaches for process intensification: State of the art,” *Chemical Engineering and Processing - Process Intensification*, vol. 133, pp. 160–210, 2018.

[3] F. J. Keil, “Process intensification,” *Reviews in Chemical Engineering*, vol. 34, no. 2, pp. 135–200, 2018.

[4] J. G. Segovia-Hernández and A. Bonilla-Petriciolet, *Process Intensification in Chemical Engineering*. Springer, 2016.

[5] J. Bielenberg and I. Palou-Rivera, “The RAPID Manufacturing Institute – Reenergizing US efforts in process intensification and modular chemical processing,” *Chemical Engineering and Processing - Process Intensification*, vol. 138, pp. 49 – 54, 2019.

[6] A. K. Tula, J. Wang, X. Chen, S. S. Mansouri, and R. Gani, “ProCACD: A computer-aided versatile tool for process control,” *Computers & Chemical Engineering*, vol. 136, p. 106771, 2020.

[7] A. K. Tula, D. K. Babi, J. Bottlaender, M. R. Eden, and R. Gani, “A computer-aided software-tool for sustainable process synthesis-intensification,” *Computers & Chemical Engineering*, vol. 105, pp. 74–95, 2017.

[8] D. C. Miller, J. D. Siirola, D. Agarwal, A. P. Burgard, A. Lee, J. C. Eslick, B. Nicholson, C. Laird, L. T. Biegler, D. Bhattacharyya, N. V. Sahinidis, I. E. Grossmann, C. E. Gounaris, and D. Gunter, “Next generation multi-scale process systems engineering framework,” *Computer Aided Chemical Engineering*, vol. 44, pp. 2209 – 2214, 2018.

[9] S. E. Demirel, J. Li, and M. F. Hasan, “Systematic process intensification using building blocks,” *Computers & Chemical Engineering*, vol. 105, pp. 2–38, 2017.

[10] K. P. Papalexandri and E. N. Pistikopoulos, “Generalized modular representation framework for process synthesis,” *AIChE Journal*, vol. 42, pp. 1010–1032, 1996.

[11] E. N. Pistikopoulos, N. A. Diangelakis, R. Oberdieck, M. M. Papathanasiou, I. Nascu, and M. Sun, “PAROC – An integrated framework and software platform for the optimisation and advanced model-based control of process systems,” *Chemical Engineering Science*, vol. 136, pp. 115 – 138, 2015.

[12] Y. Tian, I. Pappas, B. Burnak, J. Katz, and E. N. Pistikopoulos, “A systematic framework for the synthesis of operable process intensification systems – Reactive separation systems,” *Computers & Chemical Engineering*, vol. 134, p. 106675, 2020.

[13] Y. Tian and E. N. Pistikopoulos, “Synthesis of operable process intensification systems – Steady-state design with safety and operability considerations,” *Industrial & Engineering Chemistry Research*, vol. 58, no. 15, p. 6049–6068, 2019.

[14] Y. Tian and E. N. Pistikopoulos, “Toward an envelope of design solutions for combined/intensified reaction/separation systems,” *Industrial & Engineering Chemistry Research*, vol. 59, no. 24, p. 11350–11354, 2020.

[15] R. C. Reid, J. M. Prausnitz, and B. E. Poling, “The properties of gases and liquids,” 1 1987.

[16] A. K. Tula, M. R. Eden, and R. Gani, “Computer–aided process intensification: Challenges, trends and opportunities,” *AIChE Journal*, vol. 66, no. 1, 2019.

[17] M. Skiborowski, “Process synthesis and design methods for process intensification,” *Current opinion in chemical engineering*, vol. 22, pp. 216–225, 2018.

[18] Q. Chen, E. S. Johnson, J. D. Siirola, and I. E. Grossmann, “Pyomo.GDP: Disjunctive Models in Python,” *Computer Aided Chemical Engineering*, vol. 44, pp. 889–894, 2018.

[19] E. N. Pistikopoulos, M. M. F. Hasan, J. S. Kwon, M. J. Realff, F. Boukouvala, M. R. Eden, S. Cremaschi, B. J. Tatarchuk, J. B. Powell, L. Spanu, R. Bindlish, and S. Leyland, "SYN-OPSIS – Synthesis of Operable Process Intensification Systems. RAPID Institute Project 9.3, DE-EE0007888-09-03," 2020.

APPENDIX A

CONFERENCE PROCEEDING

A.1 Introduction

Process Intensification (PI) has attracted an emerging interest in the chemical engineering research community and the chemical process industry owing to its potential ability to drastically increase process profitability and efficiency [1, 2]. The past few decades have witnessed significant advances in the field of PI [3, 4]. However, a widely used process intensification commercial software is still lacking [16, 17]. Driven by several recent national initiatives such as the RAPID Institute [5], the academic community has initiated attempts to develop software prototypes for computer-aided process design, synthesis, and intensification leveraging state-of-the-art process systems engineering approaches [7, 9, 18, 8].

In this paper, we present the development towards a software prototype based on our recently proposed framework for synthesis of operable process intensification systems [12]. The prototype systematically generates intensified process systems by utilizing a novel phenomenological synthesis approach with embedded process operability, safety, and explicit model predictive control analysis criterion. Section 2 of the paper elucidates on the prototype architecture and the function of embedded suites. In section 3, a case study of a reactive separation system is presented to demonstrate the working of the prototype. The remaining section covers conclusions and future directions.

A.2 Software Prototype

The prototype platform consists of three suites, namely: (i) Synthesis Suite, (ii) Simulation Suite, and (iii) Operability & Control Suite. As illustrated in Figure A.1, the user interface (UI) built on Python brings these suites together in a seamless manner to provide a consolidated environment while allowing user input and interaction. The different functional suites take advantage of various commercially available software packages, such as GAMS® for process synthesis and optimization, ASPEN PLUS® for steady-state process simulation, and gPROMS® for dynamic modeling and control analysis. These suites work in tandem to provide key insights to the user and are coordinated through the UI as detailed below.

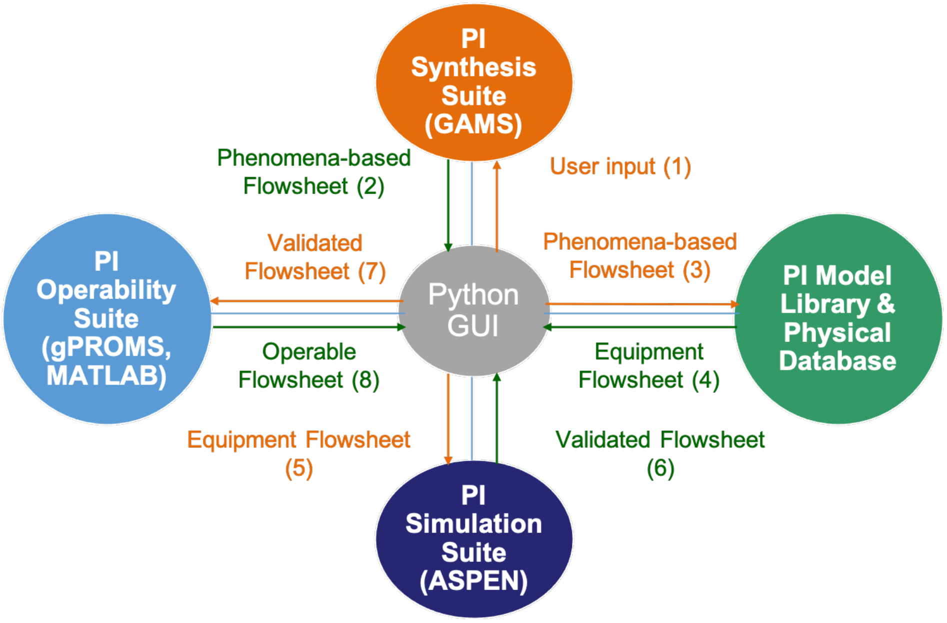

PI Synthesis Suite (GAMS)
Phenomena-based Flowsheet (2)
User input (1)
Validated Flowsheet (7)
Phenomena-based Flowsheet (3)
PI Operability Suite (gPROMS, MATLAB)
Python GUI
PI Model Library & Physical Database
Operable Flowsheet (8)
Equipment Flowsheet (4)
Equipment Flowsheet (5)
Validated Flowsheet (6)
PI Simulation Suite (ASPEN)

Figure A.1: Information flow chart for the software prototype platform

A.2.1 User Interface

The interface is programmed to facilitate communication and the transfer of data between the suites of the software prototype. Through the UI, users can select individual suites for targeted equipment/flowsheet intensification, design or analysis. However, the UI can also guide users to navigate between different suites in a step-by-step manner for integrated design with operability, safety, and/or control. To this end, the UI acts as a central node for information flow as shown in Figure A.1, and provides key input and output information to the user at every stage.

A.2.2 Synthesis Suite

The synthesis suite provides the tools required for the systematic generation of optimal and intensified process solutions. The technical foundation of the suite is laid by the Generalized Modular Representative Framework (GMF) [10]. To characterize various chemical processes, GMF employs two phenomenological modules, namely, the pure heat exchange module and the multi-functional mass/heat exchange module. Interlinked configurations of these modules, generated as a result of solving an optimization problem, can form the basis for the creation of conventional and/or novel unit operations. The optimization problem is formulated as a mixed integer nonlinear programming problem (MINLP), and solved using the Generalized Benders Decomposition algorithm in GAMS®. The parameters of the model are assumed to be deterministic and provided by the user through the UI. To exchange information between the UI and the synthesis suite, Python Application Programming Interface (API) is used. More detail on GMF modular representation and model formulation can be found in [14].

A.2.3 Simulation Suite

To validate the resulting configuration, the GMF module based flowsheet is converted to a corresponding equipment based flowsheet. This translation is achieved through a model library which comprises of information pertaining to various process equipment, and a set of rules governing the assignment of equipment to a module or a group of modules. The user can avail from a library of models and suggestions provided by the database to chose the equipment . This allows the prototype to account for equipment constraints which lends the flexibility to generate alternative flowsheets. It is imperative to note that the accuracy of translation will depend on the extensiveness of the library database. Novel equipment will have to be appended to the library in prior to achieve the desired translation. Subsequently, the equipment based flowsheet is simulated using ASPEN PLUS® to perform steady-state validation. Furthermore, high fidelity models are developed to fully capture and analyze the process dynamics.

A.2.4 Operability Suite

Model-based analyses are currently enabled for the following PI operational considerations: (i) flexibility analysis to ensure feasible operation under process uncertainty, (ii) risk analysis to evaluate the inherent safety performance of the resulting process configuration at conceptual design stage, and (iii) explicit/multi-parametric model predictive control to deliver optimal dynamic operation strategies under disturbances following the PAROC (PARametric Optimisation and Control) framework [11]. These operability analysis approaches can also be integrated with the above Synthesis Suite and Simulation Suite to simultaneously generate optimal and intensified process designs with guaranteed operability,safety, and control performance [12].

A.3 Case Study: Pentene Metathesis Reaction

A.3.1 Problem Statement

In this section, we revisit the problem of 2-Pentene (C_5H_{10}) metathesis to form 2-Butene (C_4H_8) and 2-Hexene (C_6H_{12}) which is an equilibrium limited reaction adapted from [14]. At atmospheric pressure, the reaction takes place in the liquid phase, and can be described by ideal vapor-liquid equilibrium (VLE). The production target is to obtain 50 kmol/h of 98$\%$ butene and 50 kmol/h of 98$\%$ hexene from a saturated liquid feed stream of 100 kmol/h pure pentene at atmospheric pressure. The objective is to design a process with the minimum utility cost.

A.3.2 User Input

The UI window is built on Python with the help of the *tkinter* package. It allows creation and management of window attributes like panels, buttons, and entry fields. The UI window comprises of dedicated panels for showing input/output data, run time data, superstructure layout, and comparison between alternatives of generated flowsheet. To allow for the dynamic updates of UI panels, they are created as *objects* of python defined *classes*. Users can access all functionalities of the prototype with the help of the *button* attribute. For the case study, the required input information includes physical properties like molecular weight and molar mass, thermodynamic information such as Antoine and heat capacity coefficients, reaction information such as stoichiometry and standard Gibbs energy of formation, utility costs, and input feed conditions.

A.3.3 Process Synthesis

To generate the module based flowsheet, the maximum number of modules was set to 5. This number can be increased to acquire more information about the identified unit operations or decreased to reduce the computational time. The optimal MINLP solution includes values for the number of pure heat and mass/heat exchanger modules (integer variables), presence of connections between stream and modules (binary variables), and the operating conditions (continuous variables). The results are exported to a database file (.db) to create modular structures using Python scripts. To visualize the solution, conditional statements are used to read the binary values and create its corresponding module or stream. For example, a module exists if the associated binary variable is assigned the value of 1, or 0 otherwise.

The resulting graphical module based flowsheet with the information on the constituent phenomena is shown in Figure A.2 . The modules and text in the flowsheet are sized according to the number of modules, and the size of the allocated window. Additionally, the python scripts account for the optimal layout to display process streams without overlapping with the modules. The red and blue lines show hot and cold stream condition respectively. The pure heat exchange module represented by white blocks, at top and bottom can be translated into condenser and reboiler respectively. The mass/heat exchange modules can be translated into trays of a reactive distillation column.

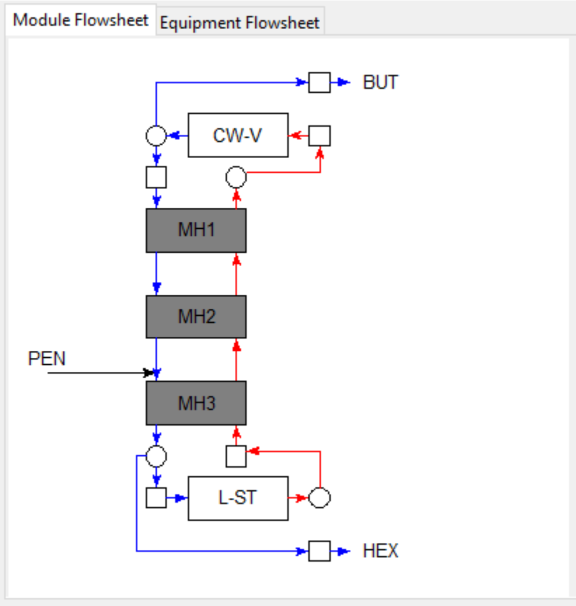


Figure A.2: Module flowsheet for pentene metathesis reaction

A.3.4 Remarks

The resulting GMF modular solutions are then identified and translated to equipment-based process alternatives with the help of a specialized PI model library developed as part of the SYNOPSIS Project [19]. The model library consists of validated rigorous and short-cut models for various intensified reaction and/or separation systems, including but not limited to, advanced distillation columns and reactive distillation columns. Moreover, the model library integrates different commercial software platforms (e.g., Aspen, gPROMS, Python) to leverage the existing unit operation models as well as to enable the flexible use of models for different computational purposes (e.g., simulation, optimization, control) in a platform-independent manner.

The safety, operability, and control performance of the PI systems can be further analyzed using the model-based metrics introduced in Section 2.4. This can be achieved either via posterior operational analysis for a given intensified process design, or via integrated process design optimization with operability, safety, and control considerations to systematically generate optimal process structures with desired operational performances.