Novel Adsorptive Reactor Configurations: Fundamental Conceptualisation for Design and Operation

Zur Erlangung des akademischen Grades eines

Dr.-Ing.

von der Fakultät Bio- und Chemieingenieurwesen
der Technischen Universität Dortmund
genehmigte Dissertation

M. Sc. Mamoon Hussainy

geboren in

Taez

Tag der mündlichen Prüfung: 18.01.2023

1. Gutachter: Prof. Dr. David W. Agar

2. Gutachter: Prof. Dr. Christoph Held

Dortmund 2023

ACKNOWLEDGEMENT

I wish to convey my heartfelt gratitude to all those who have played a significant role in the successful completion of this dissertation.

First and foremost, I am profoundly grateful to my esteemed supervisor, Prof. Dr. David W. Agar. His never-ending support, expert guidance, and continuous motivation have been invaluable throughout this research project. His wisdom and mentorship have shaped my understanding and propelled me to new heights.

I would also like to express my sincere appreciation to the distinguished members of my dissertation committee, Prof. Dr. Held, Prof. Dr. Vogt, and Prof. Dr. Lucia. Their valuable insights, thought-provoking discussions, and constructive feedback have contributed immensely to the refinement of this work.

My gratitude extends to the staff of the chemical reaction engineering chair who have engaged in enlightening conversations.

To my beloved parents, wife, and children, I cannot express my gratitude enough for your boundless love, understanding, and encouragement. Your unwavering belief in my abilities and sacrifices made on my behalf have been the bedrock of my accomplishments and made this endeavour more rewarding.

Lastly, I extend my heartfelt appreciation to the participants of this study, whose willingness to share their insights and experiences has been invaluable. Your contributions have added depth and importance to this research.

I am immensely thankful to all those who have contributed to my academic journey, enabling me to reach this important milestone.

Mamoon Hussainy

ABSTRACT

There has been considerable interest in recent years in “intensified processes” that allow for the substantial amelioration of chemical processes in terms of equipment dimensions, costs, and safety. The integration of an additional separative functionality into chemical reactors can be used to manipulate the concentration and temperature profiles and thus dramatically enhance reactor performance. The resultant improvements in conversion and selectivity can, in turn, simplify or even eliminate the downstream processing necessary. Adsorptive reactors, in which adsorptive and reactive functionalities are combined, represent a promising example of the bifunctionality in industrial chemical reactors.

In this research work, intensive yet comprehensive multiscale and multidimensional modelling and simulation studies have been conducted dealing systematically with the relations of the available degrees of freedom to one another and to the performance of adsorptive fixed bed reactors. The goal was to obtain know-how-oriented strategies to maximise the performance of adsorptive reactors. Amongst the several degrees of freedom available in design and operation of adsorptive reactors, it was found that the spatial distribution of the adsorptive and catalytic functionalities at the reactor level (macrostructuring) and the temperature profiling over the reactor length have been shown to be decisive factors for maximising adsorptive reactor performance.

Considering the industrially-relevant Claus and Deacon reactions as test cases, two novel designs have been proposed, the multilevel isothermal and the central isothermal sandwich designs, by which a multi-fold performance improvement compared to the corresponding isothermal and adiabatic simple uniform structure adsorptive reactor designs could be attained even with incorporating the regeneration process necessary, where the cyclic steady state was calculated based on the direct substitution method. The improvements in space time yields obtained by the foresaid novel designs were respectively 700% and 650% for Claus reaction and 35-fold and 18.5-fold for Deacon reaction. The overall feasibility of these novel designs can be envisaged in the light of the considerable cost reduction, which compensate for the extra costs required for the technical realisation of the proposed designs, achieved by avoiding the expensive tail-gas treatment processes in case of Claus reaction or by simplified downstream processing in case of Deacon reaction.

Zusammenfassung

In den letzten Jahren hat es ein beträchtliches Interesse an "intensivierten Prozessen" gegeben, die es ermöglichen, chemische Prozesse in Bezug auf Anlagengröße, Kosten und Sicherheit erheblich zu verbessern. Die Integration einer zusätzlichen Trennfunktion in chemische Reaktoren kann dazu genutzt werden, die Konzentrations- und Temperaturprofile zu manipulieren und so die Reaktorleistung drastisch zu verbessern. Die sich daraus ergebenden Verbesserungen bei Umsatz und Selektivität können wiederum die nachgeschaltete Verarbeitung vereinfachen oder sogar überflüssig machen. Adsorptive Reaktoren, in denen adsorptive und reaktive Funktionalitäten kombiniert werden, stellen ein vielversprechendes Beispiel für die Bifunktionalität in industriellen chemischen Reaktoren dar.

In dieser Forschungsarbeit wurden intensive und zugleich umfassende mehrskalige und mehrdimensionale Modellierungs- und Simulationsstudien durchgeführt, die sich systematisch mit den Beziehungen der verfügbaren Freiheitsgrade zueinander und zur Leistung von adsorptiven Festbettreaktoren befassten. Ziel war es, Know-how-orientierte Strategien zur Leistungsmaximierung von adsorptiven Reaktoren zu erhalten. Unter den verschiedenen Freiheitsgraden, die bei der Auslegung und dem Betrieb von adsorptiven Reaktoren zur Verfügung stehen, haben sich die räumliche Verteilung der adsorptiven und katalytischen Funktionalitäten auf der Reaktorebene (Makrostrukturierung) und die Temperaturprofilierung über die Reaktorlänge als entscheidende Faktoren für die Maximierung der adsorptiven Reaktorleistung herausgestellt.

Unter Berücksichtigung der industriell relevanten Claus- und Deacon-Reaktionen als Fallbeispiele wurden zwei neuartige Designs vorgeschlagen, das mehrstufige isotherme und das zentrale isotherme Sandwich-Design, durch die eine mehrfache Leistungsverbesserung im Vergleich zu den entsprechenden isothermen und adiabatischen adsorptiven Reaktordesigns mit einfacher einheitlicher Struktur erreicht werden konnte, selbst wenn der notwendige Regenerationsprozess einbezogen wurde, bei dem der zyklische stationäre Zustand auf der Grundlage der direkten Substitutionsmethode berechnet wurde. Die Verbesserungen der Raum-Zeit-Ausbeuten, die durch die genannten neuen Designs erzielt wurden, betrugen 700% bzw. 650% für die Claus-Reaktion und das 35-fache bzw. 18,5-fache für die Deacon-

Reaktion. Die allgemeine Durchführbarkeit dieser neuartigen Konzepte kann angesichts der beträchtlichen Kostenreduzierung, die die für die technische Realisierung der vorgeschlagenen Konzepte erforderlichen Mehrkosten aufwiegt, ins Auge gefasst werden, die durch die Vermeidung der teuren Abgasbehandlungsverfahren für Claus-Reaktion oder durch die vereinfachte Downstream-Verarbeitung für Deacon-Reaktion erreicht wird.

CONTENTS

NOMENCLATURE

b	[Pa^{-1}]	Adsorption affinity
C	[$kmol.m^{-3}$]	Concentration
$C_{H2O,eq}$	[$kmol.m^{-3}$]	Equilibrium water concentration
d	[m]	Diameter
D	[$m^2.s^{-1}$]	Mass dispersion coefficient
D_{μ}	[$m^2.s^{-1}$]	Micropore diffusivity
$D_{\mu,\infty}$	[$m^2.s^{-1}$]	Micropore diffusivity at infinite temperature
d_K	[$m^2.s^{-1}$]	Knudsen diffusion coefficient
d_m	[$m^2.s^{-1}$]	Molecular diffusivity
d_0	[$m^2.s^{-1}$]	Macropore diffusion coefficient
$E_{A,diff}$	[kJ/kmol]	Activation energy of micropore diffusion
K	[Pa^{-1}]	Inhibition constant
$k_{i,film}$	[m/s]	Mass transfer coefficient
$k_{1,Claus}$	[$kmol.kg^{-1}.s^{-1}.Pa^{-1.17}$]	Forward rate constant for Claus reaction
$k_{1,Deacon}$	[$kmol.kg^{-1}.s^{-1}.Pa^{-1.5}$]	Forward rate constant for Deacon reaction
$k_{2,Claus}$	[$kmol.kg^{-1}.s^{-1}.Pa^{-1}$]	Backward rate constant for Claus reaction
k_{LDF}	[1/s]	Linear driving force coefficient for the mass transfer
L	[m]	Reactor length
P	[Pa]	Pressure; Partial pressure
Pr	[-]	Prandtl number
q	[$kmol.kg^{-1}$]	Adsorbate loading
$\overline{q}$	[$kmol.kg^{-1}$]	Average adsorbate loading
r	[$kmol.kg^{-1}.s^{-1}$]	Rate of reaction or adsorption
r_c	[m]	Zeolite crystal radius
Re	[-]	Reynolds number
Sc	[-]	Schmidt number

T	[K]	Temperature
t	[s]	Time
u	[$m.s^{-1}$]	Superficial velocity of the gas phase
X	[-]	Conversion
x	[m]	Axial coordinate
y	[-]	Mole fraction

Greek symbols

α	[$W.m^{-2}.K^{-1}$]	Heat transfer coefficient
β	[$m^2.m^{-3}$]	Ratio of area to volume of a solid particle
φ	[-]	Volume fraction of catalyst, adsorbent, or PCM
ΔH	[$J.mol^{-1}$]	Enthalpy change
ε	[-]	Bed void fraction or porosity
κ	[-]	Equilibrium constant
λ	[$W.m^{-1}.K^{-1}$]	Heat Conductivity
Λ	[$W.m^{-1}.K^{-1}$]	Heat dispersion coefficient
μ	[Pa.s]	Viscosity
ν	[-]	Stoichiometric coefficient
ρ	[$kg.m^{-3}$]	Density
τ	[s]	Cycle time
χ	[-]	Approach to equilibrium

Subscripts

A	Adsorption
ads	Adsorbent
avg	Average
ax	Axial
cat	Catalyst

DP	Desired product; S_8 for Claus reaction and Cl_2 for Deacon reaction
eq	Equilibrium
f	Feed
g	Gas phase
i	Species *i* in the gas phase
j	Segment number
min	Minimum
p	Particle
PCM	Phase change material as functional pellets in the reactor
R	Reaction or reactor
reac	Reaction-adsorption phase
ref	Reference conditions
reg	Regeneration
s	Solid phase
sat	Saturation
Tot	Total

Abbreviations

AR	Adsorptive Reactor
LDF	Linear Driving Force
STY	Space Time Yield
PCM	Phase Change Material
RAR	Reaction – Adsorption – Reaction arrangement
IAR	Integrated Adsorptive Reactor

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The deliberate manipulation of concentration and temperature profiles through the incorporation of additional separative phenomena provides a powerful tool for enhancing reactor performance. In this chapter, the concept of multifunctional reactors is introduced with a description of the principle of adsorptive reactors and a literature review on the advances in this regard followed by presenting the two test cases adopted in this study.

1.1. Multifunctional reactors

As the name implies, multifunctional reactors are reactor concepts involving simultaneously at least one further functionality, either mass or thermal or both, in addition to the existing reactive one. Unambiguously speaking, the additional functionalities integrated to the reactive one in the same equipment should not be directly linked to the reaction. Thus, heterogeneously catalysed reactor systems cannot be classified as multifunctional reactors since the involved mass and heat transport processes are inherent components of the heterogeneous catalytic reactions. The optimally integrated functionalities can lead to improved performance of the multifunctional reactor and increase the economic efficiency of the overall process.

The concept of “multifunctional reactor” was first introduced by Agar and Ruppel in 1988 [1]. In this review article, the authors proposed classifying multifunctional reactors based on two criteria; firstly, the dominant transport process whether it is convective or diffusive and secondly the type of internal or external sources or sinks of mass and

heat. Later, another classification method based on adding a separative functionality into the reactive one has been proposed [2]. This method, however, leads to only bifunctional reactor concepts and thus does not allow identifying and designing new reactor concepts other than bifunctional types. Furthermore, this method of classification considerably impedes the comparative analogies between the different types of multifunctional reactors. Therefore, Agar [3] proposed as yet the most insightful classifying methodology of multifunctional reactors based on the functional interactivity between the participating phases as shown in Table 1. The unique advantage of this classification model is that it enables the consideration of the multifunctionality not only at the reactor-level but also at the pellet-level. Table 1 shows an example of possible resulting multifunctional processes.

Table 1: Classification of multifunctional reactors according to [3]

Type of the multifunctional reactor	Participating phases
Reactive adsorption	Gas-solid
Reactive absorption	Gas-liquid
Reactive chromatography	Liquid-solid
Reactive distillation	Gas-liquid-solid

Although the integration of different functionalities (especially the separative and reactive) offer a variety of advantages, as it will be mentioned in this section, this functionality integration causes at the same time a reduced number of the degrees of freedom according to the Gibbs phase rule depending on the number of components, phases, and reactions involved. It is therefore mandatory to pre-assess the possible applicability of the intended integration of different functionalities and to define the operating variables that have a limiting influence on the design and operation of the

integrated process. Examples of such operating variables are temperature, pressure, velocity, residence time, etc.

As it can be seen in Figure 1, considering two operating variables with a limiting influence, each of the reactive and separative functionality has its own operating window within which the temperature and pressure can be optimised independent of the other process. By integrating these two functionalities, the system loses degrees of freedom, and the operating window becomes narrower. This means in turn that more efforts in the design process are needed and the modelling of the integrated process is more demanding than before.

It must be noted that the possible applicability of a multifunctional reactor is subjected to the existence of a common operating window of all functionalities to be integrated. Otherwise, the reactor concept is not feasible.

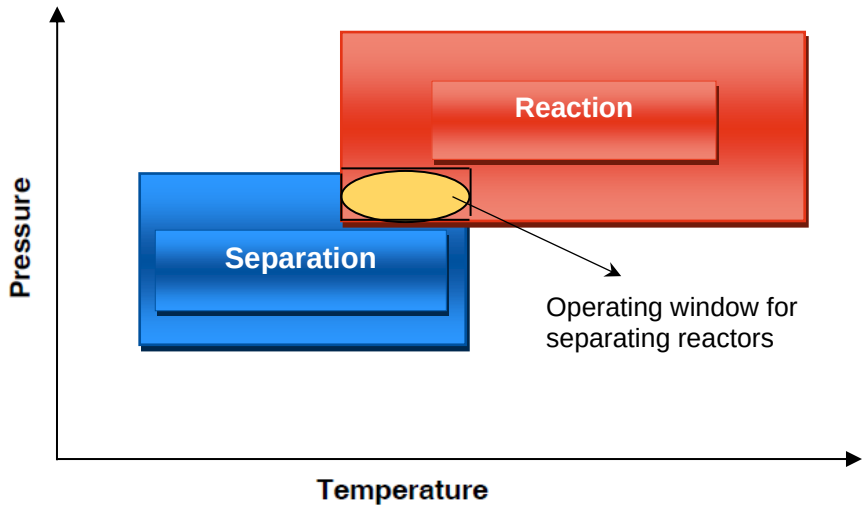


Figure 1: Illustrative sketch for operating window of separating reactors

Integration of the different functionalities within a chemical reactor can be applied for several purposes, some of which are given below as examples:

- Increasing the forward conversion of reversible, equilibrium-limited reactions.
- Increasing the selectivity of consecutive or parallel reactions through targeted manipulation in the concentration and temperature profiles.

- Avoidance of energy-consuming separation operations for product/unreacted educt recovery.

The resulting benefits of applying the appropriate multifunctional reactor concepts are multifarious, generally the following potentials can be attained:

- lowering the recycle flows or reduction of the necessary process steps in multi-stage form due to the higher conversions achieved.
- Reduction of the feed amount by increasing the selectivity.
- Minimisation of energy consumptions, possibly also of investment costs.
- Elimination of downstream processing steps for value product upgrading.

1.2. Adsorptive reactors: bifunctional reactor technology

1.2.1. Conceptual principle of adsorptive reactors

The integration of an adsorptive functionality into an existing reactive one allows to selectively remove or introduce certain species from/into the reaction mixture leading to an enhanced reactor performance (such as space-time-yield, selectivity, conversion, or overall process economics) as a result of deliberately modified concentration and temperature profiles along the reactor. The adsorptive reactor principle is applicable for many existing heterogeneously catalysed gas-phase reactions that can be operated in the temperature window from 200 °C to 300 °C [4].

For a simple equilibrium-limited conversion of the form: A + B ⇆ C + D, the *in situ* adsorptive removal of the by-product D from the reaction medium favours the formation of the desired product C (Figure 2). By adsorbing D rather than C, one circumvents unwanted side-reactions of the adsorbate and facilitates the desorption process needed to periodically regenerate the adsorbent.

During the operational progress of adsorptive reactors, the concentration profiles along the reactor can be distinguishably categorised in three zones:

- **Inlet zone:** At the reactor inlet, the adsorbent becomes fully loaded by the adsorbate and thus the reaction only proceeds up to the equilibrium state.
- **Middle zone:** The reaction rate slows down at the end of the inlet zone where the reaction reached the equilibrium. Consequently, a transition between fully loaded and unloaded adsorbent starts to form in this zone leading ideally to enhance the main reaction to proceed beyond the equilibrium value due to the simultaneous removal of one of the products.
- **Outlet zone:** In the outlet zone of the reactor, the reaction proceeds to complete conversion as the adsorbent is almost completely unloaded.

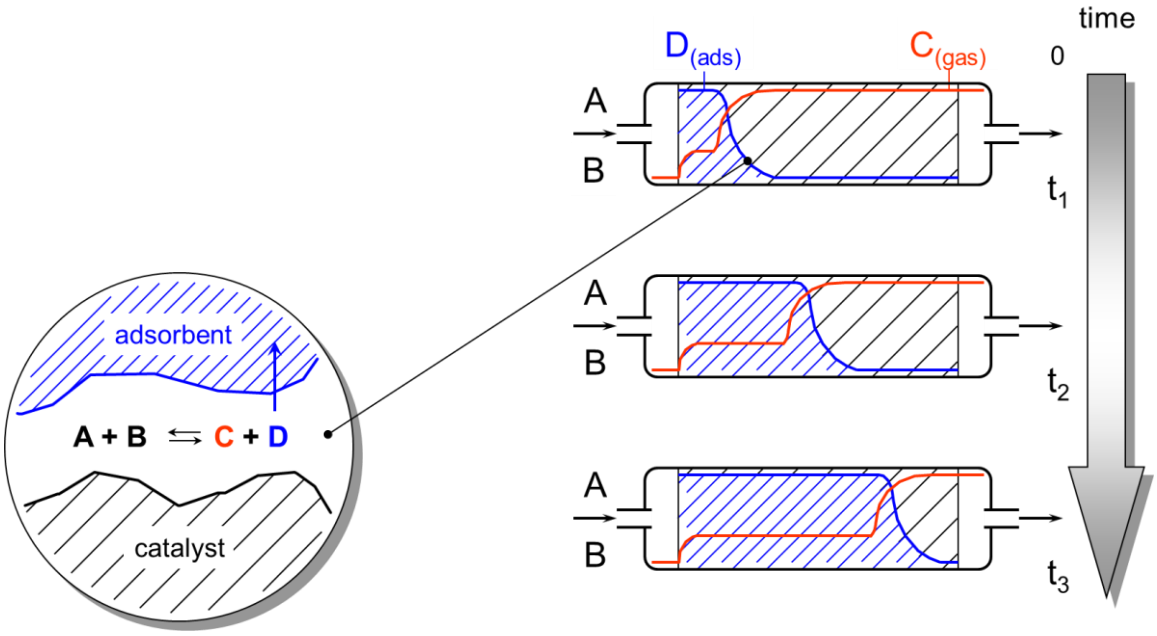


Figure 2: Adsorptive reactor concept for enhancing the conversion of reversible reactions (the concentrations' profiles shown on the right are at time instants t1, t2, t3 > 0)

The reaction front moves with the time down the adsorptive reactor through the transition middle zone and outlet zone until it finally breaks through and the reaction only takes place up to the equilibrium conversion. This reveals one of the most important characteristics of adsorptive reactors, namely the inherently discontinuous mode of operation. Effectively, the adsorptive fixed bed reactor can only be used until

shortly before breakthrough, after which the adsorbent must be regenerated. Due to the necessary alternation between reaction/adsorption and regeneration processes, multiple adsorptive reactors in parallel are necessary to establish a continuous operation of the reaction unit of the overall process.

The simplest design of adsorptive reactors consists of a homogeneous mixture of catalyst and adsorbent. Other reactor configurations with structured packings, which have locally different catalyst and adsorbent fractions, as well as moving and fluidised beds are also conceivable.

As a result of the gas-solid interactions (physisorption, chemisorption, gas-solid reaction), the gas phase composition is continuously changed until the equilibrium state is reached. At the equilibrium state, the adsorption and desorption rates are equal, so that macroscopically no change in the composition of the gas phase can be observed. Therefore, the knowledge of the temporal change of the sorption capacity is of decisive importance for the investigation of adsorptive reactor concepts.

The influence of the integrated adsorptive functionality on the side reactions of certain processes depends on whether the adsorbate is a product or a reactant in the side reaction equation and whether the main reaction is exothermic or endothermic. Hence, if the adsorbate is formed as a product in the side reaction, removing it by the integrated adsorptive functionality will enhance the side reaction progress and vice versa. In case of an endothermic equilibrium reaction system, the removal of the adsorbate by integrated adsorption allows comparable conversions to be achieved at considerably lower temperatures, which prove to be kinetically and thermodynamically disadvantageous for the undesired side reactions.

In order to establish an economically viable adsorptive reactor concept compared to the corresponding conventional process, an appropriate adsorbent should be developed that must meet certain requirements with respect to adsorption kinetics and capacity as well as to regenerative procedure. Furthermore, the adsorbent should be chosen to be highly selective to only one chemical component involved in the process. The removal of more than one component inevitably reduces the effectiveness of adsorptive reactors, since the adsorption of several components entails subsequent, usually costly treatment, which makes the overall process more cost intensive.

Therefore, development of selective adsorbents with the highest possible capacity is a crucial step in the design of adsorptive reactors.

Typical adsorption processes can be classified into three groups depending on the regeneration procedure adopted, as well known from adsorption technology [5, 6]:

- **Temperature swing**

In the case of stronger binding forces between the adsorbent and adsorbate molecules, as the case in chemisorption, regeneration by increasing the temperature and the resulting higher desorption rate is particularly effective. The heat required for the thermal regeneration can, at least partially, be taken from other parts of the overall process flowsheet where the heat is produced and is going to be dissipated. The disadvantage of temperature swing technology is the thermal inertia of the system built over the time and the resulting longer holding times. Additionally, not only the adsorbent particles should be thermally stable, but also the adsorbate itself since the adsorbed organic molecules in particular tend to decompose at high temperatures.

- **Pressure swing**

Another possibility to perform the desorption process is by reducing the operating pressure of the system. Increased pressure values favour adsorption, whenever these high values still lie within the applicable range of the involved chemical reaction scheme. During the regeneration step of the operation cycle, the applied pressure is decreased, and the adsorbed molecules are released into the gas phase again. Pressure swing is the simplest technique to implement the regeneration process and is thus the widest applied technology in industrial adsorption processes [5].

- **Flushing with an inert**

If a small reduction in partial pressure is sufficient for regeneration, this can be achieved by displacement desorption with the aid of an inert substance. From an economic point of view, this type of regeneration is only profitable if no further treatment of the adsorbed component is necessary and if an appropriate flushing medium is available.

- **Displacement purge**

Desorption of adsorbates occurs when a more preferentially adsorbed species replaces the original adsorbate [6]. However, the displacement adsorbate should adhere to the adsorbent surface with a comparable strength to the original adsorbate to ensure successful desorption. If the displacement adsorbate binds too strongly, it can pose challenges in removal from the adsorbent, potentially affecting the regeneration process. As a result, the net heat generated or consumed in the adsorbent remains close to zero, thereby maintaining a relatively constant adsorbent temperature throughout the cycle.

1.2.2. Comparable reactor technologies to adsorptive reactors

In addition to the adsorptive reactor concept as an upcoming bifunctional reactor technology, two other related reactor concepts, namely the chromatographic reactors (CR) and the membrane reactors (MR), are highlighted in this section and compared to adsorptive reactors considering three categories: the functional principle, suitability for heterogeneously catalysed gas-solid reactions, and economical and technical aspects.

a) Functional principle:

All the foresaid reactor concepts integrate a separative functionality into the reactive one and are thus all bifunctional reactors. Whereas the chromatographic reactors also involve an adsorption process like adsorptive reactors, the membrane reactors combine a component permeation as a separative process. Therefore, they all can be used to achieve higher conversions and selectivities (higher selectivities especially in case of adsorptive and membrane reactors rather than in chromatographic reactors).

Chromatographic reactors utilise the separation effect known from chromatography, in which a spatial separation of the individual reactants takes place while at the same time completely separating the products. This is due to the different interaction between the components (adsorbate) of the fluid phase and the stationary phase

(adsorbent). The choice of a suitable stationary phase with its specific adsorption properties as well as a suitable eluent flow results in different migration velocities of the peaks or fronts of individual components through the chromatographic single column. However, due to the much weaker interactions between the components to be adsorbed and the stationary phase, desorption already occurs under reaction conditions.

Membrane reactors are fixed bed catalytic reactors where the reactor walls are permeable. Alternatively, the tubular membranes can be catalytically active and thus the two functionalities can be directly integrated. In case of product removal, membrane reactors can be considered as a limiting case of adsorptive reactors with infinitely high adsorption capacity. In addition to the product removal benefits of membrane reactors, distributed feed of reactants is also a well-known application of membrane reactors to enhance the performance in terms of higher selectivity values.

In contrast to adsorptive and membrane reactors, adsorption of more than one component is an essential requirement for chromatographic reactors. However, the interaction between adsorbate and adsorbent should not be too strong to allow the necessary desorption by the eluent under reaction conditions. Thus, the equilibrium-limited, selective, and quantitative separation in adsorptive reactors differs from the kinetically controlled, non-selective, and reversible adsorption in chromatographic reactors.

b) Suitability for heterogeneously catalysed gas-solid reactions:

Due to the fact that chromatographic reactors involve complex necessary separation of reaction products from the eluent, there are only a few examples of applications for large-scale heterogeneously catalysed gas-phase syntheses, for instance hydrogenation of mesitylene (1, 3, 5-trimethylbenzene) [7], ammonia synthesis [8], and oxidative conversion of methane to ethene [9]. Therefore, chromatographic reactors have so far been widely used for the liquid-phase heterogeneously catalysed syntheses of fine and specialty chemicals.

In contrast to chromatographic reactors, the selective separation process in both adsorptive and membrane reactors as well as the thermal stability of the adsorbent in

case of adsorptive reactors or the membrane in case of membrane reactors, make these reactor concepts suitable for heterogeneously catalysed gas-phase reactions. Specifically speaking, the membrane reactors have been primarily studied for gas-phase reactions where hydrogen is separated from the reaction phase or, on the other hand, defined quantities of oxygen is spatially fed into the reactor. For the latter case, the heterogeneously catalysed partial oxidation of alkanes to the corresponding alkenes is a good technically relevant example. For the former case, overlaps in the application of adsorptive and membrane reactors occur as for the steam reforming of methane.

c) Economical and technical aspects:

In the light of above-discussed features of the three mentioned bifunctional reactor concepts, it turns out that the high dilution of the products and the subsequent unavoidable and time-consuming separation of reaction products from the eluent makes the application of chromatographic reactors for the technically relevant syntheses of gaseous products, which are mostly produced in larger quantities than basic chemicals, not reasonable/feasible in comparison to liquid-phase syntheses. This point is valid and decisive for all studied chromatographic reactor designs, like the true or simulated moving bed reactors, and the rotating cylindrical annular reactor. Additionally, and especially in case of the moving bed chromatographic reactor, specific problems, like abrasion and mechanical stress due to solids' movement and fluid dynamics, occur during the countercurrent operation which is then necessary for the continuous production in large quantities.

While the membrane reactors can be operated continuously, the adsorptive reactors are periodic in nature. Nevertheless, since that not all commercially available membranes are dense and selective, one of the main disadvantages of these reactors is the problems resulted by the poor selectivities. Moreover, slow permeation kinetics and expensive membrane materials are further decisive drawbacks of membrane reactors [10]. The former case, i.e. slow permeation, means to ensure largest possible membrane surface area to achieve the required permeation rate, since the effect of the permeation in membrane reactors is proportional to the membrane surface area, unlike the adsorptive reactors where the adsorption is proportional to the reactor

volume. In particular, ceramic membranes, typically used for heterogeneously catalysed gas-phase reaction systems, are difficult to seal and liable to microcracks at elevated temperatures which, together with other pinholes appeared during the normal operation, lead to reactant slips and consequently reduce the effectiveness of membrane reactors significantly [11].

As a summary to the discussed comparison above between the three mentioned bifunctional reactor concepts, the chromatographic reactors are unsuitable and less attractive for the technically relevant gas-solid reaction systems. In comparison to chromatographic reactors, adsorptive reactors have the following distinctions:

- involve gaseous reaction media,
- have a better-defined adsorptive discrimination,
- are not confined to exclusively elutive regeneration procedures, and
- include greater heat impacts, resulting in non-isothermal behaviour.

On the other hand, adsorptive and membrane reactors cover the same field of application. Whether a membrane reactor or an adsorptive reactor concept would ultimately be the better alternative technology for a certain application, this depends totally on the particular characteristics of the investigated reaction system. For the two reaction schemes considered in this thesis, Claus and Deacon reactions, the adsorptive reactor concept is more appropriate technology to be investigated.

1.3. State of the art of adsorptive reactors

The concept of adsorptive reactors, or alternatively found in literature as sorption-enhanced reaction processes (SERP), has been extensively studied for many existing heterogeneously catalysed gas-phase reactions in the temperature window between 200 °C and 300 °C [1, 4, 42, 44].

Since the pioneering work of Kuczynski et al. [12] on adsorptive methanol synthesis, considerable efforts have been devoted to ascertaining designs and operating modes for improving the performance of adsorptive reactors. For instance, simplified hydrogen production processes [13, 14], appropriate regeneration procedures, fixed bed

structuring strategies, and favourable concentration-temperature trajectories have been developed [15 - 22]. Furthermore, strategies to optimally distribute the functionalities at the macro- and micro-level and to “desorptive cooling”, in which desorption of an inert component from a previously loaded adsorbent in a mixed catalyst-adsorbent fixed bed is used to provide intensive pellet-scale heat removal, have been also proposed in the research-group of chemical reaction technology at the TU Dortmund University [23 - 35].

Especially in the recent years, the hydrogen production process has been intensively studied for the adsorptive reactor concept based on steam reforming process [36 – 50], gasification processes [51 - 61], or the water gas shift reaction [62, 63].

An integration of a sorption-enhanced steam reforming with water splitting process to produce hydrogen has been proposed by Saithong et al. [64]. They reported the influences of steam-to-fuel, sorbent-to-methane, and catalyst-to-methane ratios on the process efficiency. Li et al. [65] also proposed a sorption-enhanced staged gasification of biomass, in which the solids (ash and tar) formed by the biomass gasification are separated from the gaseous phase (syngas and methane) before entering the adsorptive steam reforming reactor. More recently, a comprehensive techno-economic analysis of three different reactor configurations (a packed-bed reactor, membrane reactor, and sorption-enhanced membrane reactor for carbon capture and high-purity hydrogen production) has been conducted by Lee et al. [66]. They found that the sorption-enhanced membrane reactor offers a trade-off between the carbon dioxide emission rates and the associated costs of the production process.

For any proposed adsorptive reactor design to be commercialised, it should overcome two main challenges viz., the need for periodic adsorbent regeneration in an expedient manner that does not adversely affect overall performance and the need for optimally utilised adsorptive capacities to provide reasonable cycle times. Like other common examples of cyclic fixed bed processes including temperature swing adsorption (TSA), pressure swing adsorption (PSA), and reverse flow reactors, determination and optimisation of the cyclic steady state, or periodic state, of adsorptive reactors are technically very important when developing new processes or improving already existing ones. The cyclic steady state of any periodic process operated by continuous

repeating of certain steps is reached when any state in the cycle at any point of the bed is identical to the previous cycle.

The process cycle design of adsorptive reactors and the behaviour of the cyclic steady state have been extensively addressed in literature. The working group of A. E. Rodrigues in Portugal designed three-step cyclic operation of the sorption-enhanced ethylene production process by dehydrogenation of ethane [16]. They also designed four- and five-step cycle for a sorption-enhanced operation of steam-methane reforming [21, 22]. They reported an improved performance over the conventional processes. Rawadieh and Gomes [67] reported a similar conclusion for a five-step cycle design of a sorption-enhanced steam reforming reactor. An improvement of about 200% over the conventional process has been also reported by Hufton et al. [13] for an adsorptive hydrogen production reactor. This periodic process consisted of five-step cycle including pressurisation, reaction/adsorption, depressurisation, and two successive purge steps.

Improved calculating algorithms and optimising routines are necessary to accelerate the solution process over the extremely time-consuming conventional techniques. For instance, cyclic steady states are typically calculated by the conventional and computationally infeasible successive substitution method, in which the process is simulated one cycle after another until convergence to the cyclic state is reached. Several approaches to speed up the convergence and subsequently reach the solution have been successfully proposed and implemented. A general method to derive shortcut models for fast cyclic processes was proposed by Gorbach et al. [68] which can be applied to a set of transient process equations converting them to a stationery reduced set of equations that can then be solved faster. Following this idea, two algorithms were proposed by Kolios et al. [69] to provide much faster solution process and facilitate efficient analysis and design of the system: the perturbation algorithm which its convergence is comparable to the Newton’s method on one hand, and the dual-grid algorithm on the other hand. Salinger and Eigenberger [70] conducted studies on cyclic reverse flow reactors and introduced a methodology to transform the initial-value problem into a steady state boundary-value problem in space and time. They reported the ability of their methodology to clearly identify the multiplicity region throughout the parameter space without varying the parameter values and initial

conditions as with the successive substitution method. For complex systems with a quite slow solution process, fast simple methods for the direct determination [71] of cyclic steady states will be helpful. Two types of direct calculation methods have been proposed. One is a full discretisation method [18, 72] in which the model equations are temporally and spatially discretised using either a finite difference method or a finite element method. To obtain accurate results, large discretisation dimensions in both the temporal and spatial domains are often required, resulting in a very large set of nonlinear algebraic equations [73]. The solution is then enforced towards cyclic steady state by imposing cyclic constraints to set bed conditions at the end and the beginning of each simulated cycle as identical. The second method for direct determination of periodic states is the direct substitution, in which, and unlike the first method, the nonlinear governing partial differential equations are only discretised in the spatial domain resulting in a set of ordinary differential equations that are integrated over time [73 - 75]. Recently, Munera-Parra et al. [76] studied the cyclic behaviour of the adsorptive retro shift water-gas reactor using both the direct determination methods mentioned above. They found that the direct substitution method gave enough resolution accuracy, within reasonable computer-time, to capture even sharp fronts expected for such system with highly dynamic nature. Stadler et al. [77] performed numerical simulations in MATLAB/Simulink of the dynamically operated adsorptive water-gas shift reactor and introduced a semi-discretisation method in combination with an ordinary differential equation solver implemented in Simulink. They reported a 15% increase of the sorbent usage for serial operation with adjusted switching times. Based on the 1-dimesional and pseudo-homogeneous mathematical model, Arora et al. [78] presented a generalized reaction-adsorption modelling and simulation framework to capture the complicated dynamics of adsorptive reactors using the foresaid reaction as an example, i.e. the water-gas shift reaction. They used this synthesis framework to optimise the cyclic operation of the sorption-enhanced methanol production process [79] and the adsorptive steam methane reforming reactor [80]. They reported an improved methanol yield of up to 87% and a 35% higher hydrogen productivity, respectively. Zachapolous et al. [81] performed detailed thermodynamic analysis and simulation of the adsorptive methanol production process and showed that by in-situ water removal, a 130 % higher methanol productivity in comparison to the direct hydrogenation process can be attained. Tian et al. [82]

conducted an extensive survey on process systems engineering approaches for process intensification and provided an overview of the achieved developments in reactive separation technologies.

While this previous work served to reveal the considerable promise of adsorptive reactors, it mostly failed to exploit all the degrees of freedom available in adsorptive reactor design and operation. Except for the temperature profiling and functionality structuring that has been individually studied, the combination of both or exploiting other techniques like the dynamic profiling of operating parameters, multiscale heat integration including the usage of PCM, or the distributed feed have been not considered in the previous work as far as the author is aware. Additionally, further developments of adsorptive reactor designs, system integration, and optimisation are still required to lift the technology readiness level (TRL) of the adsorptive reactor concept to the next steps. Thus, an attempt is made here to highlight the considerable potential of the deliberate coupling/deploying of the degrees of freedom available and to develop relevant and technically feasible design strategies.

1.4. Concrete reaction schemes as test cases

Adsorptive reactors have been proposed for a variety of industrially important reactions, which have been concisely reviewed elsewhere [4]. Table 2 summarises the most important reactions studied.

In the research group of chemical reaction engineering at the TU Dortmund University, several important and equilibrium-limited reactions applied in industry, such as the retro-shift conversion for functionalisation of carbon dioxide, hydrogen cyanide synthesis, and Claus process for synproportionation of hydrogen sulphide/sulphur dioxide, have been the subject of detailed experimental and theoretical studies [23 - 26, 82]. The common by-product of the foresaid reaction types is the water vapour, which can be *in situ* adsorbed on 3A zeolite simultaneously with the reaction to improve the performance indicated by the achieved conversion ratio.

It was found that the kinetics of the retro-shift reaction proved too slow at the lower operating temperatures between 200°C and 300 °C [83]. On the other hand, the

adsorptive HCN-synthesis exhibited excessive uncontrollable side-reactions. Therefore, only the Claus process was considered worthy of further investigations.

Another reaction scheme that was considered in this study, in addition to the Claus reaction, is the Deacon reaction which has been recently become an interesting candidate for the principle of adsorptive reactor operation. Whereas the temperature for the Claus reaction ($2H_2S + SO_2 \leftrightarrows 3/8S_8 + 2H_2O$) has always been compatible with the adsorption of water vapour on zeolites, the Deacon reaction ($4HCl + O_2 \leftrightarrows 2Cl_2 + 2H_2O$) has only lately become accessible for adsorptive reactors through the development of low temperature ruthenium-based catalysts [4, 95] in place of the chromium- and copper-systems previously available, which required reaction temperatures of well above 300°C.

Table 2: Different reaction types evaluated for the adsorptive reactors' principle

Reaction type	**Reference**
methanol synthesis	[12, 81]
gasification processes	[51 – 61]
steam reforming	[19, 21, 36 - 50, 86, 87]
oxidative coupling of methane	[88]
water-gas-shift reaction	[62, 63, 83, 85, 7 89]
Claus process	[90, 91]
hydrogen cyanide synthesis	[91, 92]
dehydrogenations	[72, 84, 93]

1.4.1. Claus reaction

Many of the refinery processes, such as reforming, isomerisation, alkylation, and polymerisation, are catalytic conversion processes on sulphur-sensitive catalysts. The increasing use of low-quality oil and gas means that ever larger amounts of sulphur must be removed during refining operations. To prevent poisoning of the catalyst, the sulphur compounds, amongst other poisons, must be removed. This is done by hydrotreating, i.e. catalytic hydrogenating desulphurisation (hydrodesulphurisation) followed by Claus process. The first operation produces hydrogen sulphide from the sulphur-containing compounds, which is then separated. Hydrodesulphurisation takes place at temperatures between 300 and 400 °C and at pressures between 2.5 and 6 MPa (25 - 60 bar). The catalysts are, sulphur-resistant cobalt/molybdenum supported catalysts, for instance. The feedstock is heated in a tubular furnace and then mixed with hydrogen and fed into a fixed-bed reactor. The reaction product is cooled, and the excess hydrogen is separated in a separator and recycled. The hydrogen sulphide is separated from the oil in a stripper column and the desulphurised product is removed from the column bottom. The hydrogen sulphide is converted into elemental sulphur in the three-stage Claus process, as shown in Figure 3. In a combustion chamber, the hydrogen sulphide is converted into sulphur dioxide and water with a less-content of oxygen; in three downstream catalytic fixed-bed reactors, sulphur dioxide then reacts with hydrogen sulphide via an exothermic comproportionation reaction to form sulphur and water vapour on Titania or Alumina catalysts. The produced elemental sulphur is then condensed out within several interstage condensation units, where the main gas stream needs afterwards to be reheated before it enters the catalytic Claus reactors. The resulting sulphur has a high degree of purity and can be used to produce sulphuric acid. This process described above represents a typical Claus process (known as straight-through Claus), where all the acid gas passes through the reaction furnace. However, when the hydrogen sulphide content in the gas is less than 50%, at most two thirds of the gas stream bypasses the reaction furnace to provide the necessary stoichiometric ratio 2:1 of the hydrogen sulphide to sulphur dioxide in the catalyst beds; the process is then called split-flow Claus.

The Claus reaction can be expressed in simplified manner as follows:

$$2H_2S + SO_2 \Leftrightarrow {}^{3}/_{8}\,S_8 + 2H_2O \qquad \Delta_R H = -108\ \mathrm{kJ/mol} \qquad (Eq.\ 1)$$

The by-product in Eq. 1, the water vapour, can be simultaneously adsorbed in the catalytic reactor stage on a suitable and commercially available adsorbent (3A zeolite with an adsorption capacity of 4 to 6 mol/kg within the earlier mentioned temperature range) to achieve the extremely high conversions sought without any additional subsequent and costly tail-gas processing that is otherwise required to curtail the emissions of residual hydrogen sulphide and sulphur dioxide.

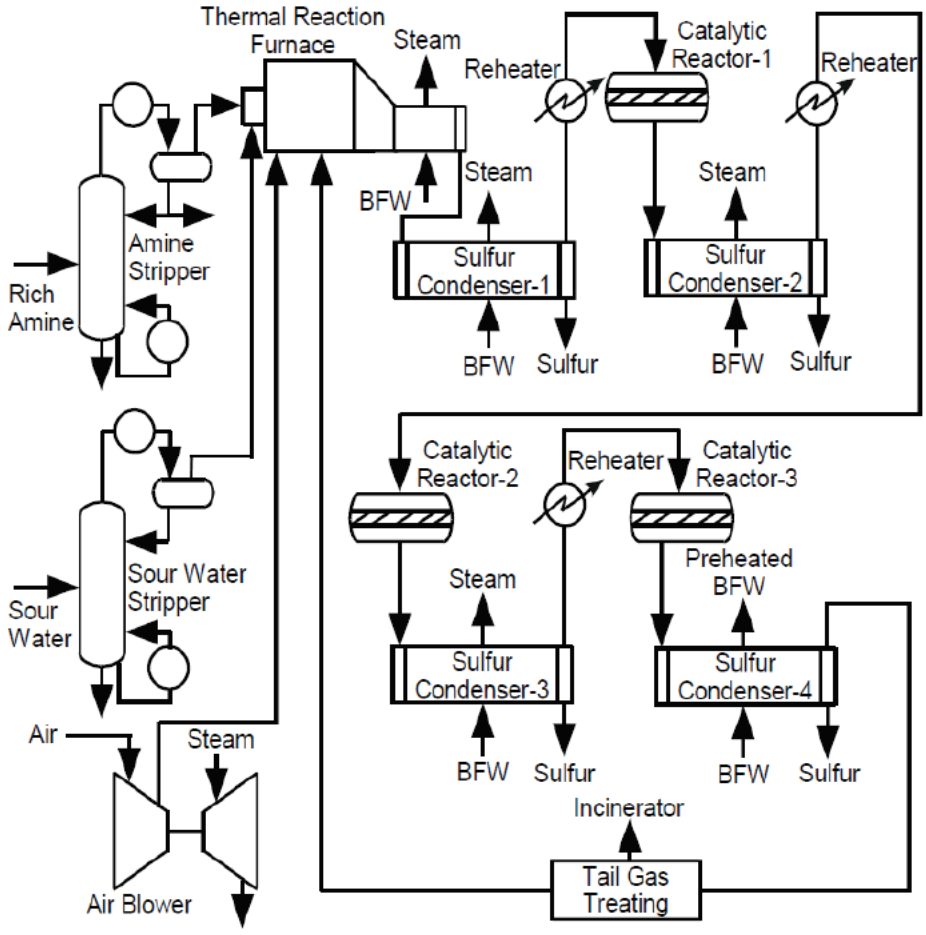


Figure 3: Process flowsheet of Claus process

Other existing technologies for sulphur recovery and tail-gas cleanup

Several improvements to the foresaid typical Claus process have been developed and commercialised, namely: direct oxidation, acid gas enrichment, oxygen enrichment, cold bed adsorption, and Shell Claus off-gas treatment (known also as SCOT process).

a) Direct oxidation process

It is special case of the split-flow Claus when the hydrogen sulphide content in the acid gas drops below 15%. In this case, the upstream acid gas is mixed with air to catalytically react hydrogen sulphide with oxygen and form sulphur dioxide rather than using a combustion burner in the process. The direct oxidation process is sensitive to catalyst poisons contained in the gas feed, mainly the hydrocarbons, and that is why is not widely used as much as other Claus technologies.

b) Acid gas enrichment

By contacting the acid gas containing low hydrogen sulphide amounts with a solvent designed to selectively absorb the entire hydrogen sulphide content while letting most of other gases slip through, the gas treating system can be enriched in hydrogen sulphide concentration by a factor of five or more enabling the reliable straight-through Claus rather than the direct oxidation to be used.

c) Oxygen enrichment

Usage of air to supply the oxygen required for the hydrogen sulphide combustion also introduces a large amount of inert nitrogen that lowers the adiabatic flame temperature of the reaction furnace. Additionally, this amount of nitrogen should be heated, cooled, and reheated through the process. Thus, using a pure oxygen or enriched sources of oxygen instead in the Claus process allows to establish higher flame temperatures with lower hydrogen sulphide concentration in the system and reduce the size of the relative equipment in proportion to the quantity of nitrogen avoided to be inserted to the process.

d) Tail-gas treatment technologies

Previously described Claus technologies achieve a sulphur recovery level of up to only 97% due to associated reaction equilibrium limitations, whereas the minimum legislatively required sulphur recovery is 99.5% which means still further processing is needed.

Tail-gas cleanup technologies can achieve the sulphur recoveries sought and can be integrated with the sulphur recovery unit. They are divided into two main categories: the dry bed processes such as the cold bed adsorption process (also known as sub-dew point Claus Process) developed and licensed by BP corporation, and the wet scrubbing processes like the Shell Claus off-gas treating (SCOT) process licensed by Shell Development and often employed in refineries to establish very high sulphur recovery levels (≥ 99.8%).

i. Cold bed adsorption (CBD) process

In the cold bed adsorption process, the same catalytic Claus reaction as the standard Claus process is operated, but at lower temperatures (120 – 150 °C) below the dew point of the produced elemental sulphur which deposits then on the catalyst bed as a liquid without inhibiting the reaction since it occurs in the gas phase. After adsorbing a certain amount of sulphur before blocking all active sites in the catalyst pores rendering the catalyst bed completely inactive, the catalyst is regenerated by stripping the sulphur from the catalyst by flowing a hot gas through the reactor and vaporise the deposited liquid sulphur to be then condensed again and removed downstream the process. Higher than 99% sulphur recovery levels can be attained by this process. Nevertheless, the CBD process is unfortunately associated with operational and maintenance problems since it involves gas switching valves in very demanding liquid sulphur environment.

ii. Shell Claus off-gas treating (SCOT) process

As stated before, very high sulphur recoveries (more than 99.8%) can be achieved by SCOT process, in which all the sulphur compounds contained in the tail-gas from standard Claus unit is converted back to hydrogen sulphide in front-end section and

then after cooling, the hydrogen sulphide is absorbed from the process gas by contacting it with a solvent (amine-based). The solvent is then regenerated, and the absorbed hydrogen sulphide is stripped out and recycled to the upstream Claus process for further conversion and recovery. A schematic illustration of the simplified SCOT process is shown in Figure 4.

The sulphur emission in the incinerator effluent can reach less than 250 ppm, however, the SCOT process does not only include high capital expenses (often 80% or more of the cost of the upstream Claus process), but also high operational expenses.

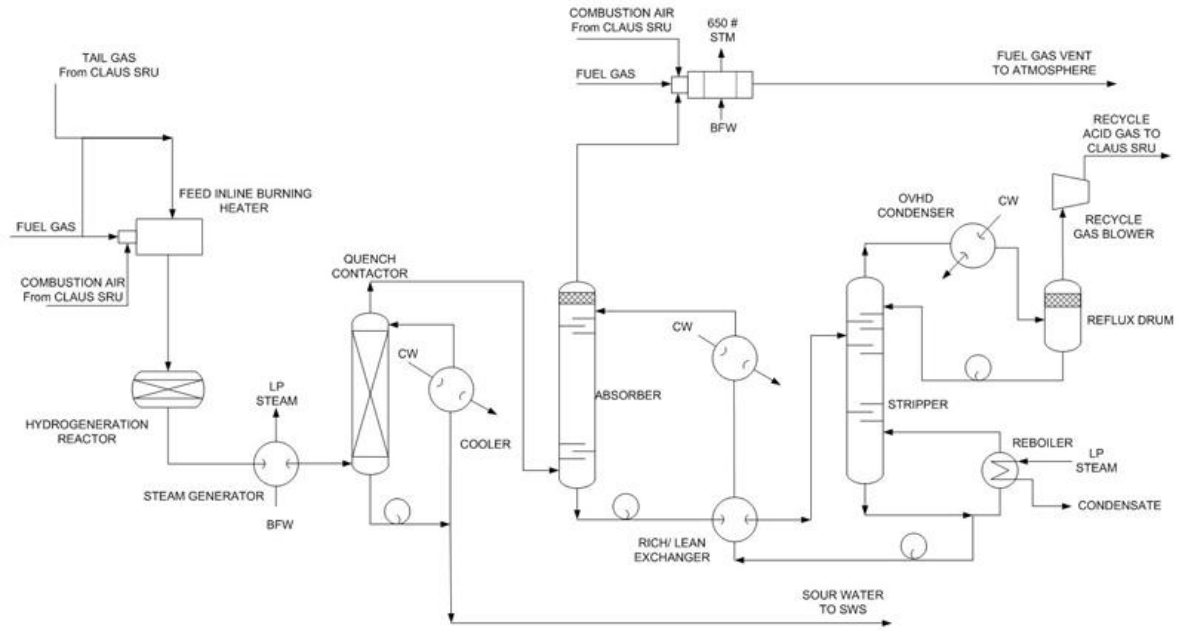


Figure 4: Simplified scheme of SCOT process

1.4.2. Deacon reaction

Chlorine chemistry is widely involved in the chemical industry and contributes indispensably to efficiently convert feedstocks to intermediates and/or end products. The inserted chlorine into the reacting system is released afterwards as chloride components, often as hydrogen chloride with a few exceptions such as vinyl chloride. To reconvert the released hydrogen chloride back into its origin, the chlorine,

electrolysis can be applied, however by using a considerable amount of electrical energy and thus it entails high investment and operating costs. Contrastingly, the power consumption of the heterogeneously catalysed chemical oxidation of hydrogen chloride, the Deacon reaction, is attractively low and the reaction itself is an exothermic process leading to a steam credit.

Despite several attempts, the industrial implementation of the Deacon process has been mainly come to grief on the partial conversions of around 85% with conventional reactors and catalysts, leading to complex and expensive processing of the resultant gas mixture. Other industrial Deacon processes developed with copper-based catalysts, like the Shell-Chlor process, have been abandoned not only because of the low achieved HCl conversions, but also due to the fast deactivation rate of the catalysts in form of volatilised chlorides of the active metal and overwhelming corrosive environment in the plant by the unreacted HCl and the existed water vapour.

Different flowsheets of the Deacon process have been proposed, two of which are presented here.

- **Air reduction Co. flowsheet**

The Deacon process flowsheet proposed by the firm Air reduction Co. is shown in Figure 5. It has been operated in a pilot-scale plant using copper chloride catalyst under temperatures between 450 – 600 °C. The unconverted hydrogen chloride separated in an absorption column is recycled back into the reaction unit and the gas stream is dried using concentrated sulphuric acid. Through a two-stage compression unit, the nitrogen and oxygen are separated in a purge stream.

- **Sumitomo flowsheet**

The process flowsheet developed by the company Sumitomo (Figure 6) was based on using a highly active and long-life ruthenium oxide catalyst. This process has been commercialised and higher yields at reasonable costs have been reported [94].

The recent development of innovative ruthenium-based catalysts [95] has decreased the reaction temperature to the point where adsorptive reactor operation is now

possible. Chronological development of catalysed chlorine-production indicating catalyst composition, reactor type, operating temperature range, and status can be found in [96]. The stability against bulk chlorination, high activity at low operating temperatures (180 – 380 °C), and the high thermal conductivity are the most important and remarkable features of the new ruthenium-based catalysts in comparison to the previously developed catalysts.

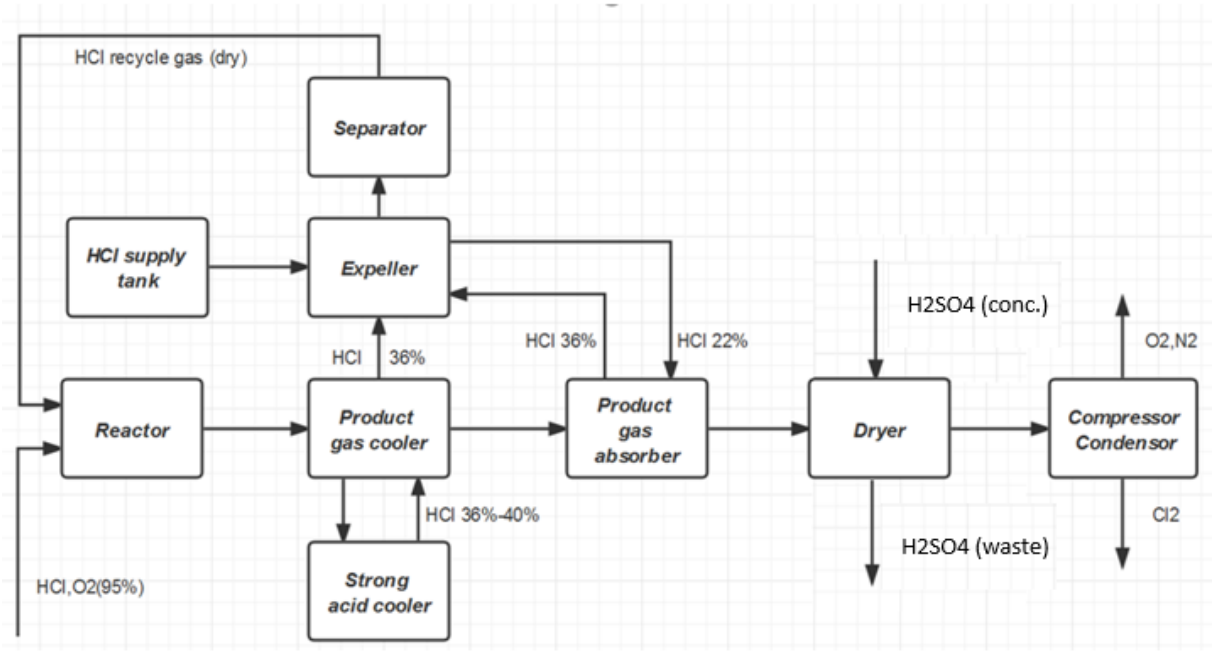


Figure 5: Deacon process proposed by Air Reduction Co.

The latter aspect, i.e. the high thermal conductivity, means practically that the formation of hot spots inside the catalyst bed is unlikely to occur and thus these catalysts are suitable for fixed bed reactor configurations. In this regard, two ruthenium-based catalysts should be mentioned as successfully commercialised examples: RuO_2/SnO_2-cassiterite developed by Bayer in 2006 and RuO_2/TiO_2-rutile licensed by Sumitomo in 2002. It is worthy to mention here, that Bayer Material Science (BMS), as being one of the largest producers of polyurethane and polycarbonates worldwide, has conducted several cooperative projects to further develop appropriate catalyst material to efficiently implement the highly relevant Deacon process to phosgene-mediated polyurethane and polycarbonates business [96].

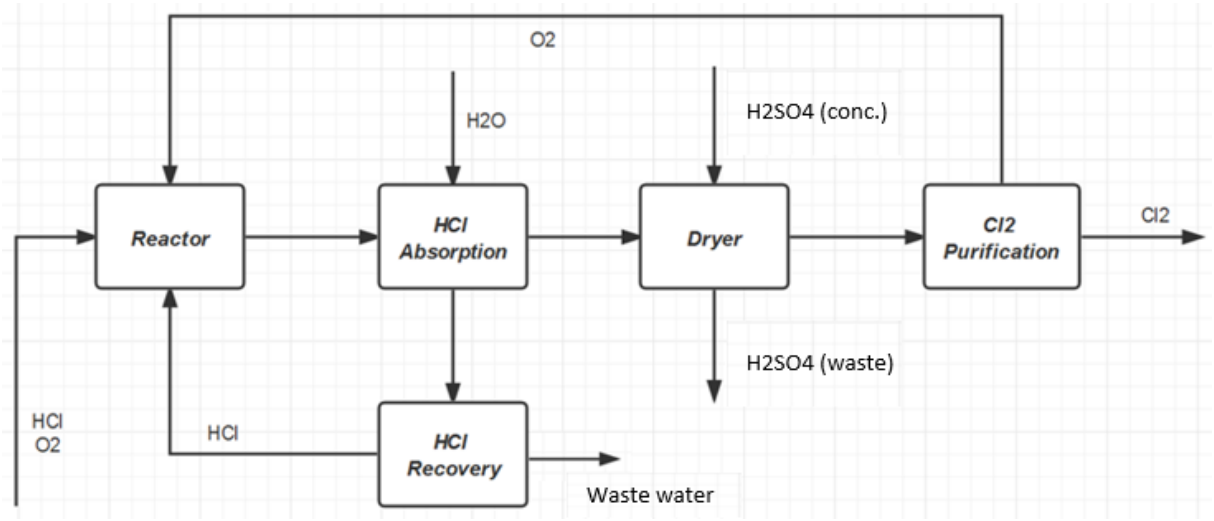


Figure 6: Deacon process proposed by Sumitumo

The chemical reaction equation of Deacon process can be written as follows:

$$4\mathrm{HCl} + \mathrm{O_2} \Leftrightarrow 2\mathrm{Cl_2} + 2\mathrm{H_2O} \qquad \Delta_R H = -57\ \mathrm{kJ/mol} \qquad (\mathrm{Eq.\ 2})$$

Provided adsorbents resistant to the very aggressive reaction medium can be found, quantitative hydrogen chloride conversion with “minimal” excess oxygen is thus a realistic option.

1.5. Objectives and outline

The objective of this thesis was to acquire an overall assessment of the technical conditions under which adsorptive reactors can optimally be operated and develop general guidelines and criteria for the application of an optimal degree of functionality integration in isothermal and adiabatic adsorptive reactor operation. Therefore, several comprehensive studies have been conducted on the abovementioned Claus- and Deacon-reactions, considering both segregated and integrated mixed fixed beds to increase the conversion of these two equilibrium-limited reactions up to a point at which further downstream processing becomes very simple or even superfluous. This was achieved by incorporating an adsorptive functionality in the catalytic fixed bed reactor and optimally utilising the degrees of freedom available in the reactor or process architecture to manipulate the concentration and temperature profiles in the reactor.

The foresaid studies were conducted in the light of the two following points that are essential for any process intensified technology:

- systematic derivation of intensified and feasible designs, and
- ensuring the operational optimality of the derived bifunctional designs.

Figure 7 depicts an overview of the studied cases of the two considered reactions with their modelling methodology used. As can be seen, multiscale (reactor and pellet level) and multidimensional (one-dimensional and two-dimensional) modelling of adsorptive fixed bed reactors has been developed. The main model used though is the one-dimensional pseudo-homogeneous one; this model represents the state of the art for modelling adsorptive fixed bed reactors [78, 97]. Although the dynamic models developed in this thesis were applied for the abovementioned two reactions, they are also applicable for other reaction schemes.

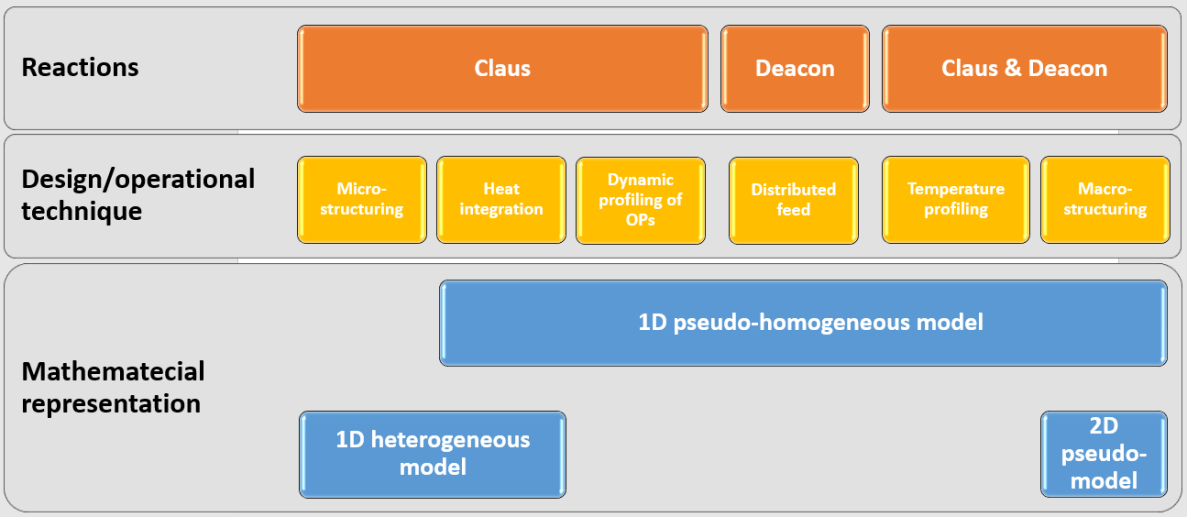


Figure 7: Methodology and scope of the thesis

In the following chapter (CHAPTER 2), the mathematical modelling of adsorptive reactors is introduced followed by highlighting the preliminary results on structuring adsorptive reactors in CHAPTER 3. The main part of this thesis is CHAPTER 4 which presents and discusses the obtained results of the studied design and operational techniques of adsorptive reactors. Each section in CHAPTER 4 was written based on

a separate publication and thus some overlap occurs. At the end of CHAPTER 4, two novel adsorptive reactor designs for further research investigations are proposed. Summary and conclusions with specific comments on the experimental verification and scale up aspects of the proposed adsorptive reactor designs developed in this work are given in CHAPTER 5.

CHAPTER 2: MATHEMATICAL MODELLING OF ADSORPTIVE REACTORS

Mathematical modelling and optimisation offer an economical tool for investigating a wide range of process alternatives and configurations. One of these is the fixed bed reactors, which are widely used in industry for heterogeneously catalysed fluid-solid reactions. Since adsorptive reactors comprise catalytic and adsorptive functionalities and thus involve gas-solid systems, they can be modelled as fixed bed reactors. In the light of the foreseen objectives of this work, multiscale and multidimensional models have been developed to describe the dynamic behaviour of isothermal as well as adiabatic fixed bed adsorptive reactors. Nevertheless, in the following sections, only the one-dimensional multiscale models will be introduced since the two-dimensional ones showed that the radial gradients of mass and heat can be neglected for the considered geometrical design of the studied adsorptive reactors (see also Section 4.3).

Both the developed one-dimensional pseudo-homogeneous and heterogeneous dynamic models encompass mainly the overall and component mass, energy, and momentum balances and are based on the following simplifying assumptions:

- backmixing is accounted for by using mass and heat axial dispersion terms
- the adsorption kinetics has been modelled considering the linear driving force approach (LDF)
- the ideal gas law applies under the considered conditions

- radial mass and heat transfer are negligible (verified by two-dimensional models)
- the adsorbent is considered to be:
    - highly selective to water
    - adsorbate-free at the start of each simulation
- the volume change during reaction is neglected due to the high dilution of both the reaction systems considered with nitrogen
- for isothermal adsorptive reactor simulations, it was assumed that there exists an infinite sink to dissipate the reaction and adsorption heats completely.

Especially for the heterogeneous model, the following additional assumptions apply:

- the pellets (catalyst, adsorbent, phase change material “PCM”) are assumed to be:
    - of spherical shape
    - uniform porous architecture
- heat conduction of the solids can be considered infinite
- the PCM is enclosed by a material with idealised infinite thermal conductivity.

The heat of reactions (given in Eq. 1 for Claus and in Eq. 2 for Deacon) are given for a temperature of 25°C and 1 bar Pressure, nevertheless they are valid for the operating temperatures applied in this study for both Claus and Deacon reactions since the change of the heat capacities of the considered gas mixtures can be neglected [23, 96]. Furthermore, all the parameters included in the model equations, which are discussed in the following sections are valid for the prevailing operating conditions chosen in this study for Claus and Deacon reactions (250 – 350°C and 1 bar).

Generally, the mass or energy balance of the studied system includes the following terms:

Accumulation = Dispersion + Convection + Transport term + Source/sink term

This general equation represents a parabolic dynamic differential equation, where the source/sink term is usually a non-linear expression with respect to the corresponding state (i.e. the concentration in case of a mass balance or the temperature in case of a heat balance equation).

The results presented in CHAPTER 3 and CHAPTER 4 were obtained by carrying out numerical simulations on an adsorptive reactor modelled and described in this chapter.

2.1. One-dimensional pseudo-homogeneous dispersion model

In the one-dimensional pseudo-homogeneous model, the concentration and temperature gradients between gas and solid surface can be neglected. This is justified by the fact that the overall reaction rate of the reaction schemes considered in this thesis is not limited by the external mass and thermal transport processes. This was emphasised when applying the one-dimensional heterogeneous model that will be presented in the following section.

Unlike the ideal plug flow reactor, the possible development of a residence time distribution inside the fixed bed reactor, due to local differences of the flow velocity, is taken into account by incorporating an effective axial mass and heat dispersion coefficients into the mass and heat balance equations, respectively.

The rigorous dynamic model of the abovementioned type derived in this work consists of overall and component mass balances, a heat balance in case of an adiabatic operation, and a momentum balance over the whole length of the multifunctional fixed bed. This set of algebraic and partial differential equations, which are described below, has been used for most of the studies conducted in this thesis.

The overall mass balance of the gas phase includes the reaction and adsorption source and sink terms as follows:

$$\frac{\partial C_{\mathrm{Tot}}}{\partial t} = -\frac{\partial (u C_{\mathrm{Tot}})}{\partial x} + \frac{\varphi_{cat}(1-\varepsilon)}{\varepsilon} \sum_i \nu_i \rho_{\mathrm{cat}} r_{\mathrm{R}} - \frac{(1-\varepsilon)}{\varepsilon} \varphi_{ads} \rho_{\mathrm{ads}} r_{\mathrm{A}} \qquad \textit{(Eq. 3)}$$

Based on the assumed highly selective adsorption process, the adsorption term only appears in the component mass balance for water vapour:

$$\frac{\partial C_i}{\partial t} = -\frac{\partial (u C_i)}{\partial x} + D_{\mathrm{ax}} \frac{\partial^2 C_i}{\partial x^2} + \frac{\varphi_{cat}(1-\varepsilon)}{\varepsilon} v_i \rho_{\mathrm{cat}} r_{\mathrm{R}} \; ; i \neq \mathrm{H_2O} \qquad \textit{(Eq. 4)}$$

$$\frac{\partial C_{\mathrm{H_2O}}}{\partial t} = -\frac{\partial \left(u\, C_{\mathrm{H_2O}}\right)}{\partial x} + D_{\mathrm{ax}} \frac{\partial^2 C_{\mathrm{H_2O}}}{\partial x^2} + \frac{\varphi_{cat}(1-\varepsilon)}{\varepsilon} v_{\mathrm{H_2O}} \rho_{\mathrm{cat}} r_{\mathrm{R}} - \frac{(1-\varepsilon)}{\varepsilon} \varphi_{ads} \rho_{\mathrm{ads}} r_{\mathrm{A}} \qquad \textit{(Eq. 5)}$$

In case of an adiabatic operation of the adsorptive reactor, the heat balance was written for the gas phase as well as for the solid phase, on which the reaction and adsorption take place:

$$\rho_{\mathrm{s}} C_{p,s} \frac{\partial T_{\mathrm{s}}}{\partial t} = \lambda_{\mathrm{s}} \frac{\partial^2 T_{\mathrm{s}}}{\partial x^2} + (-\Delta_{\mathrm{R}} H) \varphi_{cat} \rho_{\mathrm{cat}} r_{\mathrm{R}} + (-\Delta_{\mathrm{A}} H) \varphi_{ads} \rho_{\mathrm{ads}} r_{\mathrm{A}} - \alpha\beta\left(T_{\mathrm{s}} - T_{\mathrm{g}}\right) \qquad \textit{(Eq. 6)}$$

$$\rho_{\mathrm{g}} C_{p,\mathrm{g}} \frac{\partial T_{\mathrm{g}}}{\partial t} = -\rho_{\mathrm{g}} C_{p,\mathrm{g}} \frac{\partial (u T_{\mathrm{g}})}{\partial x} + \Lambda_{\mathrm{ax}} \frac{\partial^2 T_{\mathrm{g}}}{\partial x^2} + \frac{(1-\varepsilon)}{\varepsilon} \alpha\beta\left(T_{\mathrm{s}} - T_{\mathrm{g}}\right) \qquad \textit{(Eq. 7)}$$

The density (ρ_{s}), heat capacity ($C_{p,\mathrm{s}}$), and heat conductivity (λ_{s}) of the solid phase were calculated based on the volume fractions of reactive and adsorptive functionalities present in the simulated fixed bed.

The pressure drop along the packed bed was modelled as a momentum balance according to Ergun equation relating the pressure drop term to the velocity changes over the adsorptive fixed bed reactor as follows:

$$\frac{\partial P_{\mathrm{Tot}}}{\partial x} = -\frac{\rho_{\mathrm{g}} u^2 (1-\varepsilon)}{d_{\mathrm{p}} \varepsilon^3} \times \left(\frac{150(1-\varepsilon)\mu_{\mathrm{g}}}{d_{\mathrm{p}} \rho_{\mathrm{g}} u} + 1.75\right) \qquad \textit{(Eq. 8)}$$

2.1.1. Initial conditions

The initial conditions, at any *x* over the fixed bed length *L,* in this study are given by:

$$C_i|_{t=0,x} = 0\ ;\ \ i \neq \mathrm{N_2} \qquad \textit{(Eq. 9)}$$

$$C_{\mathrm{N_2}}\big|_{t=0,x} = C_{\mathrm{Tot}}|_{t=0,x} = C_{\mathrm{f}} \qquad \textit{(Eq. 10)}$$

$$T_{\mathrm{s}}|_{t=0,x} = T_{\mathrm{g}}\big|_{t=0,x} = T_{\mathrm{f}} \qquad \textit{(Eq. 11)}$$

It is worth noting that for cyclic steady state simulations, the initial conditions mentioned above represent the starting bed conditions before cycle one begins and not the initial states from cycle to cycle when calculating the cyclic steady state.

2.1.2. Boundary conditions

Since the operation of adsorptive reactors is periodic by nature, the boundary conditions, at the inlet where *x* = 0 and at the outlet where *x* = *L,* have been changed to account for the countercurrent regeneration step considered for the cyclic operation and are given by following equations for the two cycle steps adopted: the reaction-adsorption step and the desorption-cooling step. Closed-closed vessel boundary conditions have been used for the modelled fixed bed adsorptive reactors. At both ends of the reactor, plug flow with no dispersion takes place while between them dispersion and reaction occur (Eqs. 12 & 13, and Eqs. 18 & 19). At the reactor exit, the concentration and temperature are continuous, i.e. no gradient of component concentrations nor in temperatures (Eq. 17 and Eq. 23).

Reaction-adsorption step:

$$-D_{\mathrm{ax}}\frac{\partial C_i}{\partial x}\bigg|_{x=0,t} = u|_{x=0,t}\left(y_{i,\mathrm{f}}.\,C_{\mathrm{f}} - C_i|_{x=0,t}\right) \qquad \textit{(Eq. 12)}$$

$$-\Lambda_{\mathrm{ax}}\frac{\partial T_{\mathrm{g}}}{\partial x}\bigg|_{x=0,t} = \left(\rho_{\mathrm{g}} C_{p,\mathrm{g}} u\right)\big|_{x=0,t}\left(T_{\mathrm{f}} - T_{\mathrm{g}}\big|_{x=0,t}\right) \qquad \textit{(Eq. 13)}$$

$$T_s|_{x=0,t} = T_{\mathrm{f}} \qquad \textit{(Eq. 14)}$$

$$u|_{x=0,t} = u_{\mathrm{f}} \qquad \textit{(Eq. 15)}$$

$$P_{\mathrm{Tot}}|_{x=0,t} = P_{\mathrm{f}} \qquad \textit{(Eq. 16)}$$

$$\left.\frac{\partial C_i}{\partial x}\right|_{x=L,t} = \left.\frac{\partial T_{\mathrm{g}}}{\partial x}\right|_{x=L,t} = \left.\frac{\partial T_{\mathrm{s}}}{\partial x}\right|_{x=L,t} = 0 \qquad \textit{(Eq. 17)}$$

Desorption-cooling step:

$$-D_{\mathrm{ax}}\left.\frac{\partial C_i}{\partial x}\right|_{x=L,t} = u|_{x=L,t}\left(y_{i,reg}.C_{reg} - C_i|_{x=L,t}\right) \qquad \textit{(Eq. 18)}$$

$$-\Lambda_{\mathrm{ax}}\left.\frac{\partial T_{\mathrm{g}}}{\partial x}\right|_{x=L,t} = \left(\rho_{\mathrm{g}}C_{p,\mathrm{g}}u\right)\big|_{x=L,t}\left(T_{reg} - T_{\mathrm{g}}\big|_{x=L,t}\right) \qquad \textit{(Eq. 19)}$$

$$T_s|_{x=L,t} = T_{\mathrm{reg}} \qquad \textit{(Eq. 20)}$$

$$u|_{x=L,t} = -u_{reg} \qquad \textit{(Eq. 21)}$$

$$P_{\mathrm{Tot}}|_{x=L,t} = P_{reg} \qquad \textit{(Eq. 22)}$$

$$\left.\frac{\partial C_i}{\partial x}\right|_{x=0,t} = \left.\frac{\partial T_{\mathrm{g}}}{\partial x}\right|_{x=0,t} = \left.\frac{\partial T_{\mathrm{s}}}{\partial x}\right|_{x=0,t} = 0 \qquad \textit{(Eq. 23)}$$

The values used in the simulations in both feed and regeneration cycle steps for total feed concentration (C_{f} & C_{reg}), feed mole fraction of species *i* ($y_{i,\mathrm{f}}$ & $y_{i,\mathrm{reg}}$), feed temperature (T_{f} & T_{reg}), feed velocity (u_{f} & u_{reg}), and feed pressure (P_{f} & P_{reg}) of the gas phase are listed in Table 3 for the Claus and Deacon reactions.

2.2. One-dimensional heterogeneous dispersion model

To conduct simulation studies at the pellet scale, especially for microstructuring and micro-heat-integration using PCM, a one-dimensional heterogeneous model has been developed to account for possible mass and heat transport limitations between the gas phase and the separate functionality solids (catalyst, adsorbent, and PCMs).

The created model consists of an overall and component mass, heat and momentum balances of the gas phase. Additionally, the effective pore diffusion model was used to describe the mass and/or heat transfer in the different solid phases. The resulting multi-scale dynamic model equations with the corresponding initial and boundary conditions are given below.

2.2.1. Reactor model

The reactor model is a fixed bed reactor and one-dimensional axial dispersion, which incorporate the pellet models given in the following subsections. The general mass balance of the gas phase in the reactor is written as follows:

$$\frac{\partial C_{Tot}}{\partial t} = -\frac{\partial (u C_{Tot})}{\partial x} - \sum_{i=1}^{N_{comp}} k_{i,film} \times \frac{3}{r_p} \times \frac{(1-\varepsilon)}{\varepsilon} \times \varphi_{cat} \times \left(C_{i,g} - C_{i,cat}\big|_{r=r_p} \right) - \sum_{i=1}^{N_{comp}} k_{i,film} \times \frac{3}{r_p} \times \frac{(1-\varepsilon)}{\varepsilon} \times \varphi_{ads} \times \left(C_{i,g} - C_{i,ads}\big|_{r=r_p} \right) \qquad \textit{(Eq. 24)}$$

The second and third terms on the right-hand side in Eq. 24 accounts for the mass transfer between the gas bulk phase and the catalyst and adsorbent pellets, respectively. Both terms are weighted by the volume fraction factor of the reactive and adsorptive functionality within the reactor.

Analogous to the overall mass balance, the component mass balance contains the mass transfer terms, where the adsorptive mass transfer from the gas phase to the adsorbent pellets occurs only for the water vapour as in Eq. 26.

$$\frac{\partial C_{i,g}}{\partial t} = -\frac{\partial\left(uC_{i,g}\right)}{\partial x} + D_{\mathrm{ax}}\frac{\partial^2 C_{i,g}}{\partial x^2} - \sum_{i=1}^{N_{comp}} k_{i,film} \times \frac{3}{r_p} \times \frac{(1-\varepsilon)}{\varepsilon} \times \varphi_{cat} \times \left(C_{i,g} - C_{i,cat}\big|_{r=r_p}\right) \qquad \textit{(Eq. 25)}$$

$where: i \neq \mathrm{H_2O}$

$$\frac{\partial C_{\mathrm{H_2O,g}}}{\partial t} = -\frac{\partial\left(uC_{\mathrm{H_2O,g}}\right)}{\partial x} + D_{\mathrm{ax}}\frac{\partial^2 C_{\mathrm{H_2O,g}}}{\partial x^2} - \sum_{i=1}^{N_{comp}} k_{i,film} \times \frac{3}{r_p} \times \frac{(1-\varepsilon)}{\varepsilon} \times \varphi_{cat} \times \left(C_{\mathrm{H_2O,g}} - C_{i,cat}\big|_{r=r_p}\right) - \sum_{i=1}^{N_{comp}} k_{i,film} \times \frac{3}{r_p} \times \frac{(1-\varepsilon)}{\varepsilon} \times \varphi_{ads} \times \left(C_{\mathrm{H_2O,g}} - C_{i,ads}\big|_{r=r_p}\right) \qquad \textit{(Eq. 26)}$$

The general heat balance of the gas bulk phase does not include heat generation terms, but it contains the dynamic heat transfer terms between the gas phase and the existing functional pellets weighted by their volumetric contribution in the reactor. Mathematically, this balance is defined as written in Eq. 27:

$$\rho_{\mathrm{g}} C_{p,\mathrm{g}}\frac{\partial T_{\mathrm{g}}}{\partial t} = -\rho_{\mathrm{g}} C_{p,\mathrm{g}}\frac{\partial(uT_{\mathrm{g}})}{\partial x} + \Lambda_{\mathrm{ax}}\frac{\partial^2 T_{\mathrm{g}}}{\partial x^2} - \alpha_{film} \times \frac{3}{r_p} \times \frac{(1-\varepsilon)}{\varepsilon} \times \varphi_{cat} \times \left(T_g - T_{cat}|_{r=r_p}\right) - \alpha_{film} \times \frac{3}{r_p} \times \frac{(1-\varepsilon)}{\varepsilon} \times \varphi_{ads} \times \left(T_g - T_{ads}|_{r=r_p}\right) - \alpha_{film} \times \frac{3}{r_p} \times \frac{(1-\varepsilon)}{\varepsilon} \times \varphi_{PCM} \times \left(T_g - T_{PCM}|_{r=r_p}\right) \qquad \textit{(Eq. 27)}$$

A momentum balance in form of Ergun equation has been used to calculate the axial pressure drop inside the reactor as follows:

$$\frac{\partial P_{\mathrm{Tot}}}{\partial x} = -\frac{\rho_{\mathrm{g}} u^2 (1-\varepsilon)}{d_{\mathrm{p}}\varepsilon^3}\left(\frac{150(1-\varepsilon)\mu_{\mathrm{g}}}{d_{\mathrm{p}}\rho_{\mathrm{g}} u} + 1.75\right) \qquad \textit{(Eq. 28)}$$

Table 3: Feed composition and conditions used in the simulations

Variable	Value	
	Claus	Deacon
C_f & C_{reg} [kmol.m^{-3}]	0.023	0.023
$y_{N2,\,f}$, [-]	0.85	0.8
$y_{N2,\,reg}$, [-]*	1.0	1.0
$y_{H2S,\,f}$, [-]	0.1	-
$y_{SO2,\,f}$, [-]	0.05	-
$y_{HCl,\,f}$, [-]	-	0.1
$y_{O2,\,f}$, [-]	-	0.1
T_f, [K]	523	573
T_{reg}, [K]	523	573
u_f, [m.s^{-1}]	0.2	0.2
u_{reg}, [m.s^{-1}]	0.4	0.4
P_f & P_{reg}, [Pa]	10^5	10^5
**: the mole fractions of other components were set to zero during the regeneration step*		

Initial conditions:

The simulations were started under the initial conditions that the reactor contains only an inert, nitrogen, at the reference gas temperature (T_f):

$$C_i|_{t=0,x} = 0\ ;\ \ i \neq \mathrm{N_2} \qquad \textit{(Eq. 29)}$$

$$C_{\mathrm{N_2}}\big|_{t=0,x} = C_{\mathrm{Tot}}|_{t=0,x} = C_{\mathrm{f}} \qquad \textit{(Eq. 30)}$$

$$T_{\mathrm{g}}\big|_{t=0,x} = T_{\mathrm{f}} \qquad \textit{(Eq. 31)}$$

Boundary conditions:

Similar to the one-dimensional pseudo-homogeneous reactor model presented in the section above, the closed vessel boundary conditions, known as Danckwerts boundary conditions, are chosen here for the reactor-level model. According to which of the two-step cyclic operation of the adsorptive reactor is being simulated, the entrance and exit of the reactor are switched over, where no gradient of the concentration or the gas temperature at the exit is defined as boundary condition (Eq. 36 and Eq. 41).

Reaction-adsorption step:

$$-D_{\mathrm{ax}}\frac{\partial C_i}{\partial x}\bigg|_{x=0,t} = u|_{x=0,t}\left(y_{i,\mathrm{f}}.\,C_{\mathrm{f}} - C_i|_{x=0,t}\right) \qquad \textit{(Eq. 32)}$$

$$-\Lambda_{\mathrm{ax}}\frac{\partial T_{\mathrm{g}}}{\partial x}\bigg|_{x=0,t} = \left(\rho_{\mathrm{g}}C_{p,\mathrm{g}}u\right)\big|_{x=0,t}\left(T_{\mathrm{f}} - T_{\mathrm{g}}\big|_{x=0,t}\right) \qquad \textit{(Eq. 33)}$$

$$u|_{x=0,t} = u_{\mathrm{f}} \qquad \textit{(Eq. 34)}$$

$$P_{\mathrm{Tot}}|_{x=0,t} = P_{\mathrm{f}} \qquad \textit{(Eq. 35)}$$

$$\frac{\partial C_i}{\partial x}\bigg|_{x=L,t} = \frac{\partial T_{\mathrm{g}}}{\partial x}\bigg|_{x=L,t} = 0 \qquad \textit{(Eq. 36)}$$

Desorption-cooling step:

$$-D_{\mathrm{ax}}\frac{\partial C_i}{\partial x}\bigg|_{x=L,t} = u|_{x=L,t}\left(y_{i,reg}.\,C_{\mathrm{reg}} - C_i|_{x=L,t}\right) \qquad \textit{(Eq. 37)}$$

$$-\Lambda_{\mathrm{ax}}\frac{\partial T_{\mathrm{g}}}{\partial x}\bigg|_{x=L,t} = \left(\rho_{\mathrm{g}}C_{p,\mathrm{g}}u\right)\big|_{x=L,t}\left(T_{reg} - T_{\mathrm{g}}\big|_{x=L,t}\right) \qquad \textit{(Eq. 38)}$$

$$u|_{x=L,t} = -u_{reg} \qquad \textit{(Eq. 39)}$$

$$P_{\mathrm{Tot}}|_{x=L,t} = P_{\mathrm{reg}} \qquad \textit{(Eq. 40)}$$

$$\left.\frac{\partial C_i}{\partial x}\right|_{x=0,t} = \left.\frac{\partial T_{\mathrm{g}}}{\partial x}\right|_{x=0,t} = 0 \qquad \textit{(Eq. 41)}$$

The reference values of feed conditions listed in Table 3 apply also for the equations Eq. 30 to Eq. 40.

2.2.2. Catalyst model

The dynamic evolution of the concentration of component *i* inside the catalyst pellets ($C_{i,cat}$) is described by an effective mass pore diffusion ($D_{i,cat}^{eff}$) model including the mass consumption/production term as follows:

$$\frac{\partial C_{i,cat}}{\partial t} = D_{i,cat}^{eff}\left(\frac{\partial^2 C_{i,cat}}{\partial r^2} + \frac{2}{r}\frac{\partial C_{i,cat}}{\partial r}\right) + \frac{(1-\varepsilon_{cat})}{\varepsilon_{cat}} \times \nu_i \rho_{\mathrm{cat}} r_{\mathrm{R}}$$

where: *(Eq. 42)*

$$D_{i,cat}^{eff} = \frac{\varepsilon_{cat}}{\tau_{cat}} \times d_m$$

The heat transfer inside the catalyst pellets is modelled by using the effective heat pore diffusion coefficient (λ_{cat}^{eff}) considering in parallel the heat generated due to the chemical reaction on the active sites of the catalyst particles:

$$\rho_{cat} c_{p,cat} \frac{\partial T_{cat}}{\partial t} = \lambda_{cat}^{eff} \times \left(\frac{\partial^2 T_{cat}}{\partial r^2} + \frac{2}{r}\frac{\partial T_{cat}}{\partial r}\right) - \rho_{\mathrm{cat}} r_{\mathrm{R}}(-\Delta_{\mathrm{R}} H) \qquad \textit{(Eq. 43)}$$

Initial conditions:

Initially, the component concentrations and temperature inside the catalyst are identical to those of the bulk gas phase at the beginning of the simulation:

$$C_{i,cat}|_{t=0,r} = C_{i,g}\,|_{t=0} \qquad \textit{(Eq. 44)}$$

$$T_{cat}|_{t=0,r} = T_f \qquad \textit{(Eq. 45)}$$

Boundary conditions:

Assuming a linear concentration and temperature profiles at the surface of the reactive particles and symmetrical profiles in the centre of the pellets, the following boundary conditions can be obtained:

$$D_{i,cat}^{eff} \left.\frac{\partial C_{i,cat}}{\partial r}\right|_{t,r=r_p} = k_{i,film} \times \left(C_{i,g} - C_{i,cat}|_{t,r=r_p}\right) \quad \textit{(Eq. 46)}$$

$$\left.\frac{\partial C_{i,cat}}{\partial r}\right|_{t,r=0} = 0 \quad \textit{(Eq. 47)}$$

$$\lambda_{cat}^{eff} \left.\frac{\partial T_{cat}}{\partial r}\right|_{t,r=r_p} = \alpha_{film} \times \left(T_g - T_{cat}|_{t,r=r_p}\right) \quad \textit{(Eq. 48)}$$

$$\left.\frac{\partial T_{cat}}{\partial r}\right|_{t,r=0} = 0 \quad \textit{(Eq. 49)}$$

2.2.3. Adsorbent model

Adsorption processes on the adsorbent pellets exhibit similarities to the reaction on the catalytic particles regarding the mass and heat transfer resistances. Therefore, the mass balance equation for the water vapour in the adsorbent ($C_{\mathrm{H_2O,ads}}$) is written based on the effective mass pore diffusion coefficient ($D_{i,ads}^{eff}$) and the hold-up rate on the active sites of the adsorbent as in Eq. 50. As mentioned earlier, other components were assumed not to be adsorbed on the adsorbent pellets.

$$\frac{\partial C_{\mathrm{H_2O,ads}}}{\partial t} = D_{i,ads}^{eff} \left(\frac{\partial^2 C_{\mathrm{H_2O,ads}}}{\partial r^2} + \frac{2}{r}\frac{\partial C_{\mathrm{H_2O,ads}}}{\partial r}\right) - \frac{(1-\varepsilon_{ads})}{\varepsilon_{ads}} \rho_{\mathrm{ads}} r_{\mathrm{A}} \quad \textit{(Eq. 50)}$$

where:

$$D_{i,ads}^{eff} = \frac{\varepsilon_{ads}}{\tau_{ads}} \times d_m$$

Analogously, using the effective heat pore diffusion coefficient (λ_{ads}^{eff}) and the heat release due to the adsorption process, the heat balance of the adsorbent pellets can be written as follows:

$$\rho_{ads} c_{p,ads} \frac{\partial T_{ads}}{\partial t} = \lambda_{ads}^{eff} \left(\frac{\partial^2 T_{ads}}{\partial r^2} + \frac{2}{r} \frac{\partial T_{ads}}{\partial r} \right) - \rho_{\mathrm{ads}} r_{\mathrm{A}} (-\Delta_{\mathrm{A}} H) \qquad \textit{(Eq. 51)}$$

Initial conditions:

At the start point of the simulations, the concentration and temperature in the adsorbent particles are kept as initially defined in the bulk gas phase in the reactor:

$$C_{\mathrm{H_2O,ads}}\big|_{t=0,r} = C_{\mathrm{H_2O,g}}\big|_{t=0} \qquad \textit{(Eq. 52)}$$

$$T_{ads}|_{t=0,r} = T_f \qquad \textit{(Eq. 53)}$$

Boundary conditions:

Like the boundary conditions defined for the catalyst pellets, a linear concentration and temperature profiles are derived here at the outer layer of the adsorbent particles with symmetry conditions in the pellets' centre.

$$D_{i,ads}^{eff} \frac{\partial C_{\mathrm{H_2O,ads}}}{\partial r}\bigg|_{t,r=r_p} = k_{i,film} \times \left(C_{\mathrm{H_2O,g}} - C_{\mathrm{H_2O,ads}}\big|_{t,r=r_p} \right) \qquad \textit{(Eq. 54)}$$

$$\frac{\partial C_{\mathrm{H_2O,ads}}}{\partial r}\bigg|_{t,r=0} = 0 \qquad \textit{(Eq. 55)}$$

$$\lambda_{ads}^{eff} \frac{\partial T_{ads}}{\partial r}\bigg|_{t,r=r_p} = \alpha_{film} \times \left(T_g - T_{ads}|_{t,r=r_p} \right) \qquad \textit{(Eq. 56)}$$

$$\frac{\partial T_{ads}}{\partial r}\bigg|_{t,r=0} = 0 \qquad \textit{(Eq. 57)}$$

2.2.4. PCM model

In the case of conducting micro-heat-integration simulations, the pellet model of phase change material is used. For the used PCM in this study, the dimensionless heat Biot number ($\frac{\alpha_{film} \cdot \frac{3}{Rp}}{\lambda}$) was determined to be less than 0.1, meaning that the internal resistance of the heat transfer can be neglected in comparison to the dominating

external heat film diffusion between the PCM particles and the bulk gas phase in the reactor. Thus, a uniform temperature profiles exist inside the PCM particles for the solid and liquid phases.

Since neither external nor internal mass transfer occurs in the PCM pellets, only heat balance equation is derived according to the model proposed by Horstmeier et al. [98].

There are three different phases that the PCM pellets undergo during the operation: first the interior of the PCMs is still completely of a solid phase, afterwards the heat starts to be effectively transferred to the PCMs and the material inside changes its phase, and then the further heat transferred to the interior material of the PCMs lets it completely be liquified. These three stages are reversed as the heat is being restored by the gas bulk phase later during the regeneration process step. Figure 8 illustrates schematically these three stages during the cyclic operation.

1. *Solid PCMs ($T_{PCM} < T_{melting}$, $\omega = 0$); where ω is the liquid phase fraction of the PCMs particles*

In this stage, the dynamic change of the PCMs' temperature is subjected to the external heat transfer through the film layer between the gas phase and the PCMs surface as described below:

$$\frac{\partial}{\partial t}\left(\rho_{PCM,sol} \times c_{p,PCM,sol} \times T_{PCM}\right)\Big|_{x=x} = \alpha_{film}\big|_{x=x} \times \frac{3}{r_p} \times \left(T_g\big|_{x=x} - T_{PCM}|_{x=x}\right) \qquad \textit{(Eq. 58)}$$

2. *Melting/solidification phase of the PCMs ($T_{PCM} = T_{melting}$, $0 < \omega < 1$)*

As the PCMs starts to liquify/solidify, the heat of fusion should be considered in the heat balance equation as follows:

$$\frac{\partial}{\partial t}\left(\rho_{PCM,m} \times c_{p,PCM,m} \times T_{PCM}\right)\Big|_{x=x} = \alpha_{film}\big|_{x=x} \times \frac{3}{r_p} \times \left(T_g\big|_{x=x} - T_{PCM}|_{x=x}\right) - \rho_m \times \Delta_{fusion}H \times \frac{\partial \omega}{\partial t}\Big|_{x=x} \qquad \textit{(Eq. 59)}$$

3. *Liquid PCMs ($T_{PCM} > T_{melting}$, $\omega = 1$)*

For a completely liquified PCMs, the heat balance equation can be written as in stage one (Eq. 58) except for the physical properties of the material to be used, where the density and the specific heat capacity of the liquid phase instead of the solid phase are considered:

$$\frac{\partial}{\partial t}\left(\rho_{PCM,liq} \times c_{p,PCM,liq} \times T_{PCM}\right)\Big|_{x=x} = \alpha_{film}|_{x=x} \times \frac{3}{r_p} \times \left(T_g|_{x=x} - T_{PCM}|_{x=x}\right) \quad \textit{(Eq. 60)}$$

The liquified fraction of PCM particles during the cyclic operation is dynamically described by the following equation:

$$\rho_{PCM,m} \times \Delta_{fusion}H \times \frac{\partial \omega}{\partial t}\Big|_{x=x} = \alpha_{film}|_{x=x} \times \frac{3}{r_p} \times \left(T_g|_{x=x} - T_{PCM}|_{x=x}\right) \quad \textit{(Eq. 61)}$$

The mean density ($\rho_{PCM,m}$) and mean specific heat capacity ($c_{p,PCM,m}$) of the PCMs during the melting/solidification stage used in Eq. 59 are calculated according to the temporally progressed liquid-fraction (ω) of the PCMs and the liquid and solid corresponding physical properties as follows:

$$\rho_{PCM,m} = \rho_{PCM,liq} \times \omega + \rho_{PCM,sol} \times (1 - \omega) \quad \textit{(Eq. 62)}$$

$$c_{p,PCM,m} = c_{p,PCM,liq} \times \omega + c_{p,PCM,sol} \times (1 - \omega) \quad \textit{(Eq. 63)}$$

Initial conditions:

A completely solid phase interior of the PCMs and a temperature equal to the one prevailed in the gas bulk phase are derived as initial conditions at the outset of each simulation with PCM:

$$T_{PCM}|_{t=0} = T_f \quad \textit{(Eq. 64)}$$

$$\omega|_{t=0} = 0 \quad \textit{(Eq. 65)}$$

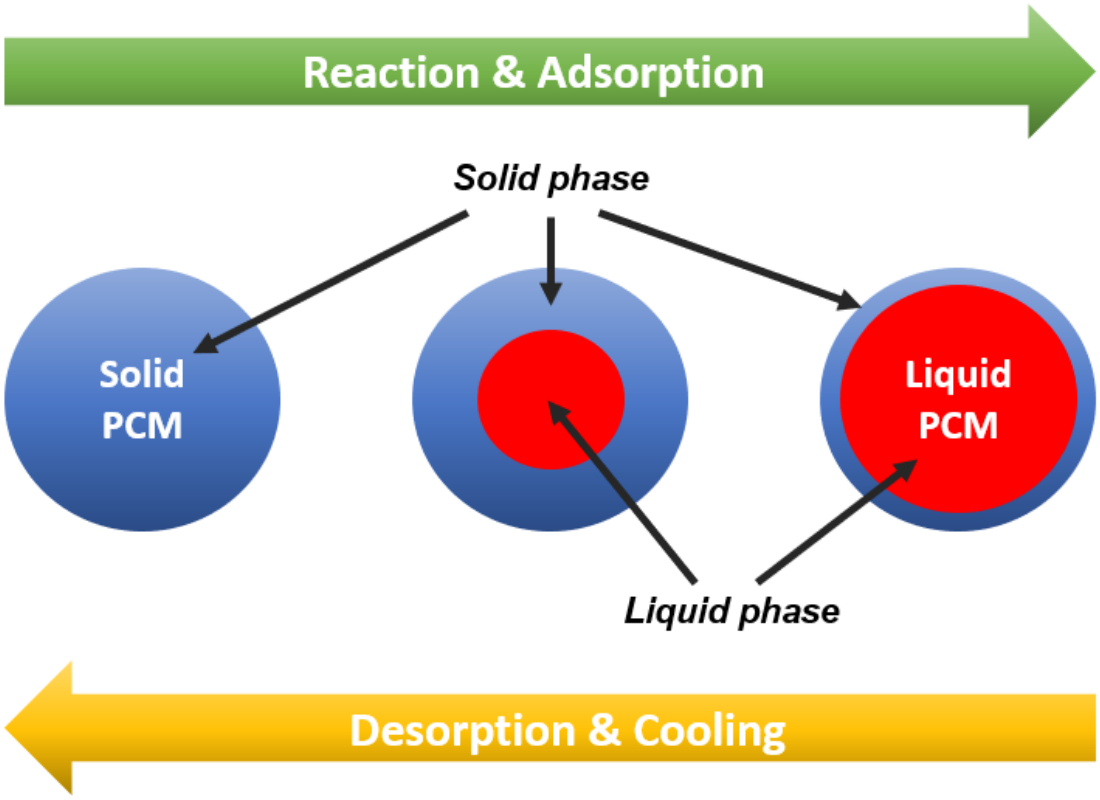


Figure 8: PCM stages during the cyclic operation of adsorptive reactors; Left: stage 1 (solid phase), Middle: stage 2 (Melting/solidification phase), and Right: stage 3 (liquid phase)

2.3. Mathematical expressions for reaction, adsorption, and transport properties

In this section, the mathematical correlations applied for the reaction rates on the catalysts and their physical properties, adsorption isotherm and adsorption rate with the physical properties of the studied adsorbent, and the transport properties of the mass/heat axial dispersion and film transfer coefficients, which are used in the system equations derived in the Sections 2.1. and 2.2., are introduced.

2.3.1. Reaction rate expressions

For the studies on Claus reaction, the gamma-Alumina catalyst has been chosen for the reactive functionality in the simulated adsorptive reactor. Based on the experimental determination of the reaction kinetics over the γ-Al_2O_3 done by Elsner [21, 71], the following expression described by a simple power law equation is adopted in this study:

$$r_{\mathrm{R}} = k_{1,\mathrm{Claus}} \times P_{\mathrm{H_2S}}^{0.95} \times P_{\mathrm{SO_2}}^{0.22} - k_{2,\mathrm{Claus}} \times P_{\mathrm{H_2O}} \qquad \textit{(Eq. 66)}$$

In case of Deacon reaction, Armute [96] and Teschner et al. [100] conducted kinetic experiments on Ruthenium-based catalyst ($RuO_2/SnO_2–Al_2O_3$) at different temperatures. The surface of the catalyst carrier $SnO_2–Al_2O_3$ was impregnated with ruthenium oxide in which its content in the catalyst was kept less than 3%.

According to the foresaid experimental investigation, a Langmuir-Hinshelwood-Hougen-Watson (LHHW) kinetics model for the reaction rate expression (Eq. 67) has been used for the Deacon adsorptive reactor simulations. The inhibition constants in the denominator of Eq. 67 ($K_{\mathrm{Cl_2}}$, $K_{\mathrm{H_2O}}$) can be lumped into one constant since that both chlorine and water vapour are produced in equimolar amounts by the reaction.

$$r_{\mathrm{R}} = \frac{k_{1,\mathrm{Deacon}} \times P_{\mathrm{O_2}} \times P_{\mathrm{HCl}}^{0.5} - \dfrac{k_{1,\mathrm{Deacon}} \times P_{\mathrm{Cl_2}} \times P_{\mathrm{H_2O}}}{\sqrt{\kappa \times P_{\mathrm{O_2}}^{-0.5} \times P_{\mathrm{HCl}}^{1.5}}}}{1 + K_{\mathrm{Cl_2}}\, P_{\mathrm{Cl_2}} + K_{\mathrm{H_2O}}\, P_{\mathrm{H_2O}}} \qquad \textit{(Eq. 67)}$$

The physical properties of the catalysts (γ-Al_2O_3 for Claus reaction and $RuO_2/SnO_2–Al_2O_3$ for Deacon reaction) used in the simulations are listed in Table 4.

Table 4: Physical properties of the considered catalysts

Property	Value		Units	Reference	
	Claus	Deacon		Claus	Deacon
ρ_{cat}	3600	6950	kg/m^3	[100]	[95, 96, 99]
$c_{p,cat}$	1050	4954.89	J/(kg·K)	[23]	[95, 96, 99]
ε_{cat}	0.53	0.53	-	[23]	
τ_{cat}	5	5	-	[101, 102]	
λ_{cat}^{eff}	5.74	8.35	W/(m·K)	[23]	[95, 96, 99]
$\Delta_R H$	-108	-57	kJ/mol	[23]	[96]

2.3.2. Adsorption kinetics and adsorption isotherm

As stated earlier in this chapter, the linear driving force approach, in which the resistances to mass transfer are lumped into one overall mas transfer coefficient, is used to describe the dynamic adsorption rate as follows:

$$r_{\mathrm{A}} = k_{LDF}(q_{\mathrm{eq}} - q) \qquad \textit{(Eq. 68)}$$

The linear driving force constant (k_{LDF}) in Eq. 68 comprises generally the film mass transfer resistance, macropore and micropore diffusion resistances. However, according to the work of Simo et al. [103], the film and micropore diffusion resistances under the prevailing temperature range, for which Claus and Deacon adsorptive reactors are operated, can be neglected. Thus, the molecular diffusion and Knudsen diffusion are considered to account for the macropore diffusion resistance during the adsorption process; both lumped into one overall macropore diffusion coefficient (d_0), which in turn is related to the linear driving force coefficient as follows [104]:

$$k_{LDF} = \frac{15}{r_c^2}\frac{d_0}{1 + \dfrac{\partial q_{eq}}{\varepsilon_{ads}\partial C_{H_2O,eq}}} \qquad \textit{(Eq. 69)}$$

where:

$$\frac{1}{d_0} = \tau_{ads}\left(\frac{1}{d_m} + \frac{1}{d_K}\right)$$

The molecular diffusivity (d_m) is estimated during the dynamic simulations using the Aspen Properties software, while the Knudsen diffusion coefficient (d_K) is calculated according to Kauzmann correlation [105]:

$$d_K = 9700 \times r_c \sqrt{\frac{T_s}{Mw_{H_2O}}} \qquad \textit{(Eq. 70)}$$

where: the Mw_{H_2O} is the molecular weight of water.

In case of the 1-D heterogeneous model, the micropore diffusivity (D_μ) has also been considered in the equation of the linear driving force constant as given by Yang [105]:

$$D_\mu = \frac{D_{\mu,\infty} \times exp\left(\frac{-E_{A,diff}}{R \cdot T_{ads}}\right)}{1 - \frac{\overline{q}}{q_{eq}}} \qquad \textit{(Eq. 71)}$$

The equilibrium loading (q_{eq}) in Eq. 68, has been computed using Langmuir adsorption isotherm on 3A-Zeolite pellets as follows [103]:

$$q_{\mathrm{eq}} = q_{\mathrm{sat}} \frac{b.P_{\mathrm{H_2O}}}{1 + b.P_{\mathrm{H_2O}}} \qquad \textit{(Eq. 72)}$$

The saturation loading (q_{sat}: the maximum possible loading at complete coverage) is given by Eq. 73 [103], and the adsorption affinity (b) is written as a temperature-dependent function as in Eq.74 [103].

$$q_{sat} = q_{ref} \exp\left(\delta\left(1 - \frac{T_s}{T_{ref}}\right)\right) \qquad \textit{(Eq. 73)}$$

$$b = \frac{b_\infty}{\sqrt{T_s}} \exp\left(\gamma \frac{T_{ref}}{T_s}\right) \qquad \textit{(Eq. 74)}$$

The physical properties of the adsorbent (3A Zeolite) and the values of the parameters in Eqs. 73 & 74 used in the simulations are summarised in Table 5.

Table 5: Adsorbent phase properties and values of the parameters in the Eqs. 73 & 74

Property	Value	Units	Reference
ρ_{ads}	1199.3	kg/m^3	[103]
$c_{p,ads}$	1045	J/(kg·K)	[103]
ε_{ads}	0.37	-	[103]
τ_{ads}	2	-	[103]
λ_{ads}^{eff}	0.12	W/(m·K)	[103]
$r_{ads,macro}$	3	µm	[106]
r_c	1	µm	[107]
$\Delta_{ad}H$	-57.95	kJ/mol	[103]
q_{ref}	0.01075	kmol/kg	[103]
δ	0.68792	-	[103]
b_∞	5.3×10^{-10}	$K^{0.5}Pa^{-1}$	[103]
γ	23.235	-	[103]
T_{ref}	300	K	[103]

2.3.3. Transport properties of the adsorptive reactor system

The axial mass dispersion coefficient (D_{ax}) and film mass transfer coefficient ($k_{i,film}$) are taken as proposed by Wakao and Funazkri [108, 109] for packed beds

$$D_{\mathrm{ax}} = \frac{d_{\mathrm{m}}}{\varepsilon} \times (20 + 0.5 \times Sc \times Re) \qquad \textit{(Eq. 75)}$$

$$k_{i,film} = \frac{d_{\mathrm{m}}}{2r_p} \times \left(2 + 1.1 \times {Sc_i}^{1/3} \times Re^{0.6}\right) \qquad \textit{(Eq. 76)}$$

For the axial heat dispersion coefficient (Λ_{ax}) and film heat transfer coefficient (α_{film}), the correlations derived by Wakao and Funazkri [110] are used to temporally calculate these coefficients along the adsorptive fixed bed reactor:

$$\Lambda_{\mathrm{ax}} = \lambda_{\mathrm{g}} \times (7 + 0.5 \times Pr \times Re) \qquad \textit{(Eq. 77)}$$

$$\alpha_{film} = \frac{\lambda_{\mathrm{g}}}{2r_p} \times (2 + 1.1 \times Pr^{1/3} \times Re^{0.6}) \qquad \textit{(Eq. 78)}$$

It should be noted that since one certain pellet diameter (3.6 mm) was assumed for all different solid phases, the film mass and heat transfer coefficients, in Eqs. 76 & 78 respectively, were the same for each solid phase.

2.4. Process simulator and numerical approach

Several process simulators are commercially available for simulating complete chemical processes or certain equipment of the process. These process simulators include common chemical process units and a database for physical properties of a wide range of chemical species.

However, less common or new process units are not existing and custom mathematical models should be specifically created. Aspen Custom Modeler (ACM) offers a convenient platform for editing and implementing mathematical modelling of novel process unit operations that can be later integrated in the wide advantageous process simulators Aspen Plus or Aspen HYSYS.

Since the mathematical equations presented in the previous sections of this chapter describe a non-common process unit, namely the adsorptive reactor, Aspen Custom Modeler has been mainly used in this study as process simulator and optimiser for the underlying process.

Besides the mathematical equation editor and property database, Aspen Custom Modeler contains also, as other Aspen simulators, different numerical methods for solving the process equations as well as the possibility to combine Aspen Properties

calculator. So Aspen Properties has been interactively used with the ACM main simulator to evaluate the gas phase properties at each discretising point upon the dynamic as well as the spatial changes of the corresponding pressure, temperature, and component mole fraction of the gas phase during the entire simulation.

Simulation parameters and numerical aspects:

The numerical simulations were executed by implementing the method of lines which is part of Aspen Custom Modeler to solve the model equations. The effective fixed bed length of the simulated adsorptive reactor was one metre, the inner diameter was 0.06 metre, and the diameter of the functional pellets (adsorbent and catalyst) was 3.6 millimetre as Figure 9 illustrates. The axial spatial domain of this reactor was discretised using the backward finite difference method, however exclusively for the cyclic steady state simulations using the central finite difference method. Based on a grid independence test with the objective of reducing the computational effort without compromising the accuracy of the solution, the number of discretising points was chosen to be fifty finite elements. Further refinement of the grid (using more discretisation points) leads to negligible changes in the solution and the simulation becomes computationally prohibitive.

Additionally, the feasible path successive quadratic programming optimiser (FEASOPT) available in Aspen Custom Modeler was employed to solve the dynamic nonlinear constrained optimisation problems, which will be presented and discussed in CHAPTER 4.

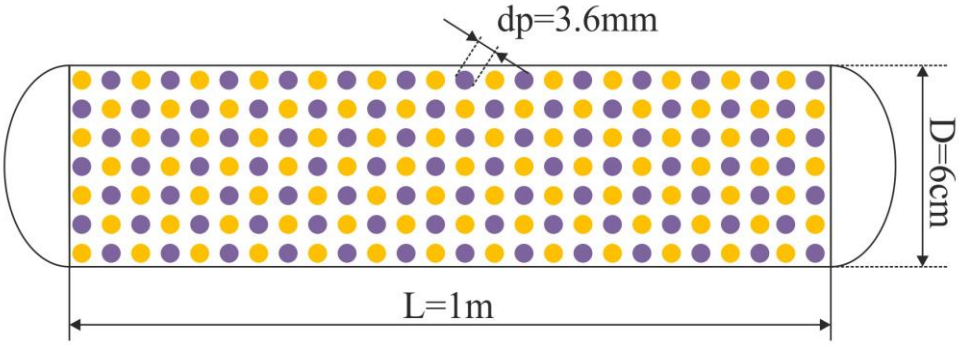


Figure 9: Dimensions of the simulated adsorptive reactor

Finally, for calculating the cyclic steady state of the simulated adsorptive reactor with a quadratic acceleration of the convergence (achieved by efficient estimation of the cycle initial conditions), the direct substitution method using Newton's method has been implemented in place of the computationally expensive and conventionally applied successive substitution approach.

2.5. Model structure in Aspen Custom Modeler

The model equations of the two model types described in the previous sections were written in the ACM language code (see Appendix A and B as examples of the written models) at different model levels to ameliorate the numerical implementation and improve the computational performance of the simulator.

Generally, the main model level is the reactor model, within which the axial length of the fixed bed for the involved reactor equations and sub-models is discretised. Two model levels are then sufficient in case of the one-dimensional pseudo-homogeneous dispersed model as shown in Figure 10.

An additional hierarchical model level was introduced in case of the one-dimensional heterogeneous dispersed model since the governing system equations were discretised at both the reactor and pellet level (Figure 11).

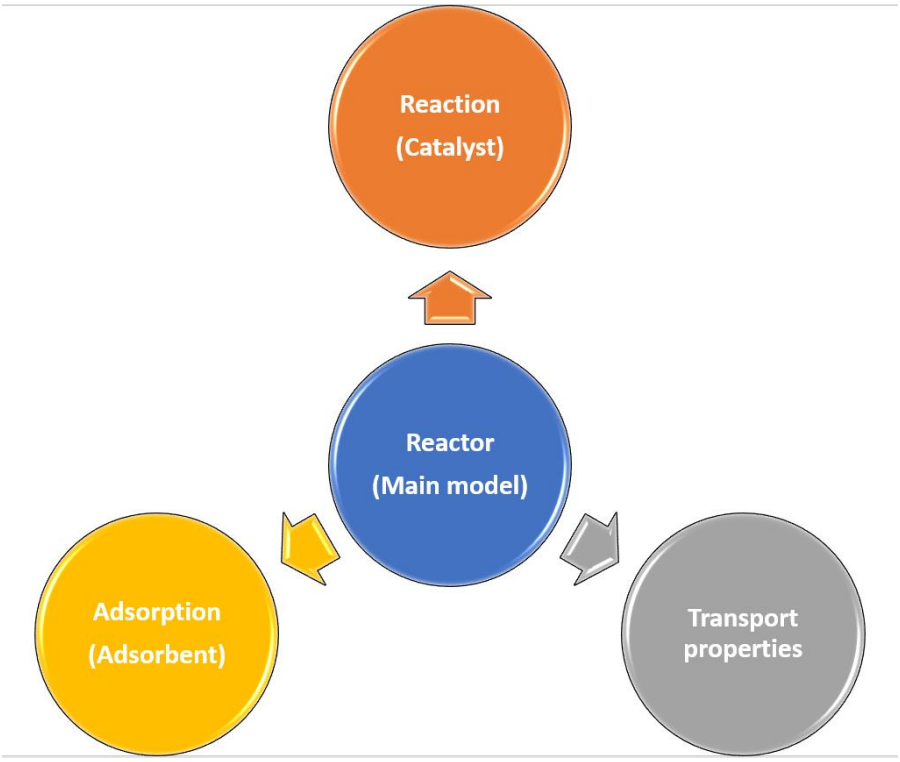


Figure 10: ACM model hierarchy of the one-dimensional pseudo-homogeneous dispersed model

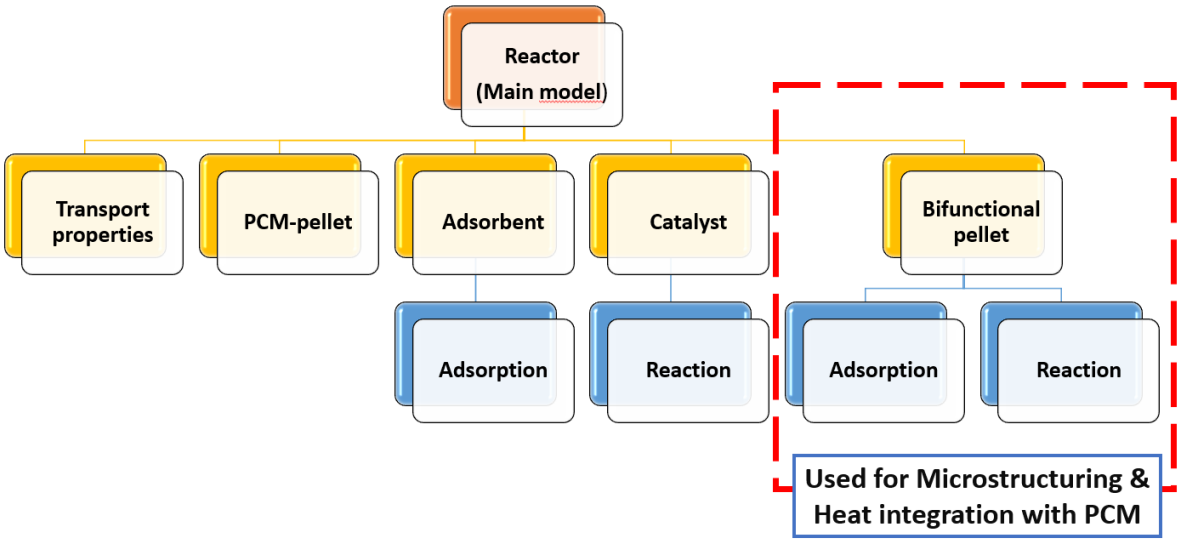


Figure 11: ACM model hierarchy of the one-dimensional heterogeneous dispersed model

CHAPTER 3: PRELIMINARY STUDIES

The advantageous application of adsorptive reactor concepts for industrial processes has been investigated in terms of three categories before detailed studies were to be conducted. These three preliminary studies are presented in this chapter highlighting the high potential of adsorptive reactors for the implementation as a cutting-edge process intensification technology.

3.1. Thermodynamic assessment

Based on reaction kinetics and chemical behaviour of reactants on the catalyst, local segregated functionalities or integrated functionalities in adsorptive reactors could be advantageous. To check the feasibility of local functionality segregation in an adsorptive reactor, thermodynamic simplified calculations of a multistage reaction process with intermediate adsorptive removal of water vapour by-product were carried out for the two test reactions: the Claus- and Deacon-reaction systems. The objective was to ascertain how many alternating reaction-adsorption stages would be needed to achieve the conversions sought, which are primarily dictated by environmental legislation and the extent of any subsequent downstream processing still necessary.

Figure 12 and Figure 13 were calculated for isothermal operation at 250 °C and for an operating pressure of 1 bar to assess the influence of intermediate water vapour removal from the reaction medium. Using the equilibrium constants for the reactions at the temperature 250° C, which are calculated according to Eq. 79 for Claus reaction [23] and Eq. 80 for Deacon reaction [123], together with the law of mass action, the equilibrium curves in Figure 12 and Figure 13 are obtained (the water vapour mole fraction on the x-axis is limited to the equilibrium state at the foresaid operating temperature and pressure). Since the feed gas mixture in the real Claus process contains approximately 5% water vapour, the progress of its mole fraction during the reaction starts from 0.05 for Claus reaction in Figure 12. The progress lines of the water

vapour mole fraction to equilibrium state were calculated from the rate law and stoichiometry of the corresponding reaction.

$$\kappa_{Claus} = exp\left(\frac{33{,}295 - 4{,}91 \cdot 10^{-2} \cdot T}{1 + 3{,}89 \cdot 10^{-3} \cdot T}\right) \qquad \textit{(Eq. 79)}$$

$$\log_{10} \kappa_{Deacon} = \frac{5881.7}{T} - 0.9303 \times \log_{10} T + 1.3704 \times 10^{-4} \times T - 1.7584 \times 10^{-8} \times T^2 - 4.1744 \qquad \textit{(Eq. 80)}$$

In the case of the Claus reaction ($2H_2S + SO_2 \leftrightarrows {}^3/_8S_8 + 2H_2O$), two reaction steps with an intermediate H_2O adsorption proved capable of attaining, at least theoretically, almost complete conversion (Figure 12) and able to meet the stringent residual sulphur specifications for the tail gas without any additional treatment. For the Deacon reaction ($4HCl + O_2 \leftrightarrows 2Cl_2 + 2H_2O$, Figure 13), the theoretical conversion in two reaction stages with intermediate H_2O adsorption is 97.8%, a level at which the resultant gas mixture can be easily processed to yield chlorine which can then be recycled internally to the synthesis reaction from which the HCl originated in the first place.

Even allowing for non-equilibrium conversion in the catalytic reactors, incomplete removal of water vapour in the intermediate adsorbers and a deterioration in performance due to non-isothermal conditions, it would appear that the alternating segregated catalytic-adsorptive functionality is fundamentally feasible for both reactions with a reasonable number of stages. Since the problem of reactant slip does not arise with the Deacon reaction, where excess oxygen can be employed, a more intimate and nuanced integration of the functionalities yielding superior performance may well be possible in this case and the simplified analysis presented based on the partial segregation of the functionalities probably underestimates the potential available.

Nevertheless, the evaluation of the idealised partially segregated adsorptive reactor behaviour offers several appealing features. For one thing it provides a relatively clear and usually somewhat conservative estimate of the feasibility of an adsorptive reactor for the reaction system being considered. Furthermore, it represents a ‘drop-in’ process modification, which is very compatible with the existing conventional non-adsorptive reactor technologies. The Claus and Deacon reactions, for instance, are both moderately exothermic and are commonly carried out in a series of adiabatic

reactors with intermediate cooling. Incorporating an additional adsorber between the reaction stages thus corresponds to a relatively minor re-engineering of the overall process flowsheet, which can even exploit some of the tools already developed to design the optimal alternating adiabatic reactor-intermediate cooler configuration – using the Bellman optimisation principle, for example [111].

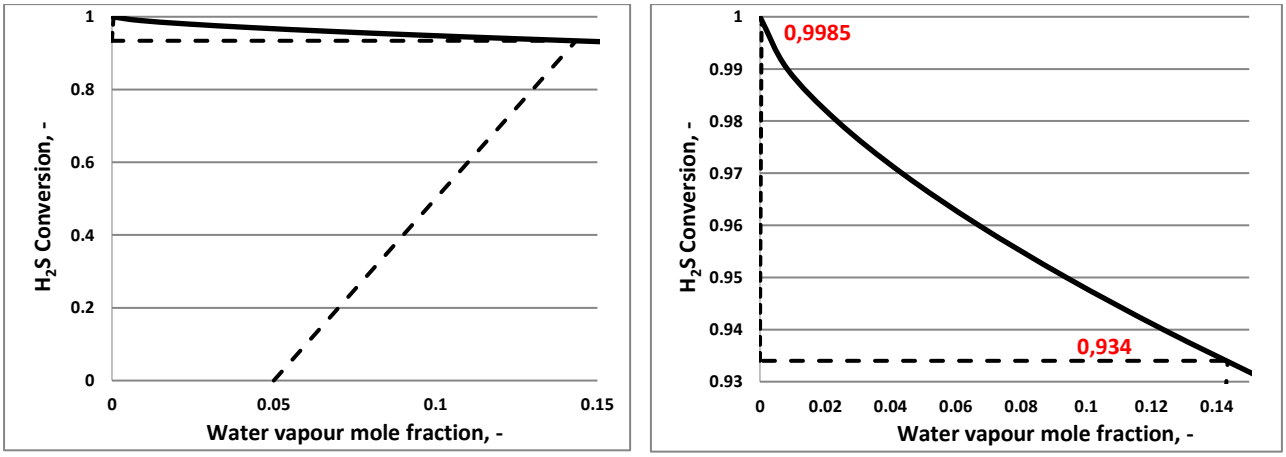


Figure 12: Maximum achievable conversion by two reaction steps with intermediate H_2O adsorption for Claus Process; full scale (left) and zoomed-in scale (right)

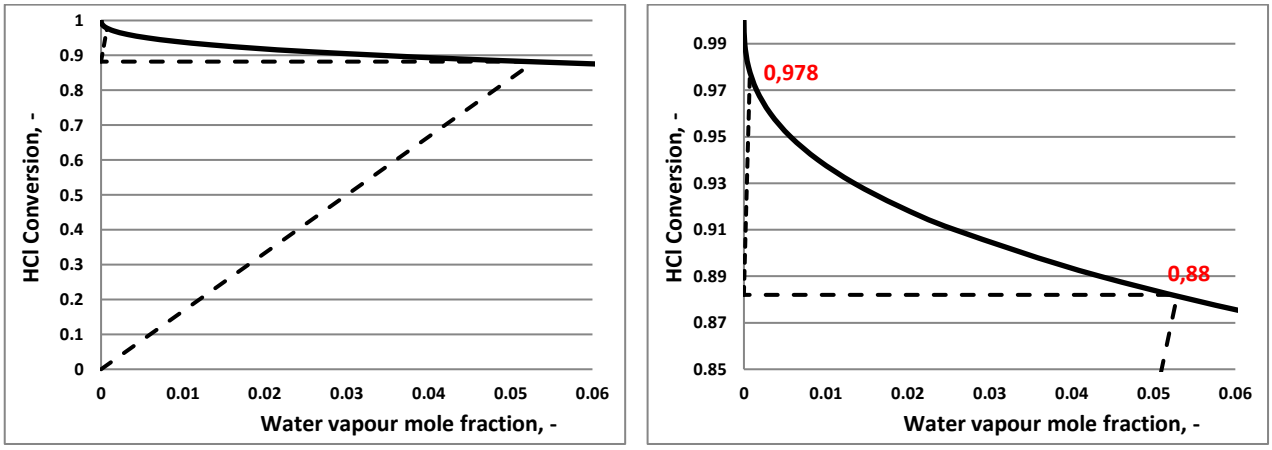


Figure 13: Maximum achievable conversion by two reaction steps with intermediate H_2O adsorption for Deacon Process; full scale (left) and zoomed-in scale (right)

3.2. Reengineering of the process flowsheet

For certain chemical plants, process integration through reactive separation or hybrid separation techniques cannot be established for plant-specific reasons. In this case, reengineering the process flowsheet represents a powerful method for process integration aiming at improved process performance and increased cost savings. One of the reaction schemes considered in this study, namely the Deacon process, can be nominated for reengineering investigation as an alternative to the adsorptive reactor concept as long as no appropriate adsorbent under the prevailing corrosive conditions has been found/developed yet (more details can be read in CHAPTER 5). Thus, a benchmark for adsorptive reactor operation of the Deacon process represented by the more thermally extravagant process configuration, in which the traditional absorptive desiccant - concentrated sulphuric acid - is used in place of high temperature zeolitic adsorption, is considered.

As stated earlier, several process flowsheets of Deacon reaction have been proposed, only few have been commercialised. The most recent one was commercialised by Sumitomo [94], based on which, a chlorine-production plant with a capacity of 300 kta is designed in this work. Design specifications and feed composition of this plant are summarised in Table 6, whereas the sequence of unit operations of the process flowsheet is shown in Figure 14.

Table 6: Design specifications of the 300 kta synthesised plant

Cl_2-Purity, mol.%	Capacity, kta	Feed Composition, Vol. %		
99.5	300	*HCl*	*O_2*	*N_2*
		20	17	63

The designed Deacon process can be categorised into six operational units (see Figure 15): the chemical reactors, HCl absorption, HCl recovery and recycling, drying, off-gas treatment, and chlorine purification.

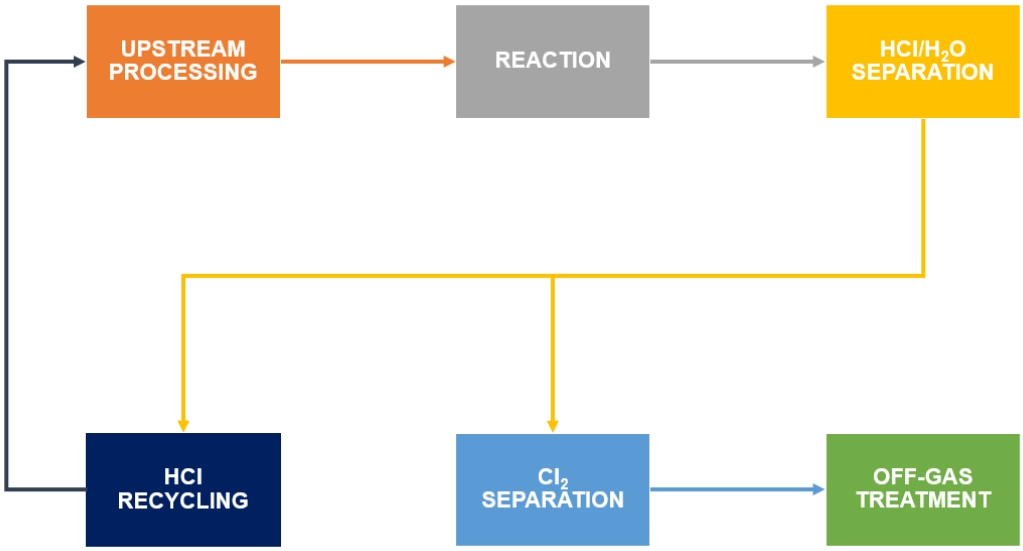


Figure 14: Sequence of the unit operations for conventional Deacon process

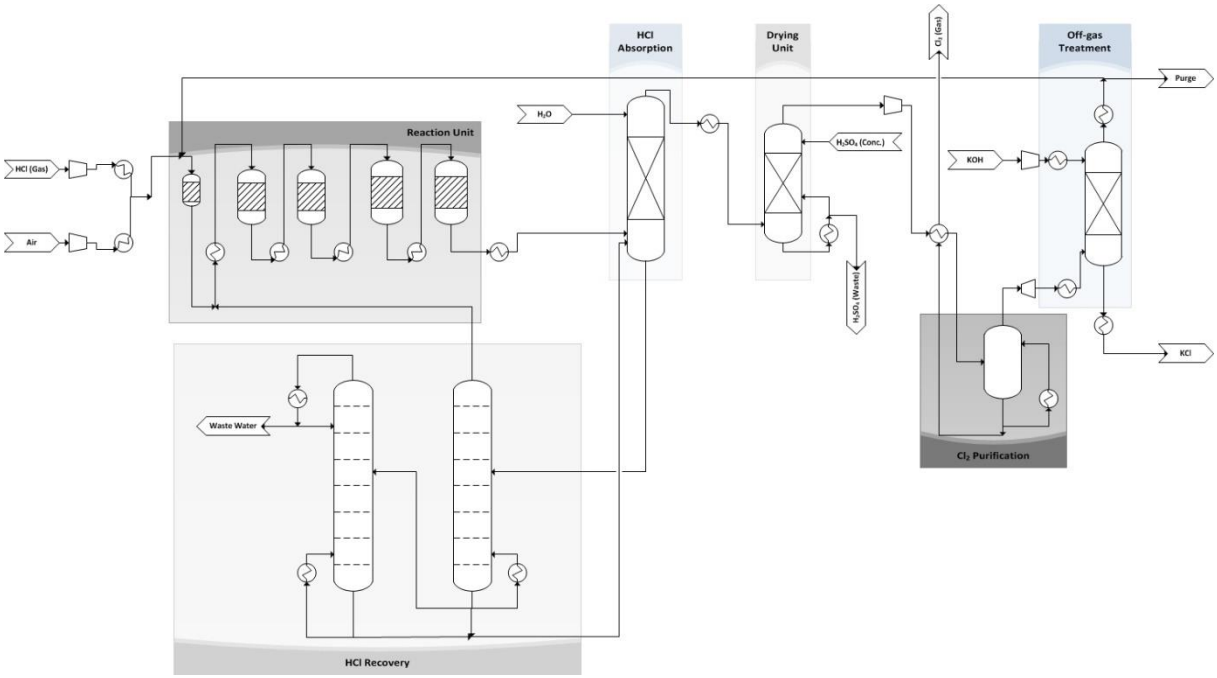


Figure 15: Plug flow diagram of the chlorine production plant simulated in this study

The chemical reactors' unit:

the reactant gases (hydrogen chloride with an excess of oxygen) is heated up to the temperature of the first reaction stage 310°C and then fed into the reactors' unit. The reactors are to be operated under 10 bars. Five adiabatic reaction stages with an intermediate cooling were determined to reach the designed conversion of 82%. Figure 16 shows the stage construction from the X-T-diagram at 10 bars considering the HCl-recycle stream and the safety margin to the maximum allowed temperature (~500°C), beyond which the catalyst would be unstable.

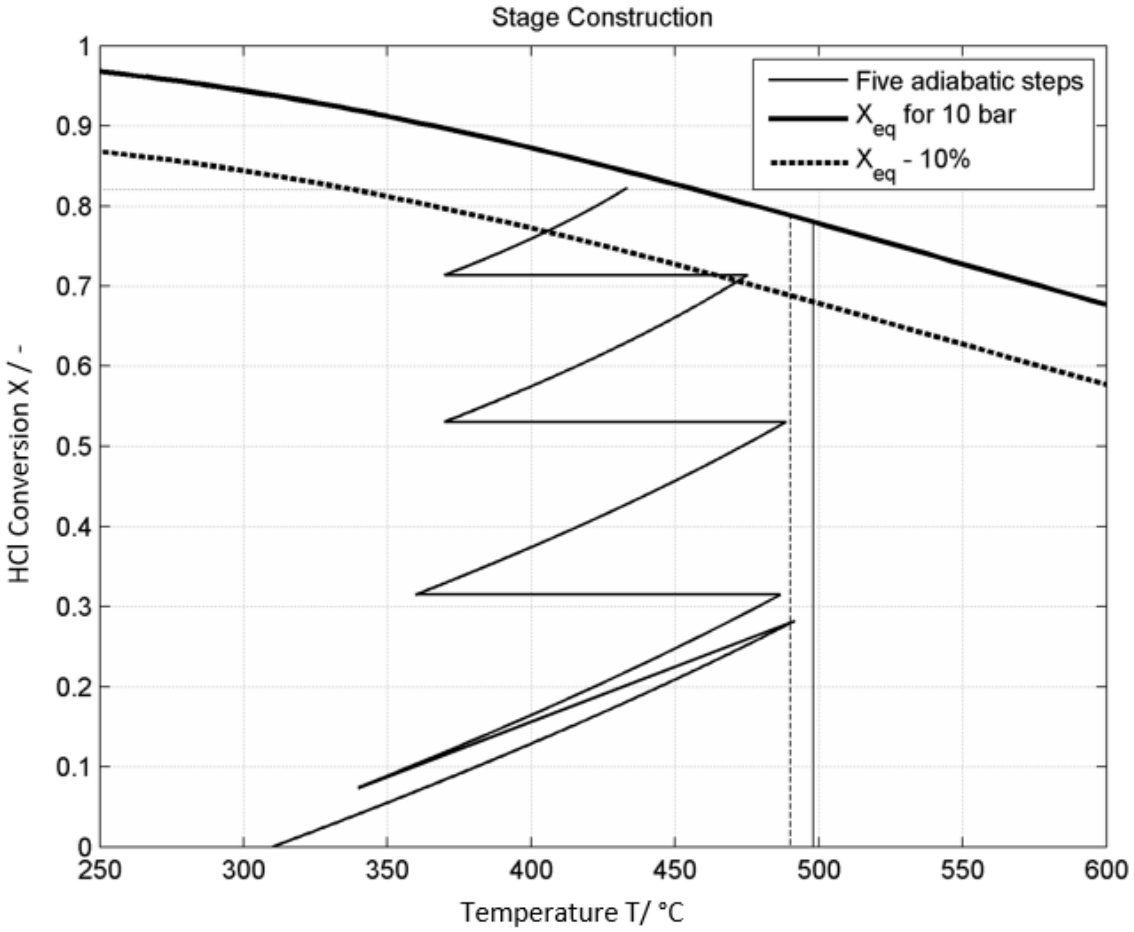


Figure 16: Conversion-temperature-diagram at 10 bars for the stage construction of the HCl oxidation reaction

HCl absorption:

the outlet from the last reaction stage will be then fed into an absorption step to separate more than 99% of the unreacted HCl by a liquid stream of water. A concentrated hydrochloric acid (HCl concentration achieved: 22.5 mol.%) is collected at the bottom of the packed column and passed through the recovery unit.

Recycling unit including the "azeotropic distillation":

the HCl-water mixture forms an azeotropic point at 108.6°C and 1 bar rendering the separation of such a mixture in a single distillation stage difficult. Therefore, a dual-pressure distillation unit, which is called here as an "azeotropic distillation", has been suggested for recovering the unreacted hydrogen chloride.

The hydrochloric acid stream is fed to a distillation column operated at a high pressure (10 bar), where the hydrogen chloride from the top of the column is recycled back to the reaction unit. Whereas the weak hydrochloric acid collected at the bottom of this distillation column is directed to the second distillation column operated under low pressure (0.1 bar). In this process step, the water is distilled from the top and recycled to the HCl-absorption tower.

Drying unit:

the water vapour still contained in the gas stream from the top of the HCl-absorption tower is separated by contacting this gas stream with a concentrated sulphuric acid (98 wt. %). The liquid content of the gas stream is reduced to below 20 ppm in this unit and thus can be directed safely to the next chlorine purification unit.

Chlorine purification unit:

like the chlorine purification process developed by Mitsui Toatsu Chemicals Inc. in Japan, the gases are initially compressed to 16 bar and are subsequently cooled to a temperature -85°C resulting in a condensate consisting of 93.7 mol.% purified chlorine. In order to further increase the chlorine purity and reach the design specification of 99.5 mol.%, the condensate from the liquefaction step is allowed to fall through a

stripping column maintained at the same pressure as the liquefaction unit, namely 16 bar.

Off-gas treatment unit:

the traces amount of chlorine in the tail gas is then separated in a reactive absorption column using a potassium hydroxide solution producing potassium chloride which can be sold as by-product.

Reengineering the process flowsheet described above, in which the drying step using the concentrated sulphuric acid is performed as an intermediate stage between only two reaction steps, has shown significant improvement of the overall flowsheet performance reflected by achieving higher conversions and thus consequently by lower operating and investment costs as shown in Table 7. The reported savings are mainly achieved by reducing the necessary multi-stage reactors due to the increased conversion obtained. This alone has a very big impact on the investment costs as the ruthenium-based catalyst used is very expensive. Additionally, the higher achieved conversion facilitates the downstream processing of the product/unreacted hydrogen chloride mixture and avoids the costly separation by the dual-stage distillation unit.

Table 7: Improvements achieved by re-engineering of the Cl_2 production conventional flowsheet

Operational improvement		**Economic improvement**	
Conversion, %		***Save in operational costs, %***	***Save in investment costs, %***
Conventional flowsheet	**Modified flowsheet**		
82	94	20	25

3.3. Adsorptive reactors as powerful technology for process intensification

Like any cyclic process, one of the most pertinent performance indicators of the adsorptive reactor is the cycle time (τ), which is simplificatively defined as the multifunctional operation (reaction-adsorption) time until a minimum conversion of the reaction cannot be maintained, or a regeneration of the adsorbent becomes necessary. The adsorptive reactor productivity and the extent of adsorbent utilisation can thus be indicated by the cycle time since the volume of the simulated fixed bed reactor and its feed conditions were kept fixed.

The minimum required conversion mentioned above is basically specified upon either environmental legislations as for Claus reaction (99.5% according to the German standards) or upon the extent of any downstream separative processing still needed as for Deacon reaction (96%).

The simplest structure of the involved functionalities within the reactor is a uniform distribution, in which the volume fraction of the adsorbent "φ_{ads}" at each position of the adsorptive reactor has a certain fixed value. The optimal value of the adsorbent volume fraction (43%) reported in [24, 91] was taken as a reference case for both Claus and Deacon reaction systems to benchmark the obtained results.

In their optimisation study of the functionality distribution at the reactor level, Lawrence et al. [24] considered thirteen different segments of the adsorptive Claus reactor and found that there exist three distinct functionally dominant regions in the reactor, two kinetically controlled zones at both ends of the reactor and one adsorptive in between. Based on this finding, a parametric study in which the length of each of the foresaid zones (Figure 17) were varied and the reactor performance (indicated by the cycle time τ) was analysed for an isothermal Claus and Deacon adsorptive reactor at 250 °C and 300 °C, respectively. The objective was to find out the best volume fraction of both involved functionalities along the reactor. Table 8 summarises some of the considered variations including the best structure.

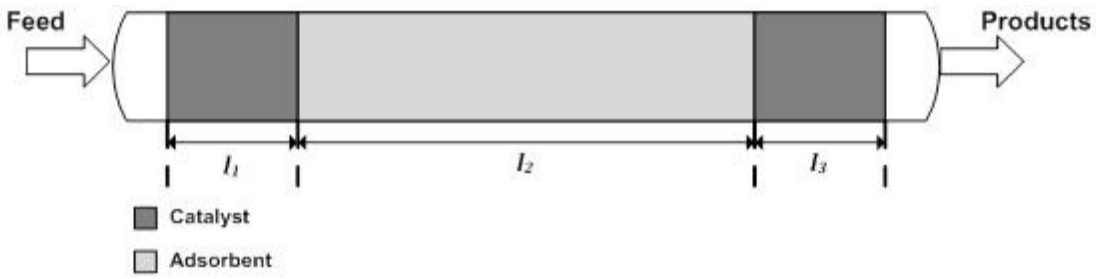


Figure 17: Three segments of variable lengths for the spatially segregated structure of adsorptive and reactive functionalities

As the results in Table 8 emphasise, the best volume fraction of the reactive and adsorptive functionalities were found by avoiding, on one hand, the kinetic limitations that cause the conversions to be well short of equilibrium (the case where too little catalyst exists), and on the other hand the rapid saturation of the existing adsorbent which urges the regeneration process and leads to brief cycle times.

Table 8: Different volume fractions of the adsorptive and reactive functionalities with obtained cycle times in seconds; the values highlighted represent the best structures found

l_1 [m]	l_2 [m]	l_3 [m]	τ [s]	
			Claus	Deacon
0.3	0.5	0.2	1240	876
0.05	0.85	0.1	192	1170
0.1	0.85	0.05	92	1170
0.1	**0.8**	**0.1**	**2040**	**1180**
0.1	0.6	0.3	1540	1120
0.4	0.5	0.1	1140	786
0.15	0.7	0.15	1810	1110
0.4	0.2	0.4	387	679

The resulting best sandwich structure, in which a 0.8 m adsorptive layer in the middle of the reactor is surrounded by two reactive layers of 0.1 m each, enhanced the performance of the adsorptive Claus and Deacon reactor by 65% and 155% respectively over the corresponding reference case (where the functionalities are uniformly distributed for both reactions as stated earlier; the benchmark cycle time was 1240 s for Claus reaction, and 533 s for Deacon reaction).

As it has already been shown, macrostructuring the reactive and adsorptive functionalities led to a significant improvement of the reactor performance, indicating the high potential of the adsorptive reactor concept for process intensification (a general overview of the state of the art of chemical engineering approaches for process intensification is given in [82]), since several more degrees of freedom are still available to be considered in the process design. These include temperature profiling, dynamic profiling of the operating parameters, distributed feeds, multiscale heat integration, and microstructuring (distribution of the functionalities at the pellet level).

The topic of tapping into more degrees of freedom for adsorptive reactor design in combination with macrostructuring is presented in the following chapter. This yields generally to a dynamic optimisation problem with the purpose of evaluating the maximum improvement of the adsorptive reactor that could be attained in comparison to the simple uniform functionality distribution operated under conventional isothermal or adiabatic operating conditions.

CHAPTER 4: DESIGN AND OPERATION OF ADSORPTIVE REACTORS

Several degrees of freedom are available in design and operation of adsorptive reactors. One of these is structuring the involved functionalities at the macro-level (i.e. reactor-level) as presented in CHAPTER 3. Other design and operational alternatives include, but not restricted to, structuring the functionalities at the micro-level (i.e. pellet-level), temperature profiling, dynamic profiling of operating parameters, distributed feed, multiscale heat integration, multi-dimensional functionality distribution, and moving bed operation. Except for the moving bed operation of adsorptive reactors, the foresaid design and operational techniques are highlighted in this chapter. The most promising design is then introduced and proposed for future consideration when operating adsorptive reactors.

4.1. Temperature profiling to expedite macrostructuring

The choice of the operating temperature profiles in conjunction with macrostructuring is a powerful tool for accommodating the contradictory demands of the reaction and adsorption processes since they exhibit conflicting temperature behaviour. This combination was addressed by setting a dynamic optimisation problem to find out the optimal adsorptive reactor structures with spatially profiled operating temperature for Claus and Deacon reaction schemes and for both isothermal and adiabatic operational modes.

4.1.1. Optimisation procedure

In light of the work of Lawrence et al. [24] and the discussion and results presented in the previous chapter, the simulated adsorptive reactor was subdivided into three segments, within which the optimal temperature and adsorbent fraction in the case of the isothermal operation mode were found by the optimiser. Whereas in case of optimising the three-segments conventional adiabatic operation, the central segment was operated isothermally and the temperature in this segment could float and to be chosen by the optimiser along with the adsorbent fraction, where the other two segments were operated adiabatically and only their adsorbent fraction could be freely varied. The following mathematical representation of the optimisation problem, in which maximising the cycle time over which a given conversion could be maintained ($X_{\min}$) was the objective function, is written for the isothermal operation:

Objective function:

max. τ

s.t.:

Simulation equations described in CHAPTER 2

$0 \leq \varphi_j \leq 1;\ j = 1,2,3\ (\text{segment number})$

$$\sum_j \varphi_j = 1$$

$473\ \text{K} \leq T_j \leq 673\ \text{K};\ j = 1,2,3$

$actual\ conversion, X(t) \geq\ X_{\min}$

Design variables:

$\varphi_j;\ j = 1,2,3$

$T_j;\ j = 1,2,3$

4.1.2. Optimal adsorptive reactor designs

The optimisation resulted in two optimally macrostructured designs for the isothermal and adiabatic operation of the adsorptive Claus and Deacon reactors. On one hand, the multilevel isothermal sandwich design (Figure 19), and on the other hand the

central isothermal sandwich design (Figure 20) are proposed in place of the simple structure (uniform distribution of the functionalities over the fixed bed) isothermal and adiabatic adsorptive reactor (Figure 18), respectively. Due to the fact of conflicting temperature dependencies of both adsorption and reaction functionalities, the choice of the optimal operating temperature profiles can only be accomplished when optimal spatially segregated reactive and adsorptive functionalities is used. This optimal functionality structure thus comprises a purely adsorptive layer (80% of the fixed bed) between two peripheral catalyst zones (20% of the active fixed bed) at the ends of the reactor confirming the structure found in CHAPTER 3. The values of the optimised design variables in case of isothermal and adiabatic operation modes are shown in Figure 19 and Figure 20 for Claus and Deacon reaction systems studied, respectively.

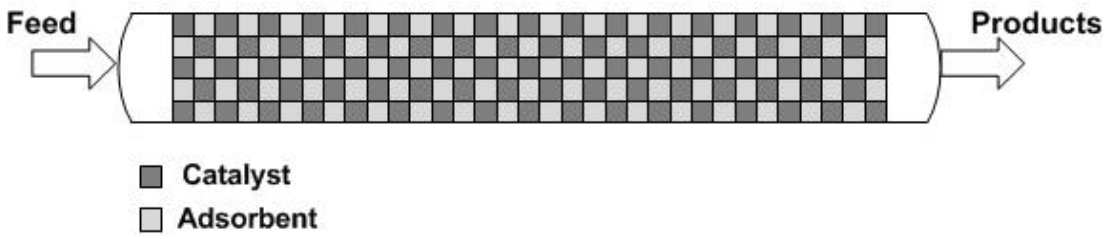


Figure 18: The isothermal/adiabatic adsorptive Claus/Deacon reactor with uniformly distributed functionalities (benchmark case)

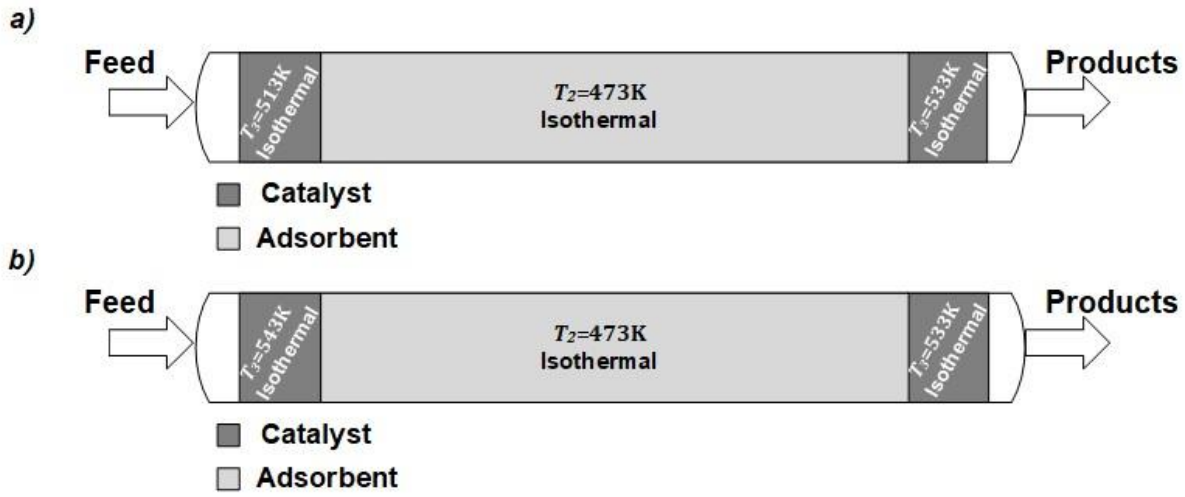


Figure 19: Novel proposed design (the multilevel isothermal sandwich design) for the isothermally operated Claus (a) and Deacon (b) adsorptive reactor

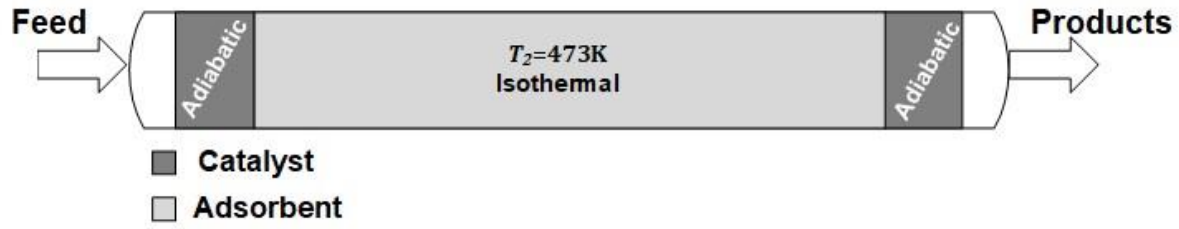


Figure 20: Novel proposed design (the central isothermal sandwich design) for the adsorptive Claus and Deacon reactor operation

The obtained cycle times are plotted against the ones obtained by the benchmark case for Claus reaction system (Figure 21) and for the Deacon reaction scheme (Figure 22). It can be clearly seen that the multilevel isothermal sandwich design improved the reactor performance by c. 600% in case of adsorptive Claus reactor, and by a thirty-fold for the adsorptive Deacon reactor, while the central isothermal design extends the cycle times for Claus reaction by roughly 400% and for Deacon by a factor of twenty-five.

In the reactor inlet, the conversion is well below the equilibrium value, so the reaction cannot profit from adsorptive by-product removal. Only when the kinetics become strongly influenced by the reverse reactor does an integrated adsorption become expedient. To achieve the best possible kinetics in the subsequent catalytic zone, the removal of the by-product from the reaction medium in the central adsorptive zone should be as complete as possible. This is achieved by allowing the adsorption to take place at a lower temperature. However, lowering the adsorption temperature is constrained by several technical issues.

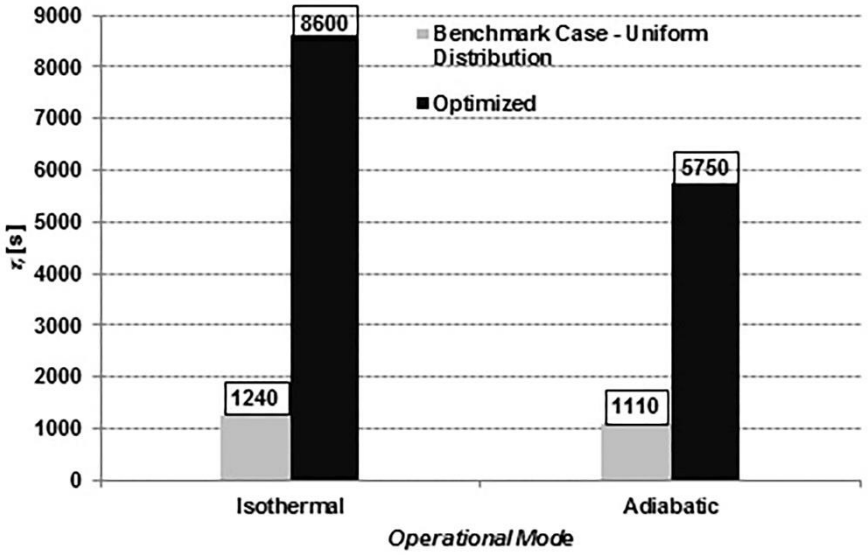


Figure 21: Obtained cycle times from the multilevel isothermal and central isothermal sandwich designs for Claus reaction in comparison to those obtained from the benchmark case

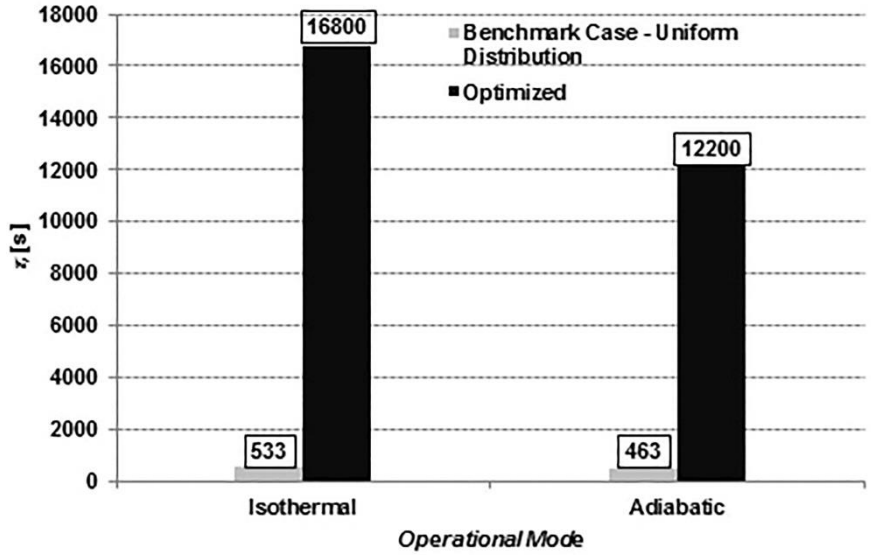


Figure 22: Obtained cycle times from the multilevel isothermal and central isothermal sandwich designs for Deacon reaction in comparison to those obtained from the benchmark case

It is worthy to note that for Claus reaction an 89% of the improvement achieved by the novel proposed designs reported above and 96% in case of Deacon reaction were contributed by combining temperature profiling with macrostructuring.

4.2. Multiscale functionality distribution

Macrostructuring the adsorptive and reactive functionalities along the axial length of the fixed bed was presented in CHAPTER 3. Another possibility to distribute the involved functionalities efficiently is the so-called microstructuring, in which they can be structured at the pellet level and thus become multifunctional (bifunctional) structured pellets. Studies on microstructuring to enhance macrostructuring, functionality distribution at the reactor as well as at the pellet level simultaneously, have been conducted for the adsorptive Claus operation using a one-dimensional, heterogeneous, dispersed, and non-isothermal adsorptive fixed bed reactor model taking the benchmark case here to be a simple uniform distribution of the functionalities at both pellet and reactor level. Thus, a multiscale distribution of the reactive and adsorptive functionalities of several segments was simulated (up to 10 segments at the reactor level and up to 6 sections at the pellet level) and the best structure was determined. Figure 23 shows an example of these structures with their obtained cycle times. It was found, that by four structured sections at pellet level and five at reactor level, c. 440% prolonged cycle time could be attained over the benchmark case. More structured sections establish an asymptotic improvement as can clearly be seen in Figure 24. However, microstructuring alone contributes only with 13.5% improvement (since by microstructuring, certain transport limitations can be circumvented) over the multiscale uniform distribution rendering this degree of freedom (microstructuring) unattractive in the design process as a de-bottlenecking concept under the considered conditions.

Having multifunctional pellets for Deacon reaction, on the other hand, was not an option since the water vapour has a strong inhibition effect on the ruthenium-based catalysts [96] and therefore an integrated functionality at the pellet level would have a negative effect on the catalytic functionality in the reactor.

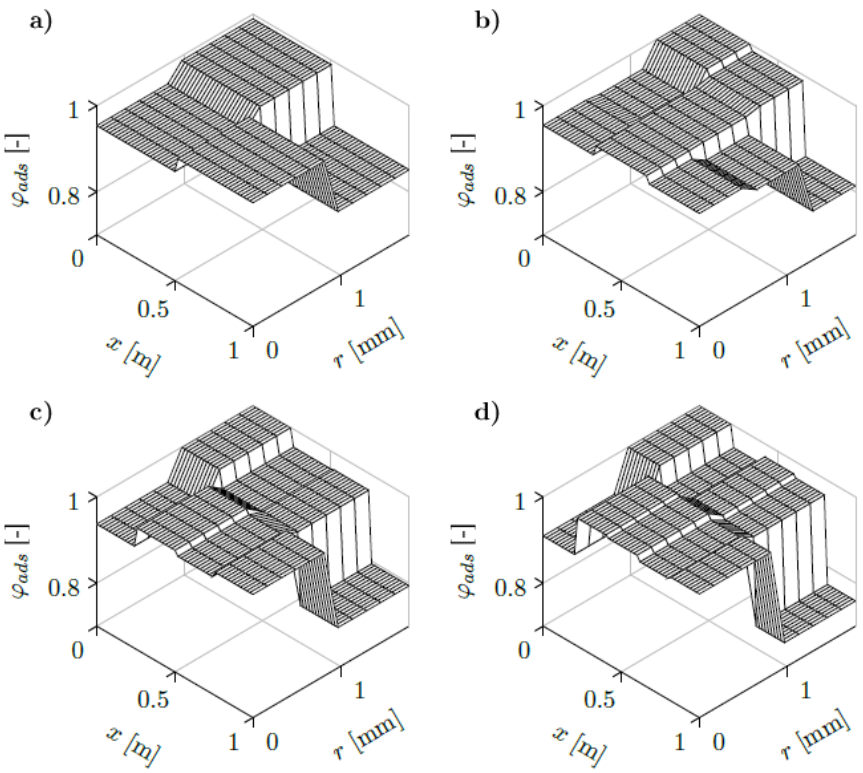


Figure 23: Multiscale structures of functionality distribution. y-axis: adsorbent volume fraction, x_1-axis in [m]: length of the fixed bed, and x_2-axis in [mm]: radius of the multifunctional pellet. a) τ = 4791 s (R=2, P=2), b) τ = 4848 s (R=3, P=3), c) τ = 4875 s (R=4, P=2), d) τ = 4902 s (R=5, P=2); P: number of discrete segments at the pellet level, and R: number of discrete segments at the reactor level

4.3. Multi-dimensional functionality distribution

The macrostructuring of the reactive and adsorptive functionalities introduced in CHAPTER 3 was done for the axial length of the fixed bed. In this section, the findings resulted from studies on the functionality distribution at the reactor radial level in addition to the distribution at the reactor axial length, the multi-dimensional macrostructuring, of an adiabatic adsorptive Claus and Deacon reactor are highlighted.

The mass and energy balance equations as well as the momentum balance equation presented in CHAPTER 2 were discretised radially on eight discrete points using the backward finite difference method. The model equations as well as the correlations for

calculating the radial mass and heat dispersion coefficients can be found in Appendix C written in the Aspen Custom Modeler code language.

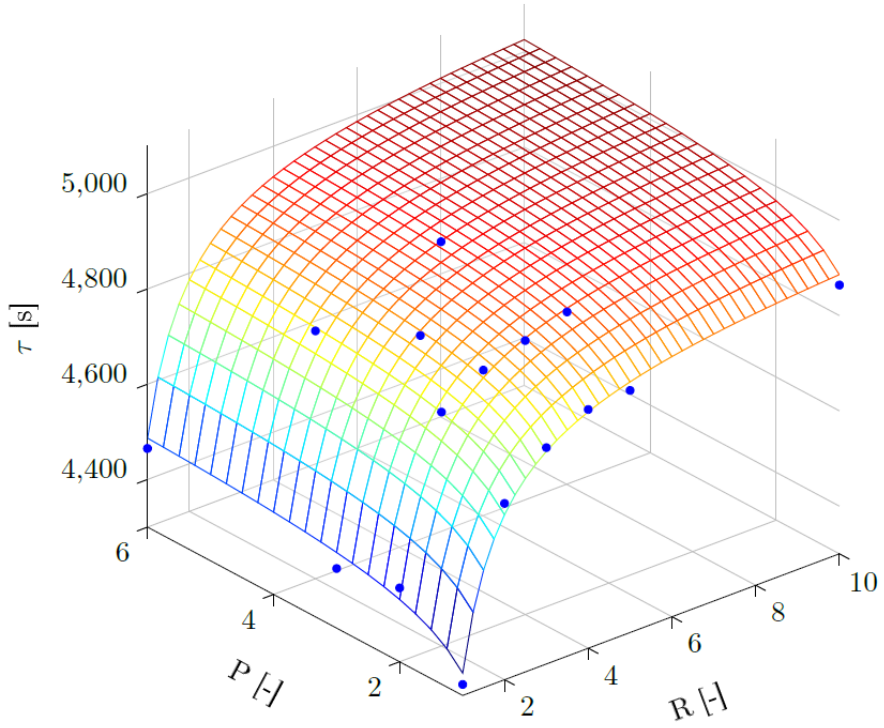


Figure 24: Obtained cycle times of the different multiscale segmented functionality structures for Claus reaction; P: number of discrete segments at the pellet level, and R: number of discrete segments at the reactor level

Figure 25 and Figure 26 show schematically the best multidimensional macrostructures of the adsorptive Claus and Deacon reactor, respectively. The following points were concluded:

- Best structures were found when the involved functionalities were spatially segregated.
- For Claus reaction, the radial and axial distinguished kinetically controlled sections at both ends of the reactor render radial structuring unnecessary.
- For Deacon reaction, radial imposition of sorption regime on both axial kinetically controlled zones (both axial ends of the reactor) and existence of central kinetically controlled section in the middle of the reactor helped to

achieve better reactor performance in comparison with the multi-dimensional uniform distribution (benchmark case) by offering better progression of the adsorbate concentration fronts (see Figure 27) since the water vapour has an inhibition effect on the catalytic functionality, as mentioned earlier.

- The results obtained reveal only slight radial variations of the concentration and temperature profiles for the reactor size used. Hence, the radial macrostructures here highlight in which extent and shape the best radial functionality distribution lies, should the radial changes of the process states be considerable.

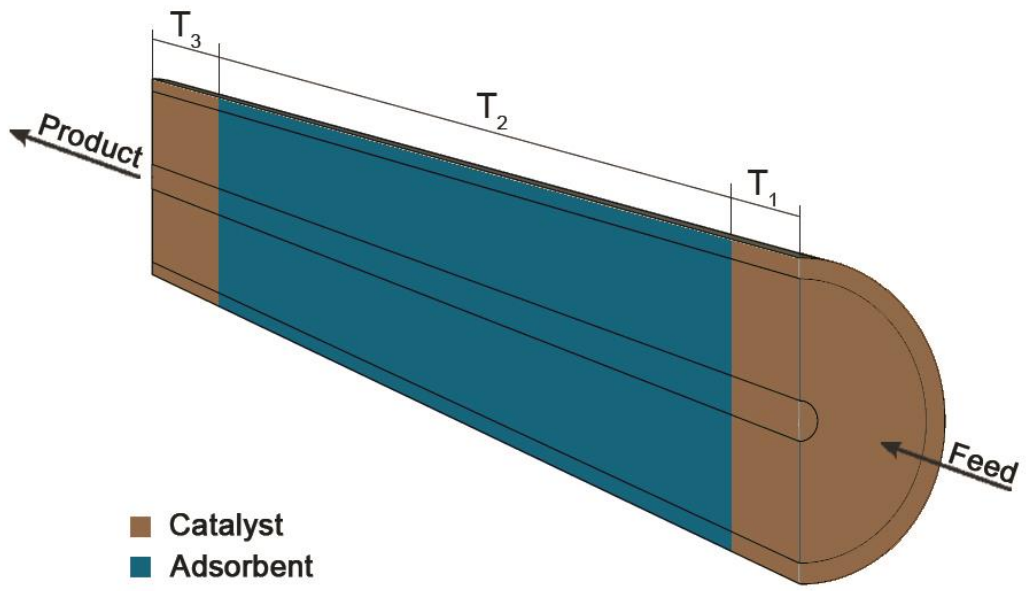


Figure 25: Best multi-dimensional macrostructure of the adsorptive Claus reactor

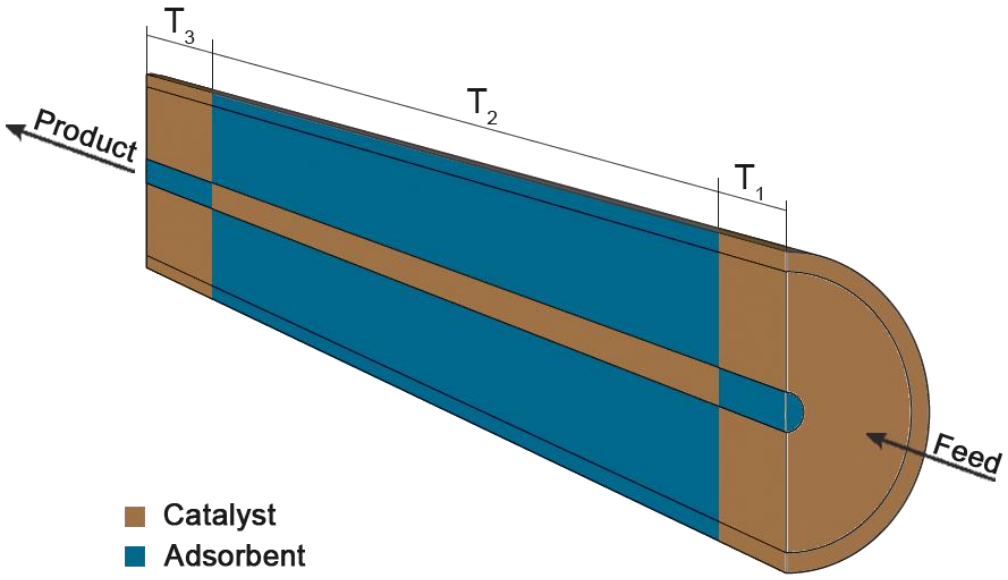


Figure 26: Best multi-dimensional macrostructure of the adsorptive Deacon reactor

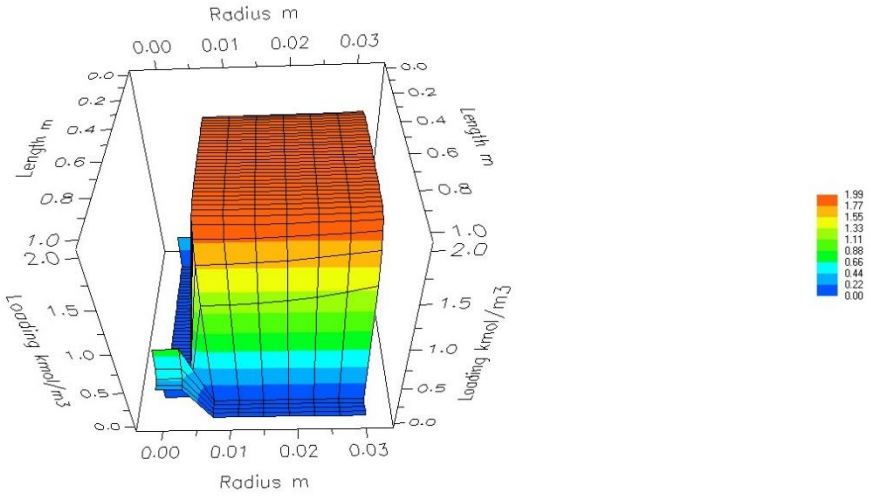


Figure 27: Radial and axial loading profile of the best multi-dimensional structure of the adsorptive Deacon reactor

4.4. Heat integration (HI) concepts

To further exploit the degrees of freedom available in improving the operational performance of adsorptive reactors, two heat integration approaches, at reactor and pellet level shown schematically in Figure 28 and Figure 29 respectively, based on utilising the heat liberated in reaction step to regenerate the adsorbent were proposed and evaluated. The reactor level approach applies recuperatively a heat source to enhance the regeneration step of one of the periodically operated adsorptive reactors. This heat source receives heat from the product gas stream of the other adsorptive reactor which is under non-isothermal reaction/adsorption step of the cycle. On the other hand, the pellet level approach comprises heat storage particles, PCMs (Phase Change Materials) [112], which release the stored heat during the regeneration step enhancing therefore the overall cycle performance. The PCM used in the simulation (lithium I nitrate, see Table 9 for its used properties) was chosen based on various selection-criteria such as: possession of a large latent heat within the operational temperature window, safety aspects, and chemical and thermal stability. Whereas the reactor level heat integration concept was modelled by the pseudo-homogeneous model, the heterogeneous model was used for the pellet level one; both presented in CHAPTER 2 of this thesis.

Table 9: PCM phase properties

Property	Value	Units	Reference
ρ_s	2366	kg/m^3	[113]
ρ_l	2.068-0.546·10^{-3}T [K]	kg/m^3	[114]
$c_{p,PCM,sol}$	0.585+2.182·10^{-3}T [K]	J/(g·K)	[114]
$c_{p,PCM,liq}$	1.681+6.389·10^{-4}T [K]	J/(g·K)	[114]
λ_s	1.4	W/(m·K)	[115]
λ_l	[13.9+0.0145·(T-252)] ×0.0418	W/(m·K)	[116]
$\Delta_{fusion}H$	373	J/g	[114]

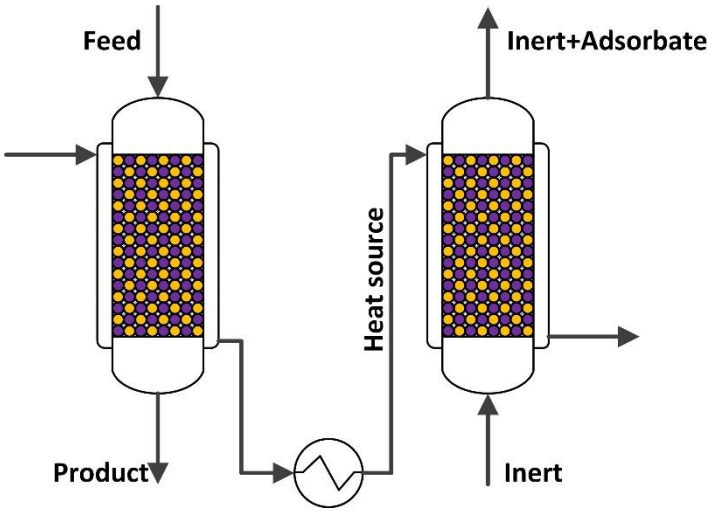


Figure 28: Reactor level heat integration approach for two periodically switching adsorptive reactors; one under adiabatic reaction/adsorption step (left) and the other under regeneration step (right)

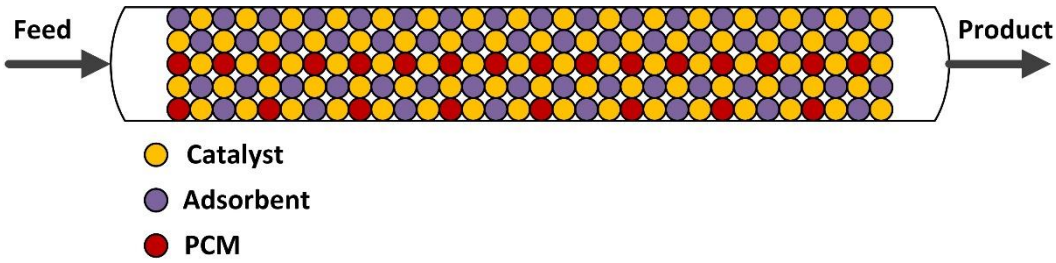


Figure 29: Pellet level heat integration approach using PCM; during the regeneration step the reactor is fed with an inert

Both the concepts can prolong the reaction/adsorption cycle time of the Claus adsorptive reactor beyond the benchmark case (the simple homogeneous distribution and non-isothermal operation). It can be clearly seen in Figure 30 that the locally segregated regeneration procedure enhances the total cycle improvement up to 245% contributing 49% of the heat duty required, whereas the micro heat integration procedure slows down the desorption/cooling step and thus limits the improvement of the total cycle time to only 3% (Figure 31). It is then clear that for future heat integration possibility when scaling up the adsorptive reactors, the macro-level heat integration to be considered.

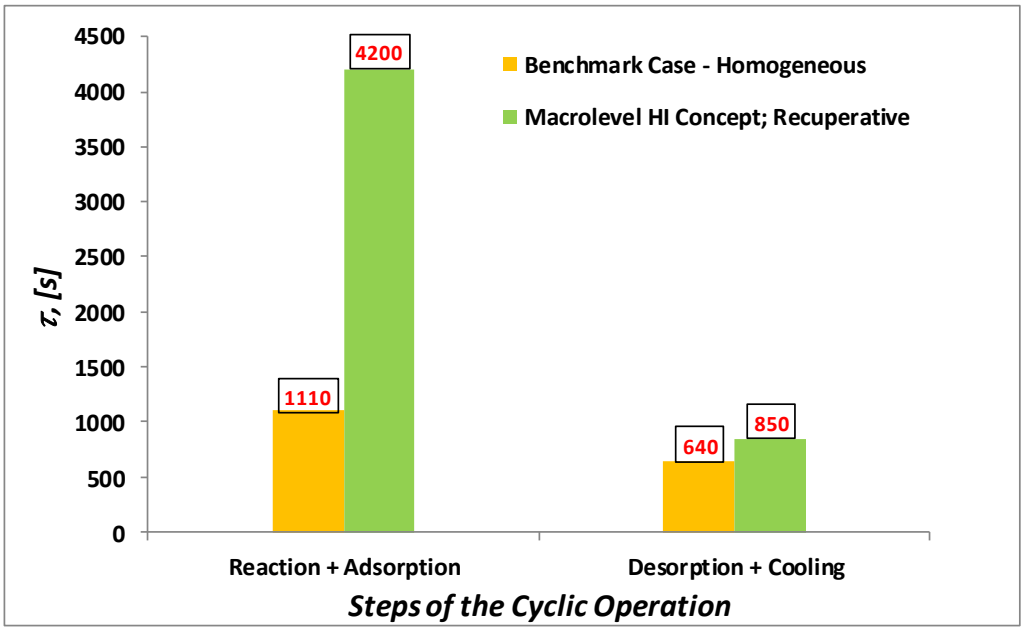


Figure 30: Cycle improvement resulted from applying the reactor level heat integration concept

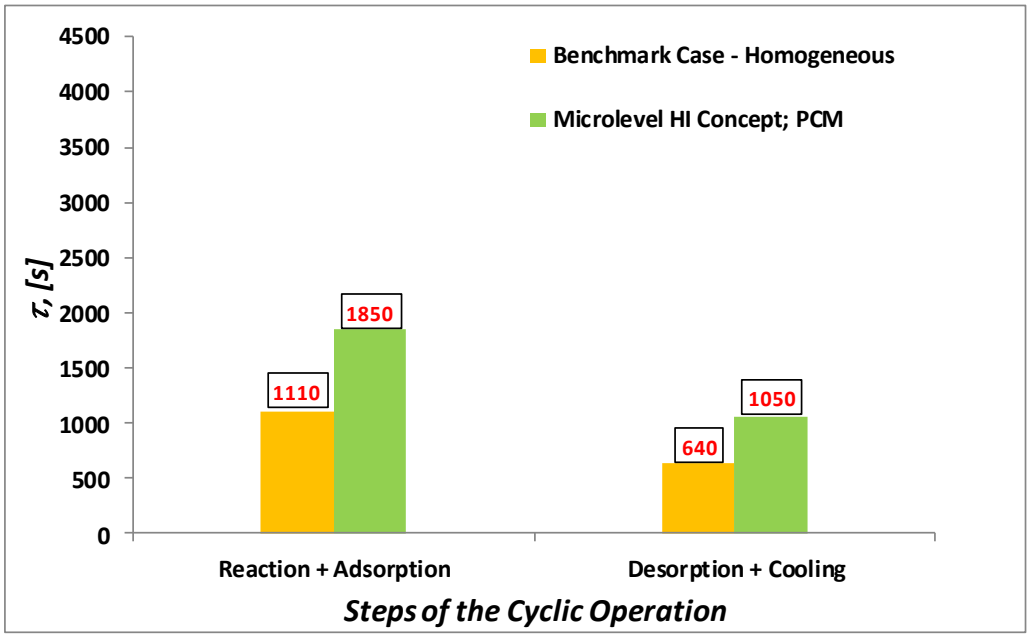


Figure 31: Cycle improvement resulted from applying the pellet level heat integration concept

4.5. Dynamic profiling of operating parameters

Another degree of freedom to be further used in design and operation of adsorptive reactor is the temporal profiling, in which process operating variables (such as feed velocity, concentration, temperature and/or pressure) are dynamically profiled. It is addressed here to a parametric study to elucidate the possible tendency towards improved adsorptive reactor concepts.

In this work, the following three operating variables were chosen to be dynamically profiled during the reaction-adsorption cycle step (see also Figure 32):

(i) feed molar flow rate, i.e. off-gas molar flow rate (using a buffer tank),

(ii) molar flow rate of the nitrogen to further dilute the system, and

(iii) inlet temperature.

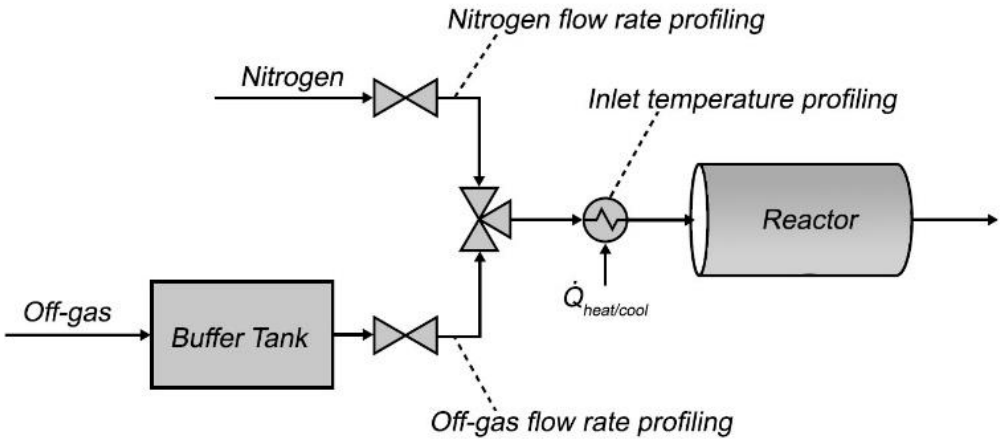


Figure 32: In-process illustration showing the operating parameters to be dynamically profiled

These parameters can be restrictedly profiled. From industrial point of view, the off-gas flow rate fed to the Claus reactor unit has a predefined value. To meet this condition while temporal profiling, the mean value of the profiled off-gas flow rate should remain constant by using a buffer tank, for instance. Additionally, the nitrogen can be added to the system under consideration (and thus be diluted) but cannot be

removed. Finally, too low or too high inlet temperature values are to be technically and/or operationally avoided. The benchmark (BM) case chosen in this study is an adsorptive Claus reactor with a uniform functionality distribution and dynamically constant operating parameters. Table 10 summarises the values of these parameters for the benchmark case. Based on this, the parametric space, within which the operating variables can be temporally profiled, for each above-mentioned variable was determined (Figure 33).

Table 10: Values of the operating parameters used for the benchmark case

Variable	**Value**
P_{in}, [bar]	1
T_{in}, [K]	523.15
n_{in}, [mol/h]	16.85
$y_{N2,f}$, [-]	0.85
$y_{H2S,f}$, [-]	0.1
$y_{SO2,f}$, [-]	0.05

Different mathematical functions can be used to shape the profile of each considered operating parameter. Sigmoid curves [117], which allow smooth S-shaped profiles as shown in Figure 34, was applied. The temporal profiling of a variable can be done from high to low or from low to high values or with constant values higher or lower than those of the benchmark case as long as there are no limiting constraints as discussed above.

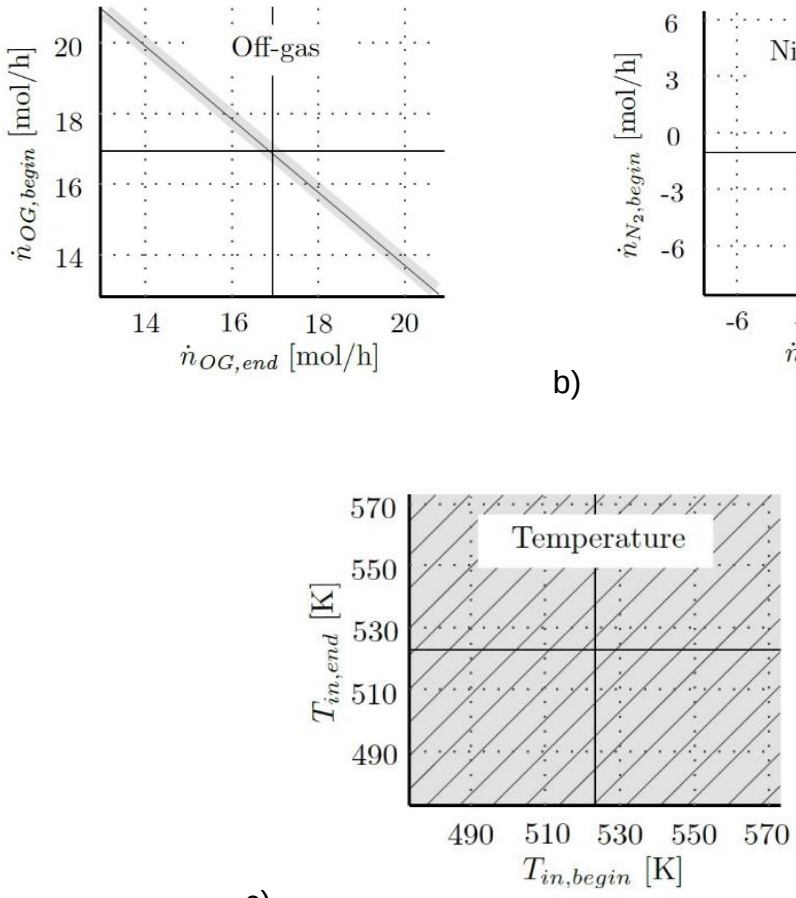


Figure 33: Parametric space applied for temporally profiled parameters; (a) Off-gas inlet flowrate, (b) Nitrogen-dilution, (c) Inlet temperature

The resulting, and generally possible, temporal profiling strategies are depicted in Figure 35 for an operating variable X which can be temporally profiled during the reaction-adsorption step from an X_{begin} value at t_{begin} to an X_{end} value at t_{end}, which is effectively t_{reac}. Mathematically, sigmoid curves can be expressed as follows:

$$sig(t) = (1 + e^{-t})^{-1} \qquad \textit{(Eq. 81)}$$

Thus, for an operating variable X, the temporal profile according to sigmoid curves is a function of the following parameters:

$$X(t) = f\,(X_{start}, X_{end}, S_{slope}, S_{duration}) \qquad \textit{(Eq. 82)}$$

where S_{slope} is the slope of the sigmoid curve and $S_{duration}$ is the reaction time within which the variable X is profiled (see Figure 34).

The temporal profiling strategies adopted here were implemented on the process under the cyclic steady state conditions. The relative ratio between cycle-step durations t_{ratio} was used as a performance indicator to evaluate the obtained results and is called cycle time ratio:

$$t_{ratio} = {}^{t_{reac}}/_{(t_{desorp} + t_{cool})} \qquad \textit{(Eq. 83)}$$

Higher cycle time ratios mean improved performance and vice versa. The obtained results are shown in Figure 36. The value span of the temporally profiled variables was normalised by the corresponding benchmark values given in Table 10.

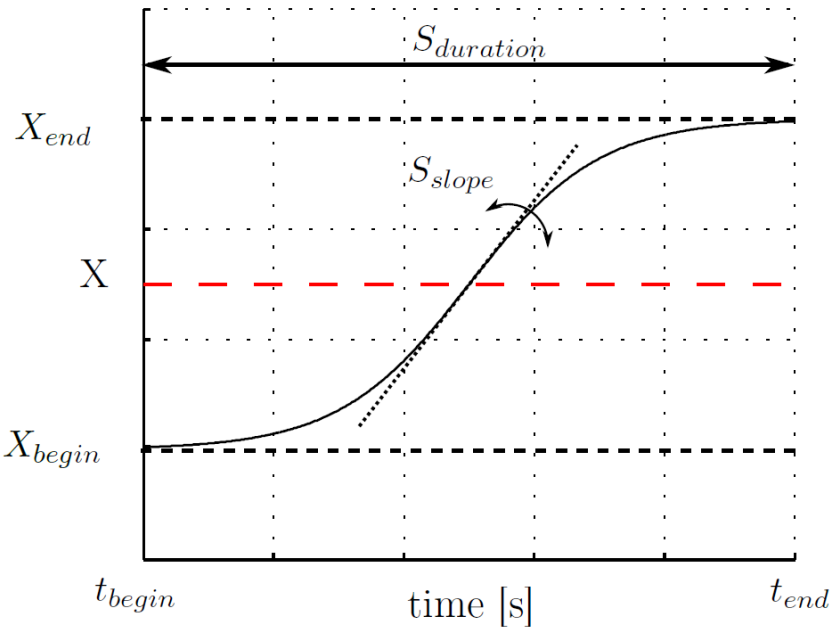


Figure 34: Sigmoid curve used for the temporal profiling of an operating parameter X

As it can be seen in Figure 36, when dynamically profiling of all considered variables, process improvement could be attained. However, inlet temperature profiling showed the best improvement with a clear tendency from low to high values.

Specifically speaking, by profiling the inlet temperature from about 498 to 548 K, an improvement of about 16.7 % can be obtained. Further increase in the upper profiled value of the inlet temperature causes the conversion to drop below its specification; 99.5 % (determined by environmental regulations) during the reaction-adsorption stage. The improvements by profiling the other two variables were marginal.

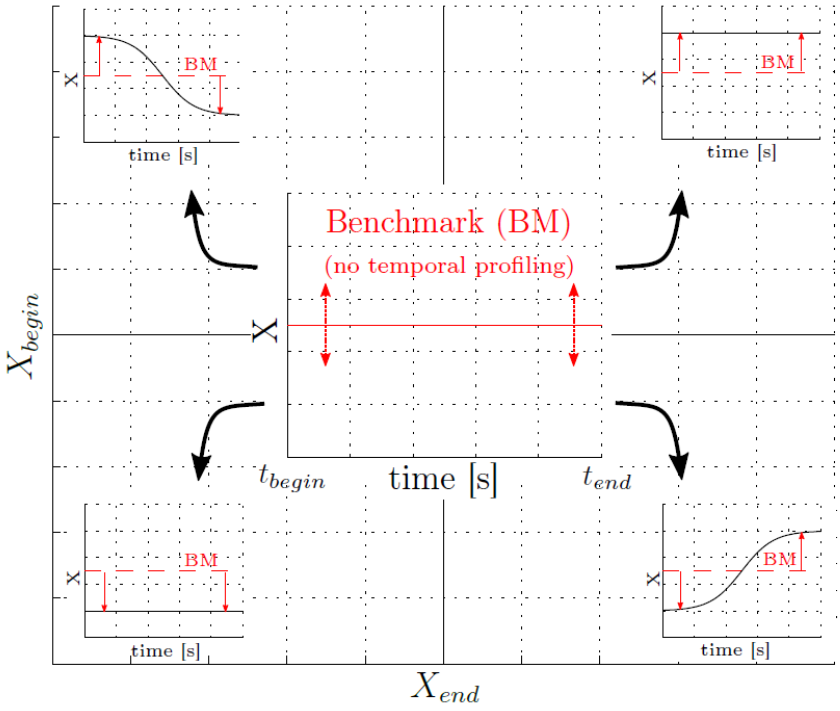


Figure 35: Possible temporal profiling strategies for an operating parameter X

This improvement was mainly caused by the visibly improved exploitation of the adsorbent pellets compared to the benchmark case, especially for the middle third of the fixed bed length. In spite of the fact that the adsorbed water was partly desorbed due to higher temperatures at the start front of the simulated adsorptive reactor, the overall exploitation of the adsorbent within the fixed bed was enhanced by about 50%. Consequently, the reaction-adsorption step could be prolonged by about 50%. The increased exploitation of the adsorbent, on the other hand, increases the desorption and cooling times required for the regeneration step of the cycle rendering the overall improvement relatively low.

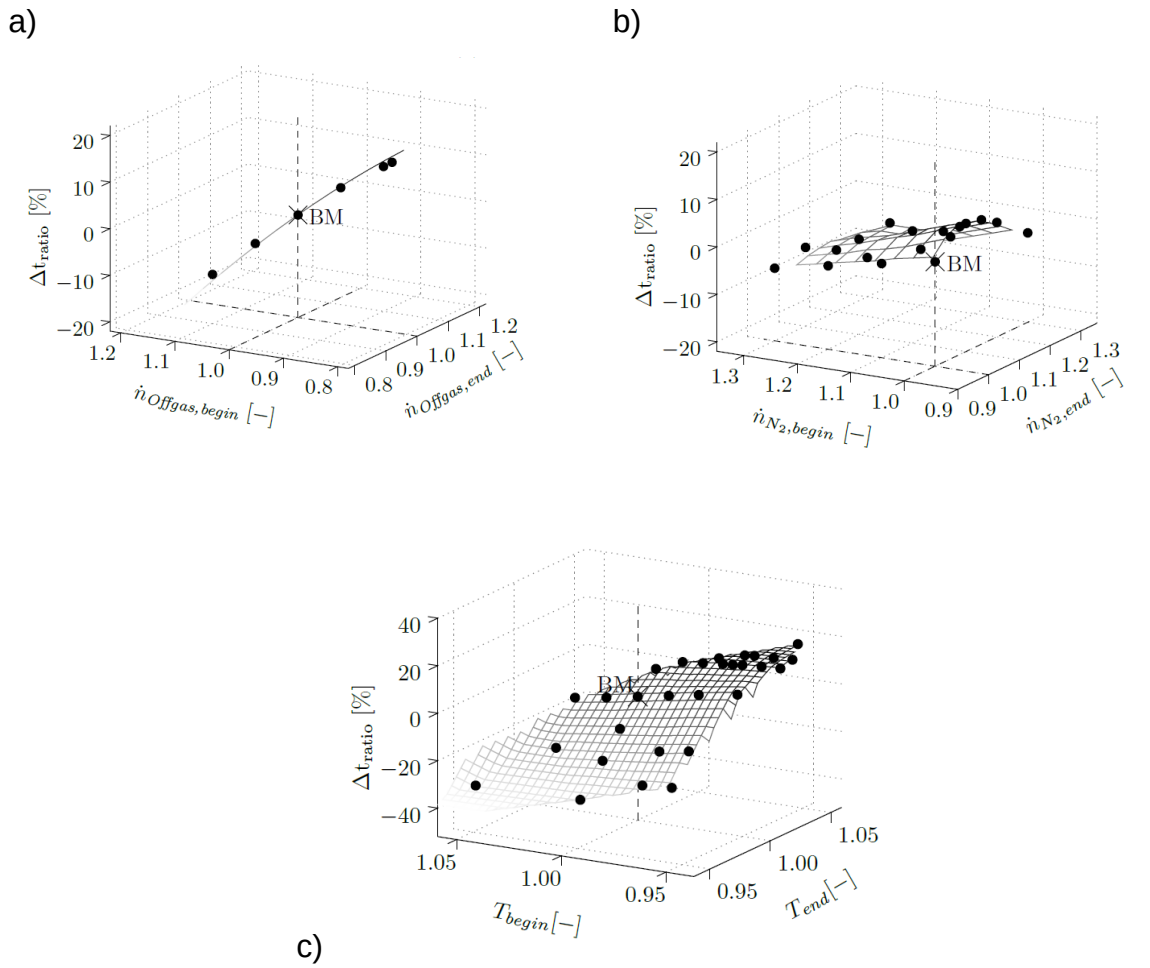


Figure 36: Obtained results of temporal profiling of: (a) the off-gas inlet flowrate, (b) the nitrogen-dilution, (c) the inlet temperature

Considering the applicability and realisation of the inlet temperature profiling, the main drawback would be an increased process operation complexity and increased energy demand, about 45% since an additional heat exchanger is necessary. The resulting additional costs would not be compensated for by the 16.7 % process improvement reported here. Therefore, as an operational technique for enhanced adsorptive reactor performance, the temporal profiling of operating parameters seems not to be feasible.

4.6. Distributed feed

Since it is strictly recommended by Elsner [23] to maintain the stoichiometric ratio of the reactants for the adsorptive Claus reactor to avoid possible chromatographic reactant slip, the distributed feed strategy to enhance the adsorptive reactor performance was addressed for Deacon reaction.

Several feed strategies, taken from [118, 119], have been elaborated. Generally, there are three different design parameters available, the number of feeds N, the percentage distribution of the feeds q_n, and the position of each feed p_n. Since the optimisation study considering these three parameters altogether requires much high computational effort, the number of feeds was restricted to only three and several parametric studies have been conducted to find out the best structure of distributed-feed adsorptive reactor. Figure 37 show an example of three distributed feeds of an adsorptive reactor with uniformly distributed adsorptive and reactive functionalities.

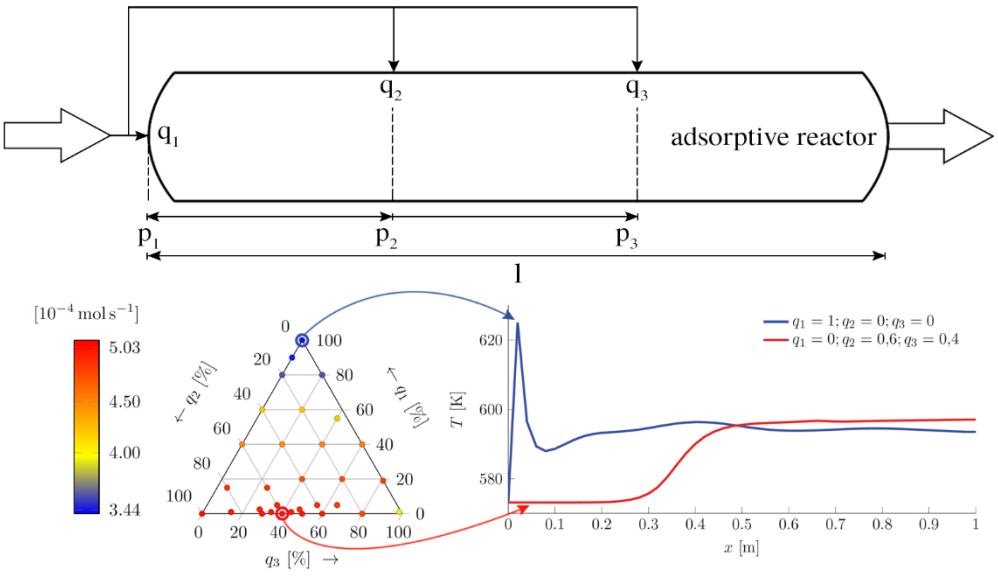


Figure 37: An adsorptive reactor with distributed feeds (q_1, q_2, and q_3) at different positions (p_1, p_2, and p_3)

The parametric studies considered the individual distribution of the reactants, O_2 and HCl, as well as the overall feed allowing for unequal distributed amounts and for irregular cross feeding points over the whole reactor length.

The most promising results were obtained for the case where the oxygen was unequally distributed over the reactor length. This, however, led to an improvement, over the benchmark case with uniform functionality distribution and no distributed feed, of up to 48% by which only a slight simplification of the downstream processing can be attained.

4.7. Novel designs considering the regeneration process

Although adsorptive reactors offer effective manipulation of the concentration and temperature profiles and thus achieve enhanced performance over their conventional counterparts, the operation of these reactors is inherently periodic (unsteady state) which complicates the design and operation process.

In order to capture, comprehend and capitalise upon the rich dynamic texture of adsorptive reactors, it is necessary to employ cyclic steady state algorithms describing the entire reaction-adsorption/desorption cycle. Furthermore, this cyclic steady state should be stable for efficient design and operation of adsorptive reactors. In this section, the cyclic steady state of previously proposed (in section 4.1) novel adsorptive reactor designs has been calculated and then optimised to give maximum space-time yields.

Based on the results presented in CHAPTER 3 and in the preceding sections of this chapter, macrostructuring of the adsorptive and reactive functionalities enhanced by a suitable spatial temperature profiling offers the most promising ameliorated performance of adsorptive reactors in comparison to other studied design and operational degrees of freedom. Nonetheless, to establish the feasibility and verify the overall optimal operation of the novel adsorptive reactor designs, and thus make them commercially viable and benchmark them against conventional technology, not only the adsorption kinetics and capacity must match specific required profiles, but also the adsorbent-regeneration procedure. In order to guarantee quasi-continuous operation of these proposed designs, the time needed for regeneration should be markedly lower than the time taken till complete adsorption. Amongst the typical procedures of regenration known in adsorption technology, flushing with an inert (nitrogen; N_2) was

chosen for this study. Thus, a regeneration step, including desorption and cooling (if necessary) stages, was considered in the whole operational cycle and the cyclic steady state was then calculated using direct determination approach. Finally, this cyclic steady state was optimised to give maximum space time yield (STY) of the adsorptive reactor under consideration keeping a minimum conversion of $X_{\min}$; $X_{\min}$ was like before: 99.5% in case of adsorptive Claus reactor and 96% for Deacon reaction.

A two-step cycle has been designed for the periodic operation of the adsorptive Claus and Deacon reactors. The two steps considered were as follows (Figure 38 shows schematically the designed cycle):

<u>Cycle step 1:</u> Reaction and adsorption under isothermal or adiabatic operating conditions

In this step, the mixture of reactants carried by nitrogen gas was fed into the adsorptive reactor at a temperature (T_f) and feed velocity (u_f). The step was terminated when the calculated conversion of hydrogen sulphide reached a value below 99.5 in case of Claus reaction and when the calculated conversion of hydrogen chloride dropped below 96% for Deacon reaction. The duration of this step is labelled as reaction-adsorption time (t_{reac}).

<u>Cycle step 2:</u> Desorption (and cooling) step with countercurrent boundary conditions

Here a switch in the boundary conditions was performed to simulate the countercurrent desorption, and the cooling in case of the adiabatic operation in the reactive sections of the simulated adsorptive reactor, achieved by introducing a nitogen gas stream at a regeneration velocity (u_{reg}). This step was terminated and a switch back to step 1 was activated when the average loading in the adsorption section reached or dropped below 10^{-5} kmol/m^3 and the average solid temperature reached or dropped below 250°C. The duration of step 2 was denoted as regeneration time (t_{reg}).

The values used in the simulations for feed operating parameters used in the cyclic steady state simulations are the same as those given in Table 3 in CHAPTER 3 for Claus as well as for Deacon reactions. However, the values mentioned in Table 3 of

the feed velocity u_f and regeneration velocity u_{reg} in case of the isothermal and adiabatic operations, and the regeneration temperature T_{reg} for the adiabatic operation were taken for the reference case, since the foresaid parameters were chosen to be design variables in the performed optimising simulation of the cyclic steady state, as it will be explained in the following subsection.

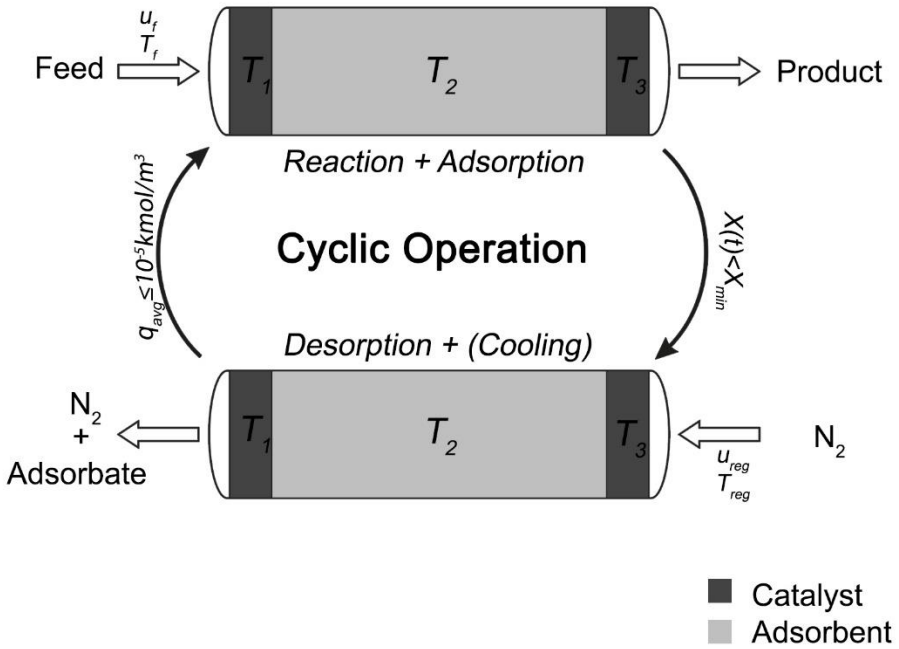


Figure 38: The two-step designed cycle for the periodic operation of the studied adsorptive Claus and Deacon reactors

4.7.1. Optimisation problem description of the cyclic steady state

The novel proposed adsorptive reactor designs introduced in Section 4.1.2. of this chapter were a result of optimised functionality structure and operating temperature during the reaction-adsorption cycle step. Yet, considering the regeneration step in the entire cyclic operation is absolutely fundamental to verify the overall feasibility of those designs. Therefore, the cyclic operational optimality of those designs was sought by setting an optimisation problem addressing the role of the regeneration step.

Without any exception to other transient and cyclic operations, the space time yield (STY), which is characterised as the molar production rate per unit volume of the reactor, is an appropriate key performance indicator for adsorptive reactor operation [88]. Hence, the objective function was to maximise the space time yield of the simulated adsorptive reactor by finding the optimal feed velocity in cycle step 1 and the optimal regeneration velocity in cycle step 2 of the designed cycle. Additionally, in case of the central isothermal design, the regeneration feed temperature was to be chosen by the optimiser. The upper and lower values of the optimising variables were set according to technical or safety considerations. For instance, the upper limit for velocity was calculated according to the fluidisation point of the bed. The upper limit of the regeneration temperature, on the other hand, was determined based on safety issues and to avoid damaging the zeolite particles, the lower limit was ascertained considering technical limitations by super-cooling the fixed bed. Mathematically, the optimisation problem is expressed as follows:

Objective function:

$$\max. STY = \frac{\int_{t_0}^{t_{\mathrm{reac}}} u|_{x=L,t} \times C_{\mathrm{DP}}|_{x=L,t} \times \mathrm{d}t}{L \times (t_{\mathrm{reac}} + t_{\mathrm{reg}})} \qquad \textit{(Eq. 84)}$$

s.t.:

Simulation equations described in Section 2.2.

$0.01 \leq u_{\mathrm{f}} \leq 1.5\ m/s$

$0.01 \leq u_{\mathrm{reg}} \leq 1.5\ m/s$

$\Delta P < 0.1\ bar$

$200\ ℃ \leq T_{\mathrm{reg}} \leq 400\ ℃\ (for\ adiabatic\ operation)$

Design variables:

$u_{\mathrm{f}}, u_{\mathrm{reg}}, and\ (T_{\mathrm{reg}}\ for\ adiabatic\ operation)$

4.7.2. Results and discussion

The obtained optimised values of the design operating variables are listed in Table 11. Figure 39 and Figure 40 show the dynamic and spatial development of the loading profiles over the whole cycle for the benchmark cases and sandwich designs in case of Claus and Deacon reaction schemes, respectively.

In comparison to the benchmark cases, the increased adsorbent's capacity and appropriately modified temperature profiles established by the proposed novel adsorptive reactor arrangements (as can be seen in Figure 39-a to 39-f and Figure 40-a to 40-f) resulted in maximised productivities in the first cycle step. Additionally, the early-desorption phenomenon already occurring during the reaction-adsorption cycle step at the entrance of the adsorptive reactor (Figure 39-c and Figure 40-c) observed for the adiabatic benchmark cases, caused by the hot-spot formation at that location (as shown in Figure 39-e and Figure 40-e), can be avoided by spatially segregating the adsorptive and reactive functionalities as suggested by the novel sandwich designs. This spatial segregation of involved functionalities reflects the fact of existence of three distinct regions/zones in the adsorptive fixed bed reactor.

In order to comprehensively analyse these zones and get insights into the behaviour and optimisation of the process, the approach to equilibrium parameter (χ) defined by Eq. 85 represents a direct measure to indicate the truly utilisation of the functionalities at any location within the fixed bed and describes the potential driving force for a reversible reaction to take place.

$$\chi = \frac{\kappa}{\prod_i c_i^{\nu_i}} \qquad \textit{(Eq. 85)}$$

The reaction is at thermodynamic equilibrium at $\chi = 1$. When $\chi < 1$, the backward reaction takes place, while the forward reaction occurs if $\chi > 1$.

Figure 41 shows the variation of the approach to equilibrium parameter in space and time for the isothermal and adiabatic adsorptive Claus reactor.

The first zone (the first tenth of the fixed bed) represents the initial stage of the reaction where the forward reaction predominates. Here, the value of χ is always greater than one. This suggests that the forward reaction is favoured and progresses at a faster

rate than the reverse reaction. In this kinetically controlled zone, the concentration profiles of reactants do not exhibit significant changes, indicating that their consumption is limited. However, the concentration of the adsorbate component gradually increases until it reaches a state of saturation. It is important to note that the product concentration is not yet high enough to influence the overall reaction rate significantly. Surprisingly, the adsorbent material employed in this zone has minimal positive influence on the reactor performance. This suggests the non-efficiency of the adsorptive functionality in this region.

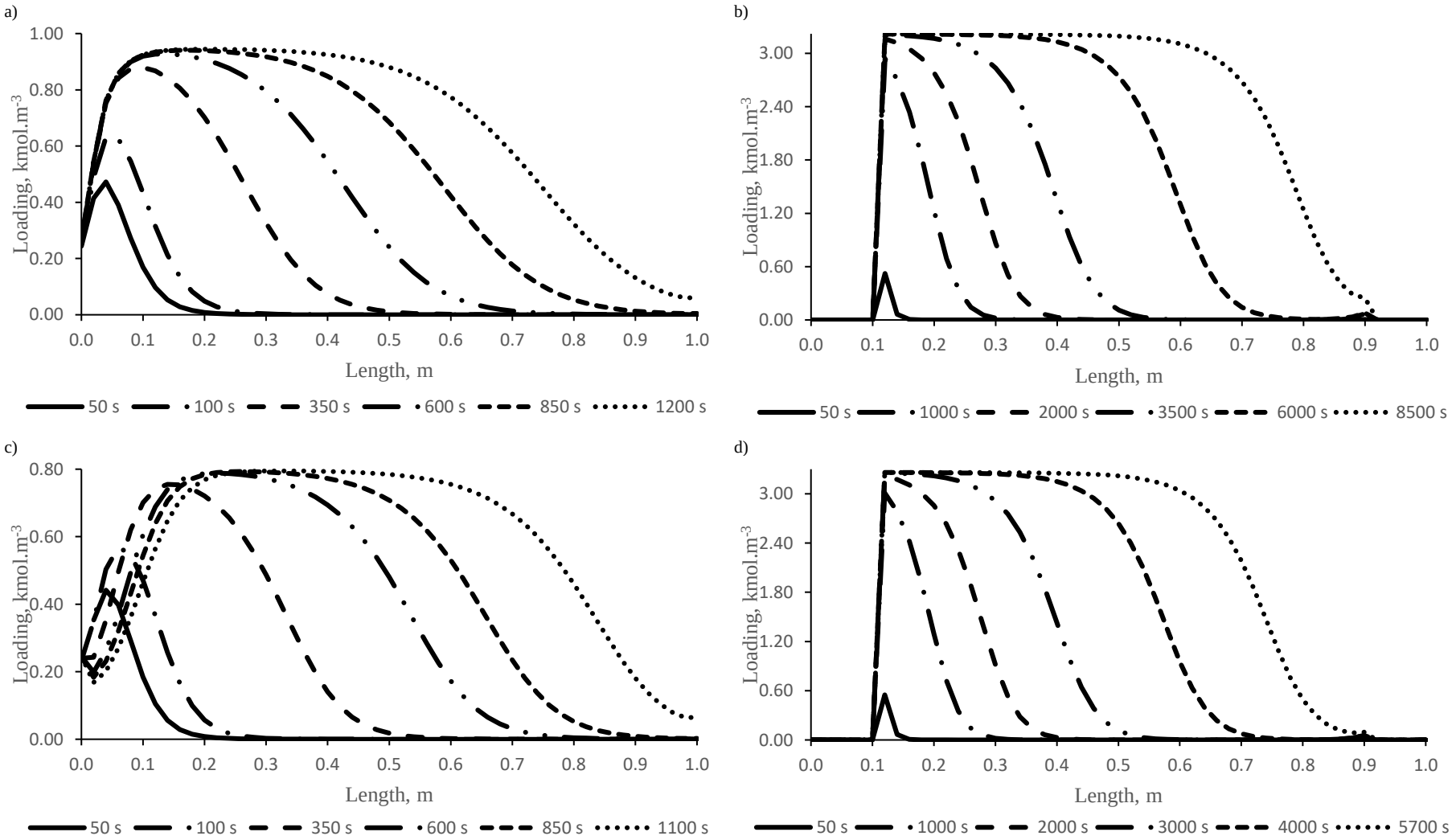
Table 11: Optimisation results of the periodically operated adsorptive reactor

Optimised design variables	**Claus reaction**		**Deacon reaction**	
	Isothermal	Adiabatic	Isothermal	Adiabatic
u_f **[m.s^{-1}]**	0.2	0.21	0.18	0.16
u_{reg} **[m.s^{-1}]**	1.2	1.2	1.2	1.2
T_{reg} **[K]**	-	523	-	523
ΔP, reaction-adsorption **[Pa]**	611.8	651.3	551.0	502.0
ΔP, regeneration **[Pa]**	3694.0	3659.4	3711.1	3657.3

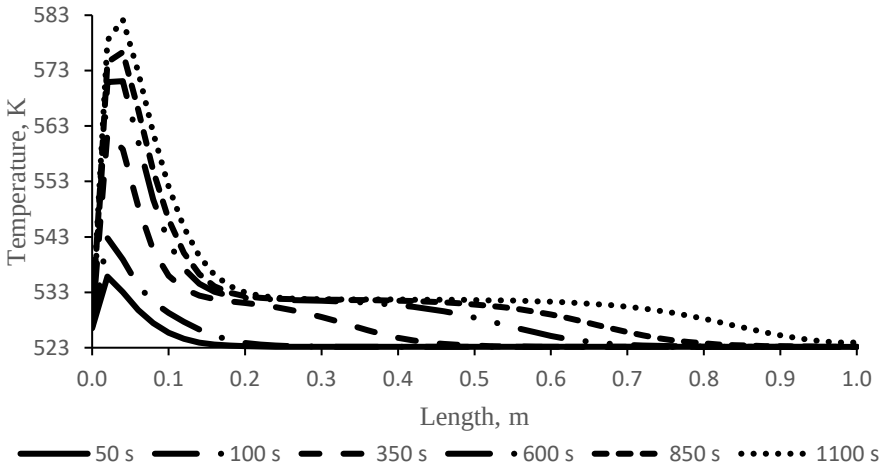
Moving into zone 2 (within nearly the next eight-tenths of the reactor), the behaviour of the system undergoes some notable changes. Initially, χ is still greater than 1, suggesting a kinetically controlled regime. However, as the adsorbent becomes saturated, χ eventually reaches a value of 1, indicating the establishment of equilibrium conditions. In this zone, the concentrations of reactants are considerably lower compared to zone 1, as their primary reaction occurs in the previous zone. Conversely, the product concentration in zone 2 is higher, resulting in a slower overall reaction rate. To sustain the forward reaction in this region, the adsorptive functionality of the adsorbent material becomes crucial. It facilitates the removal of the adsorbate, driving the reaction towards completion. Thus, the presence of an efficient adsorbent is vital for maintaining reaction progress and enhancing reactor performance in zone 2, whereas the existence of any reactive functionality in this region would be limitedly beneficial.

Zone 3 (the last one tenth of the reactor) shares similarities with zone 2 in terms of χ values, but it possesses distinct characteristics that set it apart. Unlike zone 2, the adsorbent in this zone never reaches saturation, even when process constraints are breached. The loading front, indicating the extent of adsorbate penetration into the adsorbent bed, does not advance beyond this segment before the breach occurs. Consequently, any adsorbate present in zone 3 originates within this zone and does not significantly impact the overall process. Therefore, zone 3 remains kinetically controlled throughout the entire process cycle. The same tendency was reported by Lawrence et al. [24].

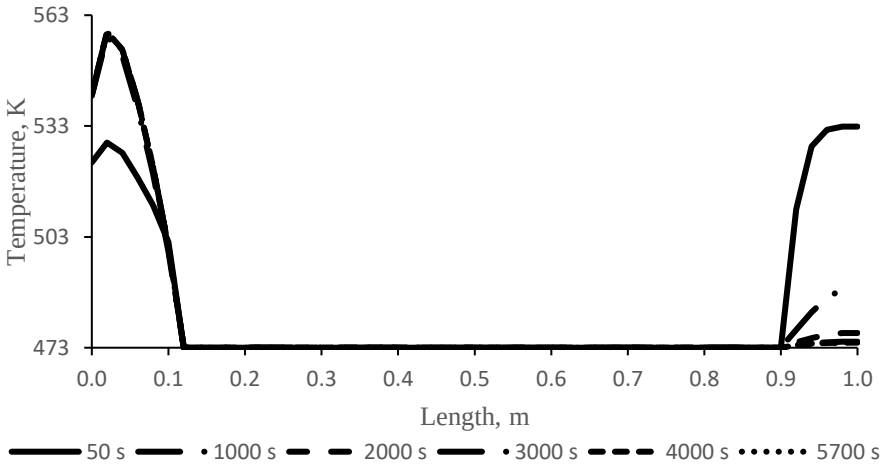
In summary, the three distinct zones in adsorptive reaction engineering provide valuable insights into reactor performance. Zone 1 represents the initial kinetically controlled stage, where the forward reaction dominates. Zone 2 transitions from kinetic control to equilibrium conditions as the adsorbent becomes saturated, necessitating its adsorptive functionality. Lastly, zone 3 remains kinetically controlled, with the adsorbent never reaching saturation, even during process constraints breaches. Understanding and optimising the behaviour of these zones have aided in designing efficient spatially segregated adsorptive reaction processes, ultimately leading to a substantially improved reactor performance.


a)
Loading, kmol.m-3
Length, m
50 s
100 s
350 s
600 s
850 s
1200 s
b)
Loading, kmol.m-3
Length, m
50 s
1000 s
2000 s
3500 s
6000 s
8500 s
c)
Loading, kmol.m-3
Length, m
50 s
100 s
350 s
600 s
850 s
1100 s
d)
Loading, kmol.m-3
Length, m
50 s
1000 s
2000 s
3000 s
4000 s
5700 s

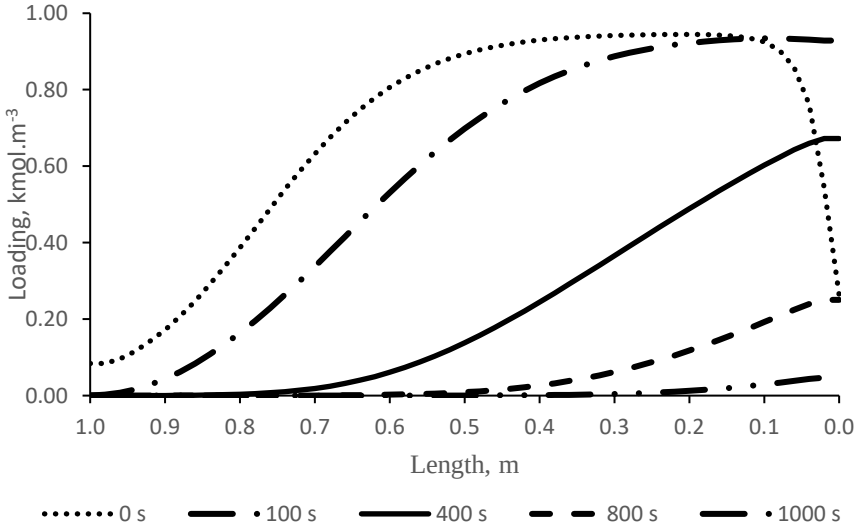
e)


583
573
563
553
543
533
523
Temperature, K
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Length, m
50 s
100 s
350 s
600 s
850 s
1100 s

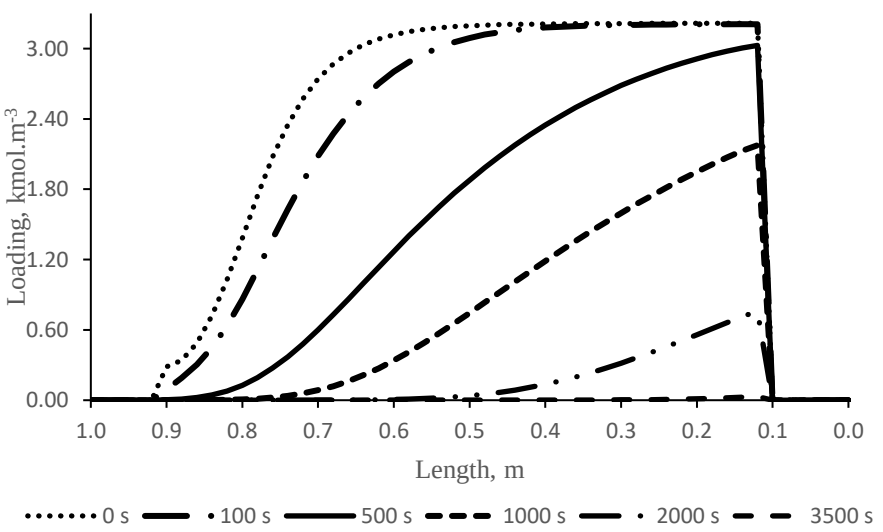
f)


563
533
503
473
Temperature, K
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Length, m
50 s
1000 s
2000 s
3000 s
4000 s
5700 s

g)


1.00
0.80
0.60
0.40
0.20
0.00
Loading, kmol.m-3
1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
Length, m
0 s
100 s
400 s
800 s
1000 s

h)


3.00
2.40
1.80
1.20
0.60
0.00
Loading, kmol.m-3
1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
Length, m
0 s
100 s
500 s
1000 s
2000 s
3500 s

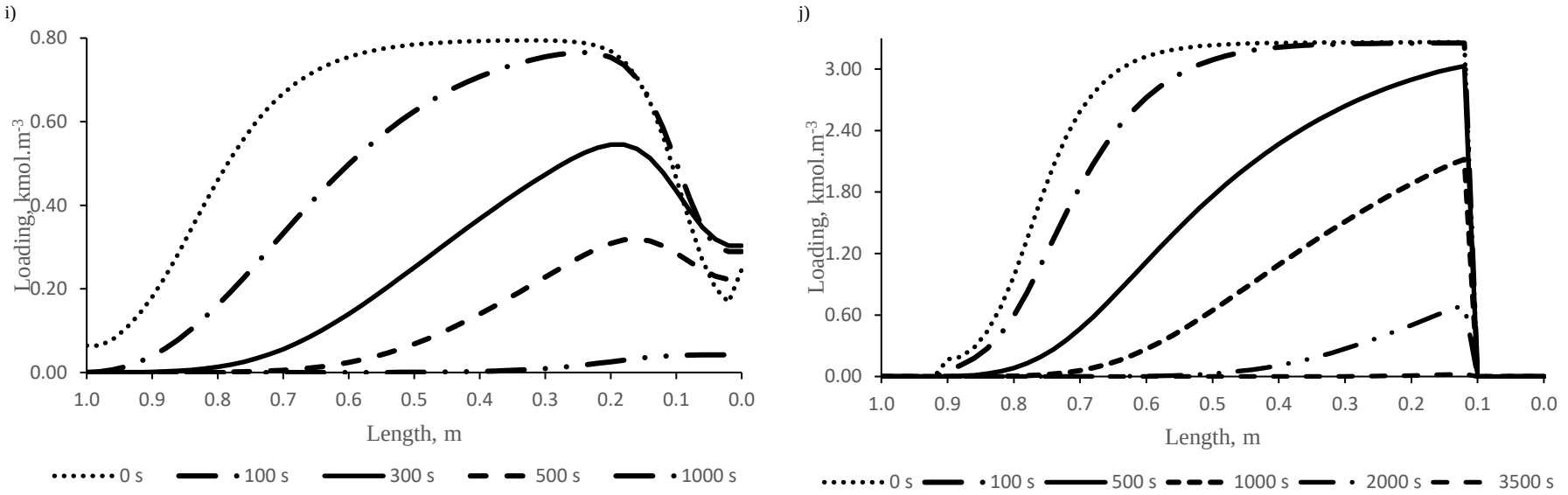
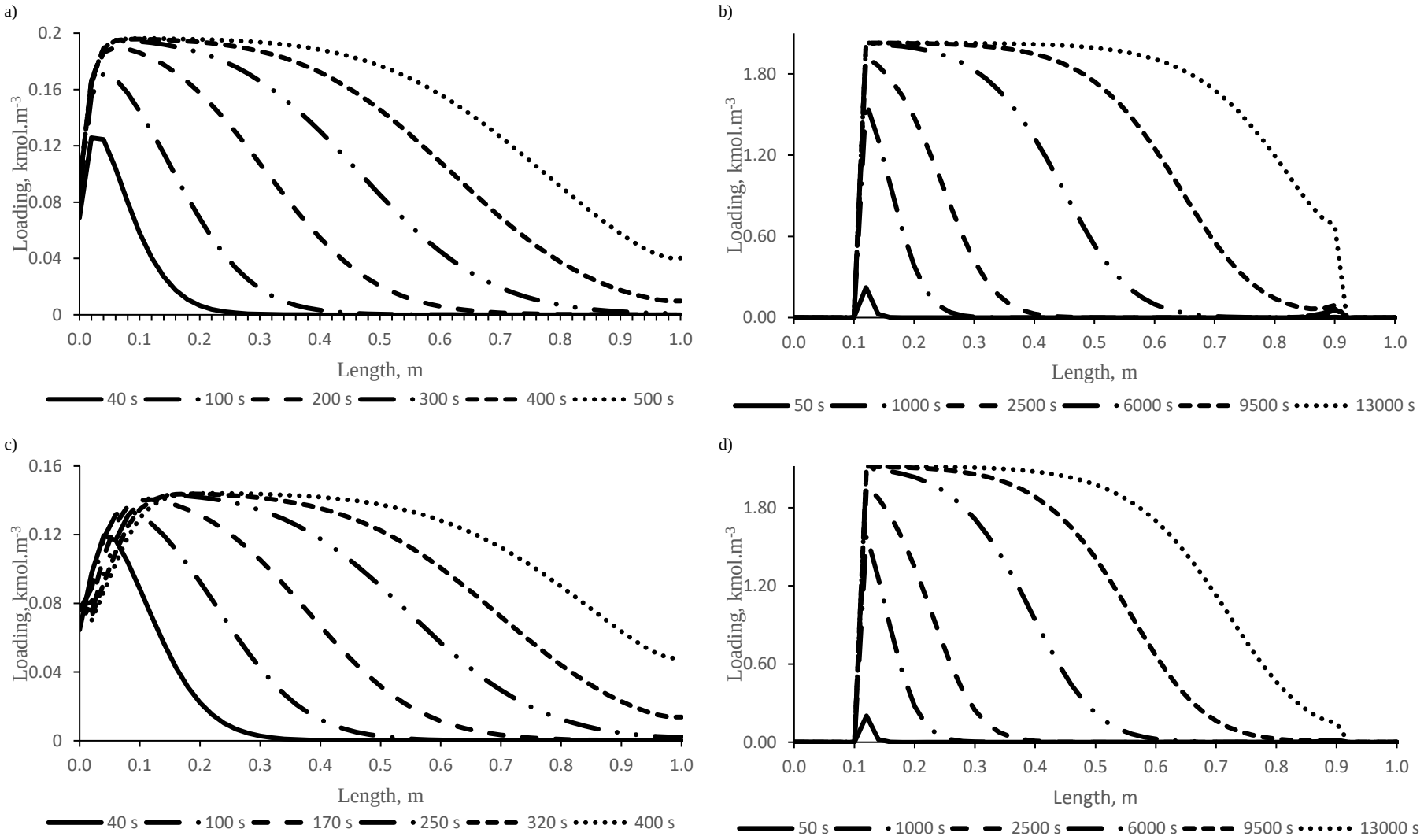
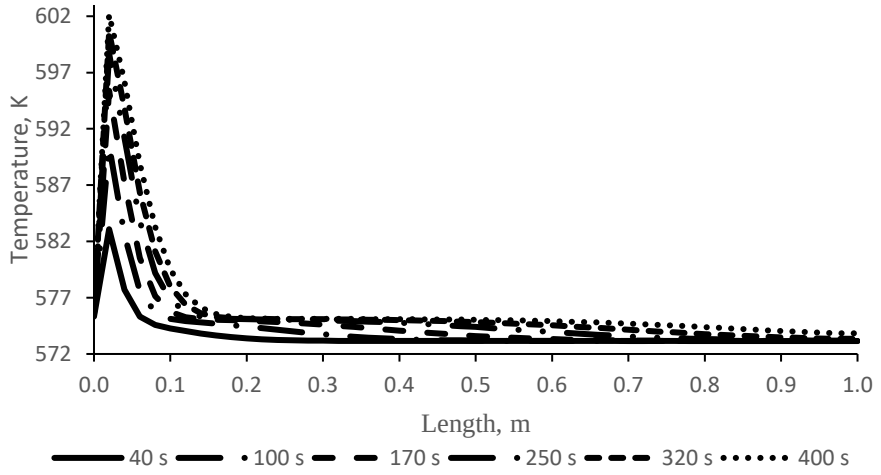


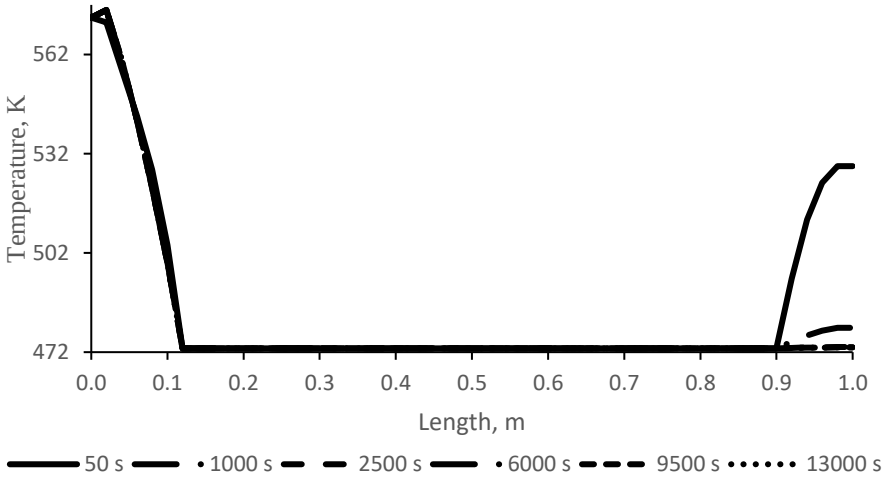
Figure 39: The dynamic and spatial development of the loading and solids temperature profiles for Claus reaction. Left (a, c, e, g, and i): benchmark cases, Right (b, d, f, h, and j): optimal adsorptive reactor arrangements. Cycle step 1 (reaction-adsorption): a to f, and Cycle step 2 (regeneration): g to j (the loading profile at t = 0 here represents the profile at the end of the first cycle step). Isothermal operation: a and b, Adiabatic operation: c to f.


a)
Loading, kmol.m-3
0.2
0.16
0.12
0.08
0.04
0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Length, m
40 s 100 s 200 s 300 s 400 s 500 s
b)
Loading, kmol.m-3
1.80
1.20
0.60
0.00
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Length, m
50 s 1000 s 2500 s 6000 s 9500 s 13000 s
c)
Loading, kmol.m-3
0.16
0.12
0.08
0.04
0
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Length, m
40 s 100 s 170 s 250 s 320 s 400 s
d)
Loading, kmol.m-3
1.80
1.20
0.60
0.00
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Length, m
50 s 1000 s 2500 s 6000 s 9500 s 13000 s

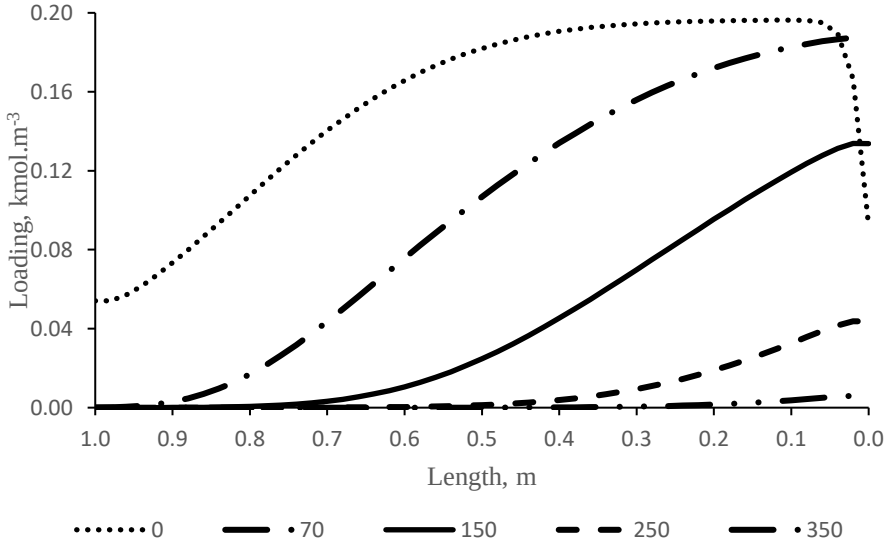
e)


602
597
592
587
582
577
572
Temperature, K
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Length, m
40 s
100 s
170 s
250 s
320 s
400 s

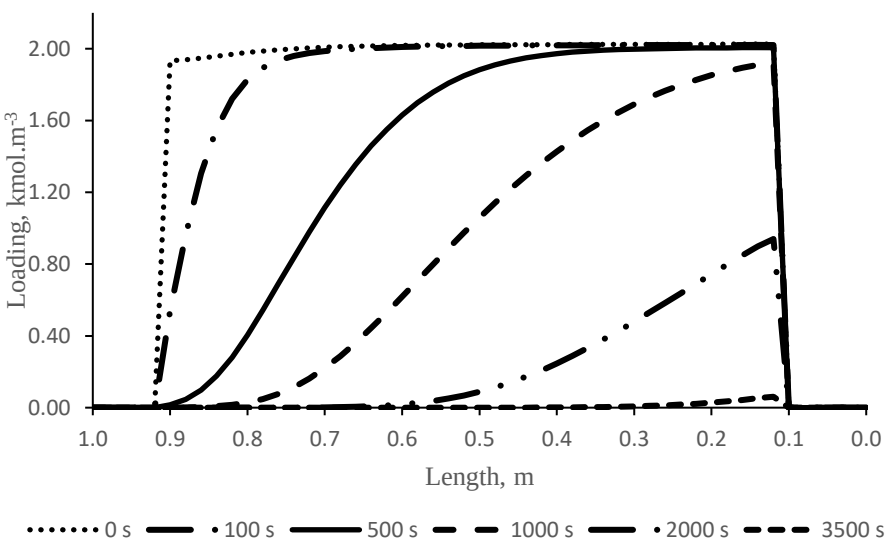
f)


562
532
502
472
Temperature, K
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Length, m
50 s
1000 s
2500 s
6000 s
9500 s
13000 s

g)


0.20
0.16
0.12
0.08
0.04
0.00
Loading, kmol.m-3
1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
Length, m
0
70
150
250
350

h)


2.00
1.60
1.20
0.80
0.40
0.00
Loading, kmol.m-3
1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0
Length, m
0 s
100 s
500 s
1000 s
2000 s
3500 s

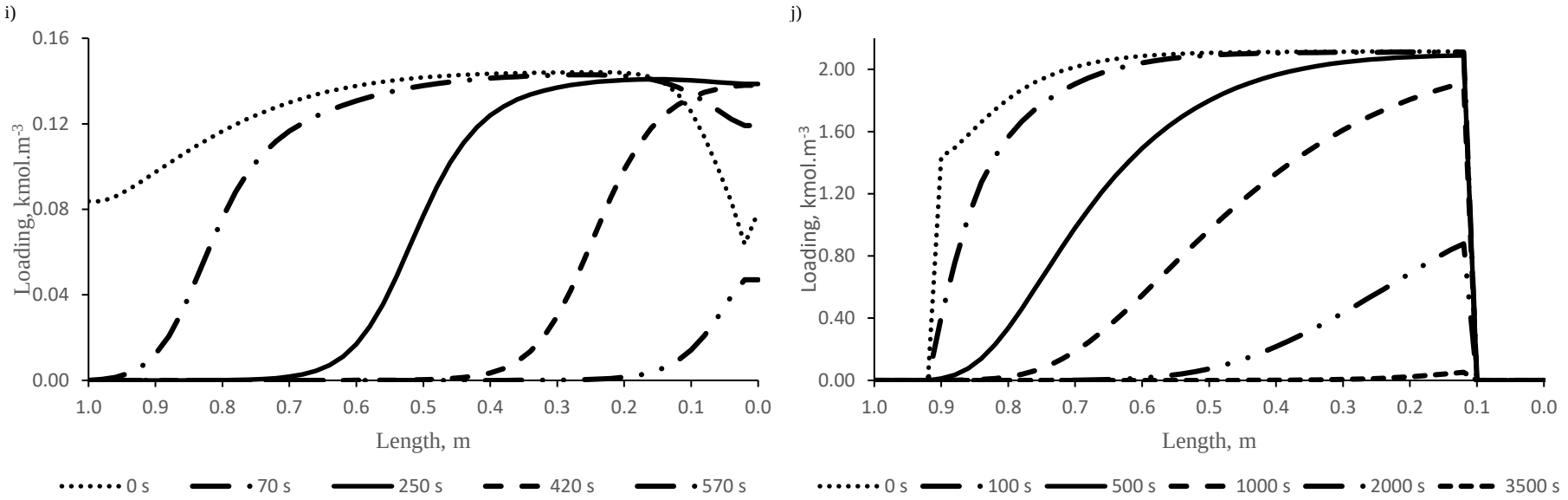
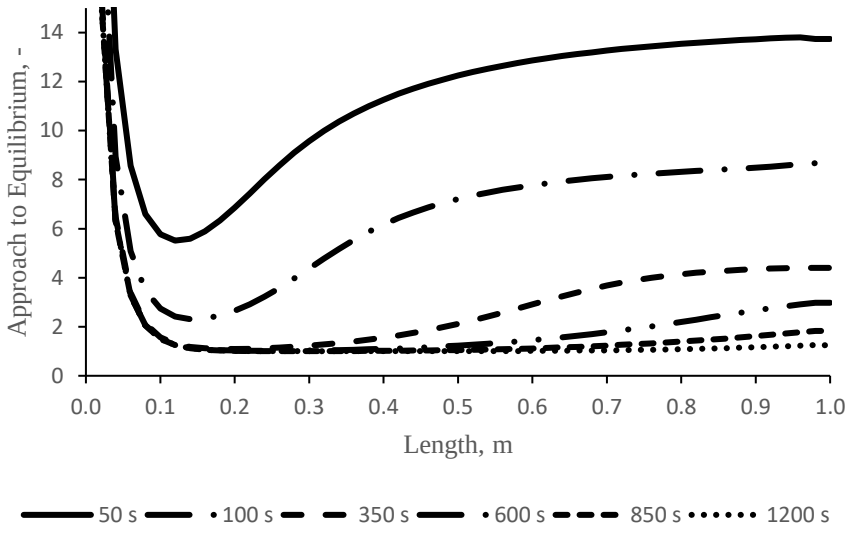


Figure 40: The dynamic and spatial development of the loading and solids temperature profiles for Deacon reaction. Left (a, c, e, g, and i): benchmark cases, Right (b, d, f, h, and j): optimal adsorptive reactor arrangements. Cycle step 1 (reaction-adsorption): a to f, and Cycle step 2 (regeneration): g to j (the loading profile at t = 0 here represents the profile at the end of the first cycle step). Isothermal operation: a and b, Adiabatic operation: c to f.

a)



b)

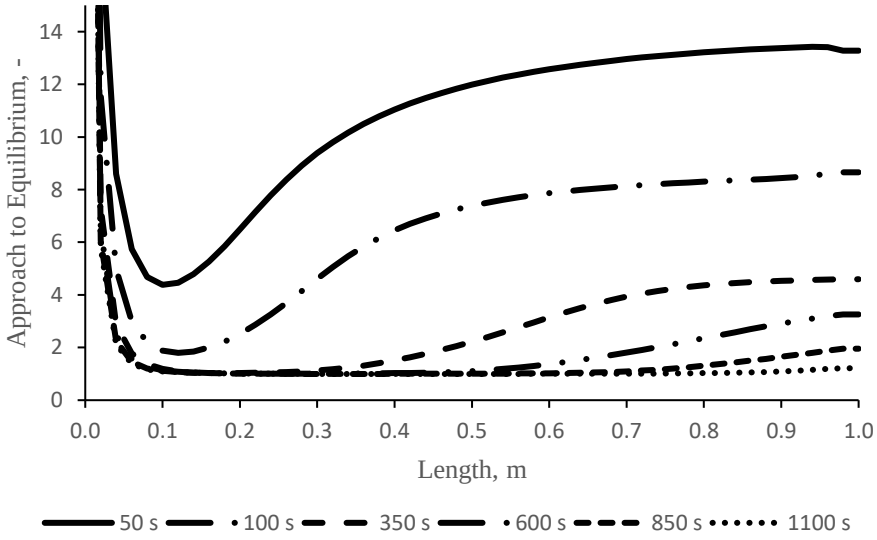


Figure 41: The dynamic and spatial profiles of the approach to equilibrium parameter for the benchmark case (the adsorbent volume fraction at any spatial location is 43%) of adsorptive Claus reactor during the first cycle step (reaction-adsorption). a): isothermal operation and b): adiabatic operation

The optimised value for the regeneration velocity was the same in all cases and it is three times the corresponding value for the benchmark case. This value guarantees much faster regeneration process maintaining at the same time the allowed pressure drop in the bed and lies well below the fluidisation value. Moreover, it assures avoiding the negatively affecting re-adsorption during the regeneration phase at the entrance of the adiabatically operated conventional adsorptive reactor (see Figure 39-i and Figure 40-i). In contrast, the temperature of the regeneration feed does not differ from its value for the benchmark case, where higher values will lead to a slow cooling of the adiabatic

reactive segments and thus longer regeneration time. In other words, the enhancement to faster desorption by higher regeneration feed temperature will not compensate the much longer cooling duration of the reactive sections in the central isothermal adsorptive reactor design.

The obtained values of the objective function under the optimised cyclic steady state conditions of the novel adsorptive reactor designs are starkly juxtaposed with those of the conventional adsorptive counterparts for the Claus reaction in Figure 42 and for the Deacon reaction in Figure 43. It is obvious that the multilevel isothermal sandwich design raises the space time yield by c. 700% for the Claus reaction and thirty-five-fold for Deacon reaction, while the central isothermal sandwich design magnifies the space time yield by about 650% for the Claus and by roughly a factor of 18.5 for the Deacon process. The superior improvement reported here is coming from the extra prolonged production time (i.e. the reaction-adsorption time) achieved by macrostructuring and temperature profiling and the much more efficient regeneration process (short regeneration time well below the production time) established by setting the optimal feed and regeneration velocities.

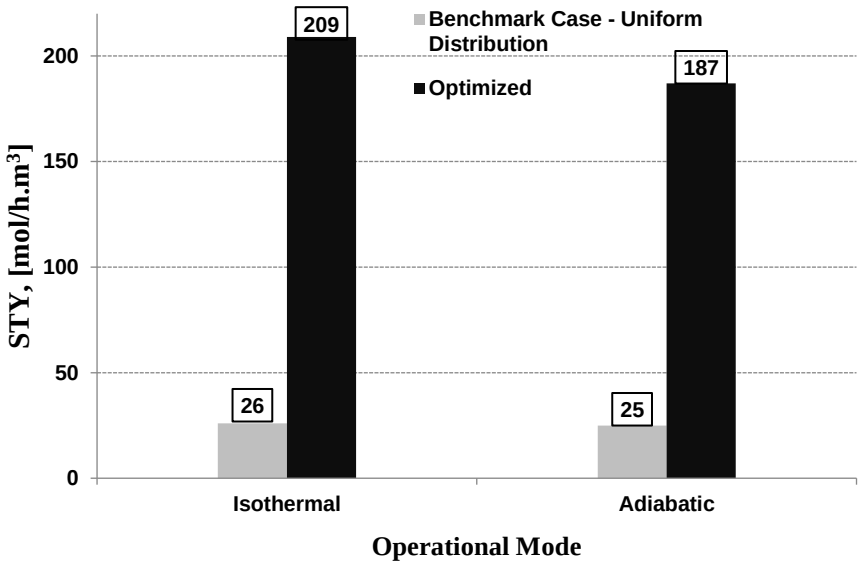


Figure 42: The space time yield under periodic steady state conditions obtained for the conventional and the novel adsorptive Claus reactor designs

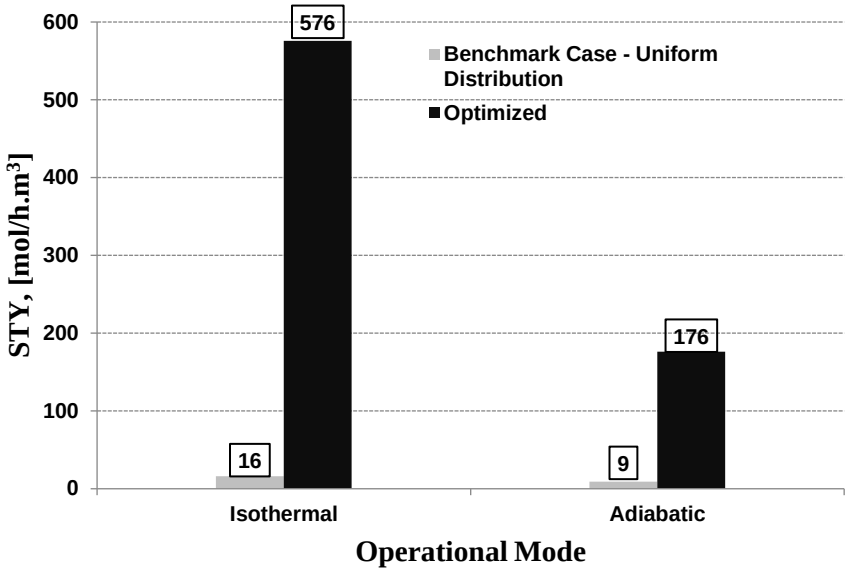


Figure 43: The space time yield under periodic steady state conditions obtained for the conventional and the novel adsorptive Deacon reactor designs

Since the adiabatic temperature increase of both studied reactions does not exceed 60 °C, the tremendously high-cost savings, resulting from the simplification of the downstream processing of the overall adsorptive Claus and Deacon process flowsheets, will compensate for the extra operational and investment costs required for carrying out necessary heating/cooling required for implementing the novel proposed designs.

CHAPTER 5: CONCLUSIONS AND FUTURE OUTLOOK

5.1. Summary and conclusions

Adsorptive reactors represent a promising genre of bifunctional reactors combining the highly compatible adsorptive and reactive functionalities. Although they offer multiple benefits to the overall process flowsheet, overcoming the low space time yields, which are an inherent drawback of their cyclic operation, and unwanted functionality interference remain a challenge.

Multi-dimensional and multiscale numerical simulation and dynamic optimisation studies were carried out using ACM software to elucidate the potential benefits of available design and operational alternatives, for instance, the temperature profiling, side-stream feeds, temporal profiling of operating parameters, and regenerative heat storage. Two test reaction systems, that are industrially relevant, have been chosen in this work – the technically relevant Claus and the Deacon reactions, the former is for recovering sulphur from acid gas and the latter reaction is for the plant-internal recycling of chlorine through the oxidation of hydrogen chloride.

Preliminary simulations highlighted the effectiveness of a simple macrostructuring approach involving a catalyst-adsorbent-catalyst sandwich configuration. This macrostructuring technique significantly prolonged the duration of the adsorptive reaction phase compared to a uniform distribution of catalyst and adsorbent along the entire reactor. Further analysis revealed that a volumetric distribution comprising a central section containing 80% adsorbent enclosed by catalyst end-zones of 10% each proved beneficial for both reactions under investigation.

The localised segregation of functionalities facilitated the implementation of temperature profiling without encountering contradictions between the functionalities. Detailed simulation results showcased the impact of this approach, with catalytic regions exhibiting high temperatures and the central adsorbent region maintaining a

lower temperature of 200°C. As a result, cycle times for the Claus and Deacon reactions were extended by factors ranging from six to thirty compared to the base cases. Notably, even the use of simpler adiabatic reaction zones failed to diminish these gains significantly.

Microstructuring of functionalities at the pellet level, however, yielded only marginal improvements, indicating that mass transport resistances were not decisive factors in the system. The distribution of functionalities across the reactor radius proved to be significant mainly in its ability to suppress the inhibitory effect of water vapour on the catalytic kinetics of the Deacon reaction.

Regarding heat integration strategies, macroscale recuperative heat integration between two parallel reactors operating in the adsorptive and desorptive phases outperformed the regenerative strategy employing phase change materials (PCM). The inflexibility of the PCM-based approach likely contributed to the superior performance of the recuperative heat integration.

Moreover, dynamic profiling the reactor inlet concentrations and temperatures throughout the reaction cycle demonstrated limited benefits for the former case and modest advantages for the latter. However, the technique was deemed complicated and costly, reducing its practical viability.

The supply of oxygen-enriched sidestreams along the adsorptive Deacon reactor only provided a minor boost to the reaction kinetics, while a staged feed supply was found to be inappropriate for the fixed stoichiometry of the Claus reaction.

Thus, the potential of temperature profiling, in particular, to expedite the macrostructuring of adsorptive reactors for performance optimisation was found to give the best promising design of integrated adsorptive and reactive functionalities.

Accordingly, novel adsorptive reactor designs have been proposed for the isothermal and adiabatic operational modes: the multilevel isothermal and central isothermal sandwich designs, respectively (see Figure 44).

The feasibility of these novel adsorptive Claus and Deacon reactor designs (the multilevel isothermal and central isothermal sandwich designs) were verified by further numerical simulations and dynamic optimisations. The cyclic steady state was first calculated using the direct substitution approach based on the methods of line in Aspen

Custom Modeler and then optimised to demonstrate the overall optimality of the proposed adsorptive reactor designs.

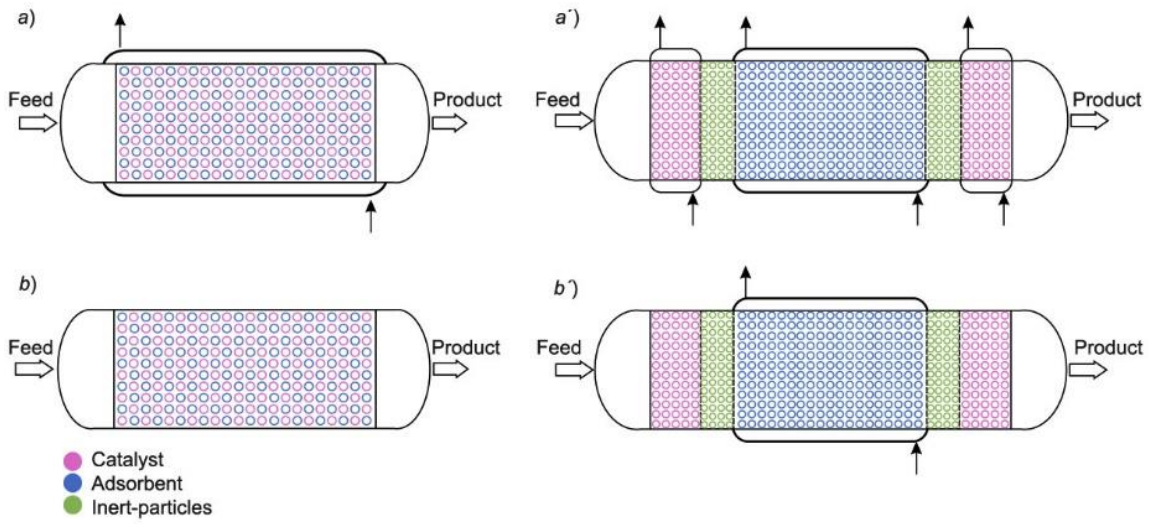


Figure 44: Structural and operational optimality of adsorptive reactors; a) homogeneous isothermal adsorptive, a´) structured multilevel-isothermal adsorptive reactor, b) homogeneous non-isothermal adsorptive, and b´) structured central isothermal adsorptive reactor

The results obtained revealed unambiguously that an improvement potential of up to multi-fold level could be attained under the optimised cyclic steady state conditions. This additional improvement resulted from the reduction of the regeneration time well below the reaction-adsorption time, which means, in turn, more space-time yield.

Additionally, the simulations showed that the flow of the eluent, rather than its temperature, emerged as the decisive parameter for expediting adsorbent regeneration, and thus, the efficient regeneration process proposed here eliminates possible problems associated with other reactor configurations, such as fluidised reactors. The Chi parameter, characterising the approach to chemical equilibrium, proved to be a valuable tool for analysing the local reactor behaviour in these simulations.

Overall, these comprehensive dynamic optimisation studies using extensive numerical simulations shed light on the potential benefits of various design and operational strategies in adsorptive reaction engineering. The findings emphasise the significance of functionality macrostructuring, temperature profiling, and eluent flow in optimising reactor performance and maximising space time yields.

The two test reaction systems chosen here exhibit both similarities, such as the adsorptive removal of water vapour and their moderate exothermicity, and significant differences, for instance the greater influence of reactant composition in the Claus process, which suggests the suitability of the proposed designs for other reactions lying between these two reactions. Further generalisation requires the consideration of more, completely distinct, reaction systems.

5.2. Future outlook

The results of this thesis incentivise more research to further develop advanced adsorbents and catalysts and conduct pilot-plant studies as the next step, following the currently available experimental studies and process simulations alike, to move with adsorptive reactors, as an up-and-coming technology, up to the next technology readiness levels.

As a next step, the experimental verification of the proposed designs should be implemented followed by a scale-up study of the adsorptive reactors. In the following subsections, preliminary experimental studies, which serve as a basis for future further experimental investigations, as well as some remarks on the scalability of adsorptive reactors are presented.

5.2.1. Preliminary experimental investigations

It should be noted that despite the significant improvement of process performance achieved which reflects the extremely attractiveness of adsorptive reactors as an up-and-coming technology, there are still technical challenges to be overcome and these should by no means be underestimated to successfully commercialise this concept. These challenges include sulphur condensation in case of the Claus reaction, and the as-yet unsolved availability of an appropriate adsorbent to resist the highly corrosive conditions of Deacon reaction system. For the latter reaction scheme, an absorptive stage instead of the high temperature adsorption between two reaction steps has been suggested and calculated to be taken as benchmark for an adsorptive Deacon reactor.

a) Adsorbent check-up for Deacon reaction

Experimentally, and specifically for Deacon reaction, the 3A zeolite applied in the simulations, along with an alternative commercially available adsorbent Zeolite Type AW-500 developed by UOP, which was claimed to withstand such extremely aggressive reaction medium, were exposed for longer periods to hydrogen chloride vapour at increased temperatures and to aqueous hydrochloric acid at room temperature. Disappointingly, the resilience of both adsorbents could not be ascertained as can be shown in Figure 45.

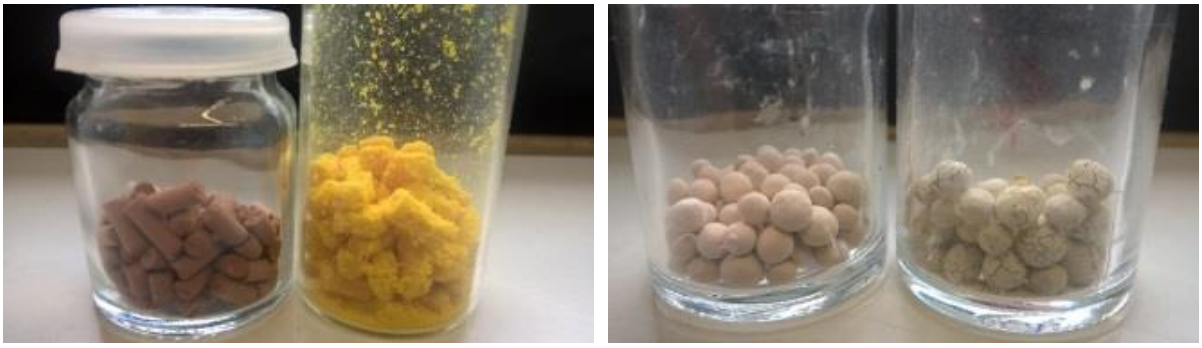


Figure 45: 3A Zeolite (right) and Zeolite AW-500 (left) before and after the exposure to HCl vapour and aq-HCl solution

In this case, an absorptive stage, using the traditional absorptive desiccant (concentrated sulphuric acid), instead of the high temperature zeolitic adsorption between two reaction steps thus is a more realistic option (as discussed in Section 3.2) and could be taken as benchmark for adsorptive operation of Deacon process. The intermediate removal of water vapour from the reaction medium enables one to enhance the conversion achieved well-above the 85% typical for conventional operation. This would enable a dramatic rationalisation of the downstream processing, since neither an electrolysis nor an “azeotropic distillation” would be required to recover unconverted hydrogen chloride, and a simple cryogenic separation of chlorine from oxygen would suffice.

b) Experimental set-up for adsorptive reactor concepts

A preliminary experimental work was commenced by rebuilding an already existing bench-scale plant due to the urgent renovation work in the laboratory space of the chair of chemical reaction engineering at TU Dortmund University in 2016/2017. This step

involved not just the reactor construction, but also encompassed measurement and control systems, pipework, the oil heaters needed to maintain reaction and regeneration temperatures and safety equipment. The experimental set-up for adsorptive Claus process (RAR-concept, i.e. reaction-adsorption-reaction) is depicted in Figure 46.

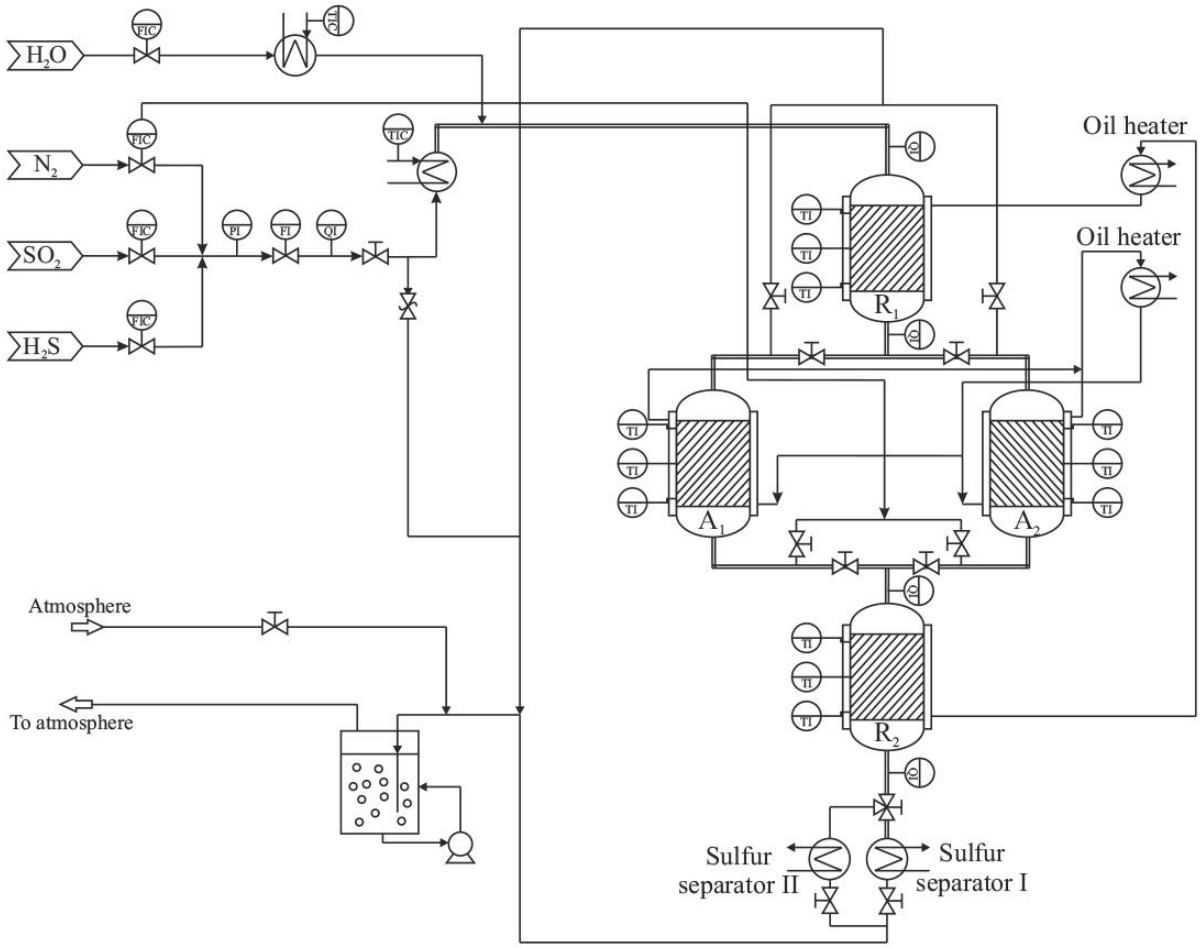


Figure 46: Flow diagram of the bench-scale plant for Claus process in the RAR-arrangement

The desired flow rates of reactants H_2S and SO_2 (Table 12) as well as of inert gas N_2 are regulated by means of mass flow controllers. Increased flow rates are necessary to study adiabatic operation, since otherwise the heat losses will dissipate the heat liberated by reaction and/or adsorption and thus distort the temperature profiles measured. Figure 47 shows the installed mass flow controllers at the bench-scale plant followed by a digital differential pressure manometer, to detect any blockage that may occur in the entire plant during operation, and a rotameter was installed afterwards to

measure the volume flow of the incoming mixed gas stream to the insulated preheater (shown in Figure 48), where it is electrically heated up to the operating temperature of around 250 °C before they enter the first reaction step.

Table 12: Flowrate ranges of the involved components in Claus adsorptive reactor operation

Component	Minimum flow rate [l/min]	Maximum flow rate [l/min]
N_2	3.5	35
H_2S	0.25	2.5
SO_2	0.125	1.25

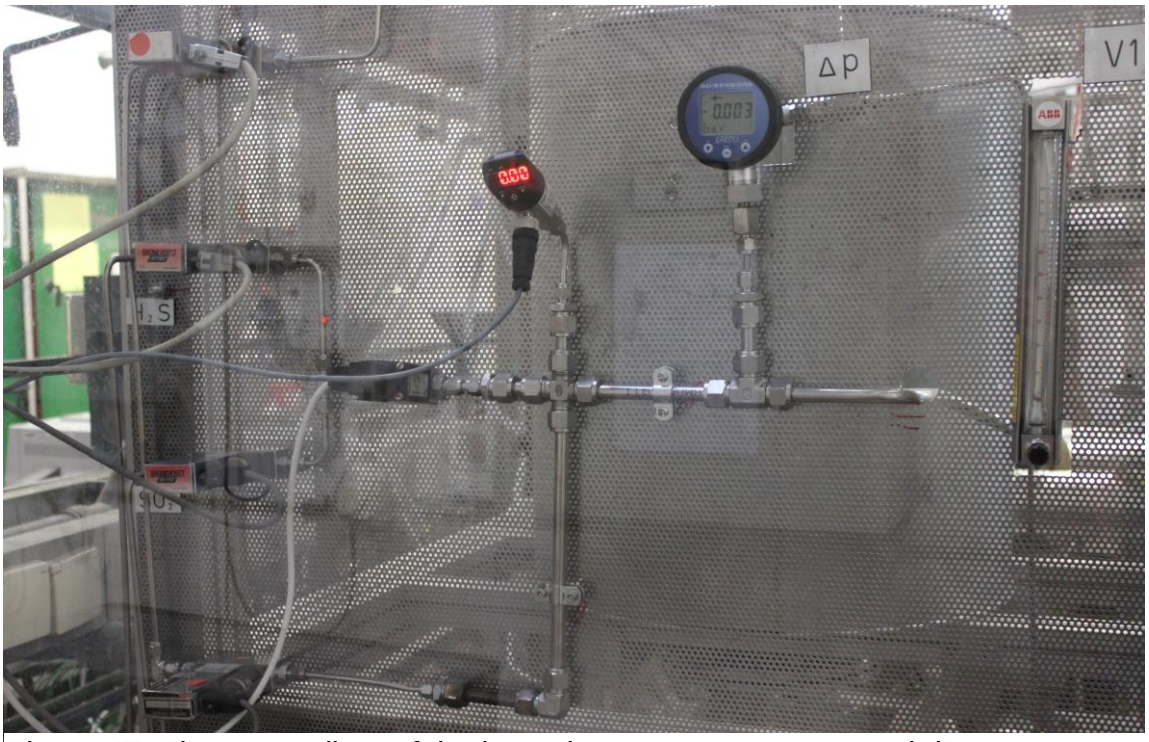


Figure 47: Flow controllers of the incoming gas components and dP measurement of the bench scale plant

Figure 48: The electrical preheater of the inflowing gas stream

Since that in the technical Claus process H_2O-vapour is present in the feed, additional H_2O-vapour will be deliberately introduced into the feed in order to evaluate its influence on the reactor performance. The H_2O-dosing-unit comprises a mass flow controller and an electrically heated evaporator.

After the first reactor stage, the gases pass through one of the two periodically operated adsorber columns where the H_2O-vapour is to be removed. While one of the identical parallel adsorbers is serving as an adsorber, the other is being regenerated by raising the temperature and flushing it through with nitrogen. The gases leaving the adsorption step are then fed to the second reactor operating under the same conditions as the first. Elemental sulphur formed in the Claus-reaction is then separated downstream of the second reactor in one of two scraped-wall desublimators (see Figure 49). This parallel arrangement is necessary to guarantee prolonged uninterrupted operation, since the desublimator is subject to occasional blockages. Any residual H_2S and SO_2 remaining in the gas stream are absorbed by passing them through NaOH-solution in a neutralisation tank.

The 3A zeolite adsorbent and the promoted alumina (type: DD-931) Claus reaction catalyst, provided by BASF SE, were used for the experiments. The particles of both the adsorbent and catalyst have a spherical shape of 4 mm diameter.

The samples of the gas taken at different points of the bench-scale plant were analysed using a gas-chromatography device of the type GC-2010 Plus from the manufacturer Shimadzu. The separation column is an HP-Plot Q, which is particularly appropriate for separating inorganic substances. The detector is a thermal conductivity detector, which uses helium as the reference gas. Helium is also used as a carrier gas for the separation in the column. The sample is taken using a gas-tight syringe and then manually injected into the gas chromatograph. The sample volume can be adjusted by using different syringes and ranges from 50 µl to 25 ml. To avoid contamination of the environment with the toxic gases, a fume extractor was mounted directly above the gas chromatograph, as it can be seen in Figure 50.

Figure 49: Scraped-wall desublimators for sulphur separation

The RAR-concept encountered problems in setting up the pilot plant. The joints between the glass reactor or adsorber and the pipeline (Figure 51) could not be adequately sealed, and the gas leakage could not be avoided. A big problem was to

get the glass-metal connections tight. Often the graphite seal slipped during the screwing process and there was no longer a proper sealing surface. Slippage of the

Figure 50: A fume extractor mounted above the gas chromatograph

gasket also damaged the gasket itself, as it lay obliquely on the glass or metal flange. Consequently, the seal had to be disposed afterwards. For twenty glass-metal joints (5 joints per reactor or adsorber), it was very difficult to get every joint tight. Switching to stainless steel reactor/adsorber was unfortunately not feasible as the new plant parts needed a very long time to be manufactured.

For the reasons mentioned above and the high risk that the toxic gases leak, the RAR-concept was replaced by the integrated adsorptive reactor concept (IAR) which could be successfully installed. Figure 52 shows both the RAR and IAR concepts.

a)  b) 

Figure 51: Glass-metal-joint; a) assembled, and b) dissembled

At the beginning, the IAR was filled by a catalyst-inert particles mixture with a uniformly distributed catalyst volumetric ratio of 57%, so that the benchmark case could firstly be investigated. The reactor was run under isothermal conditions in a temperature range of 230 to only 250 °C because of the thermal stability of the seals within the valves.

The experiments for the benchmark at temperatures 230 and 250 °C have shown that the reactants H_2S and SO_2 were completely converted, i.e. the measured conversion at the outlet of the reactor was always 100%. The measured conversion value before the inlet of the reactor was already over 50%. The fast kinetics of the Claus reaction thus enables rapid conversion of the reactants even before the reactor.

For this reason, further attempts at lower operating temperatures were executed, but the overall conversion measured at the outlet of the reactor was always at the 100% value.

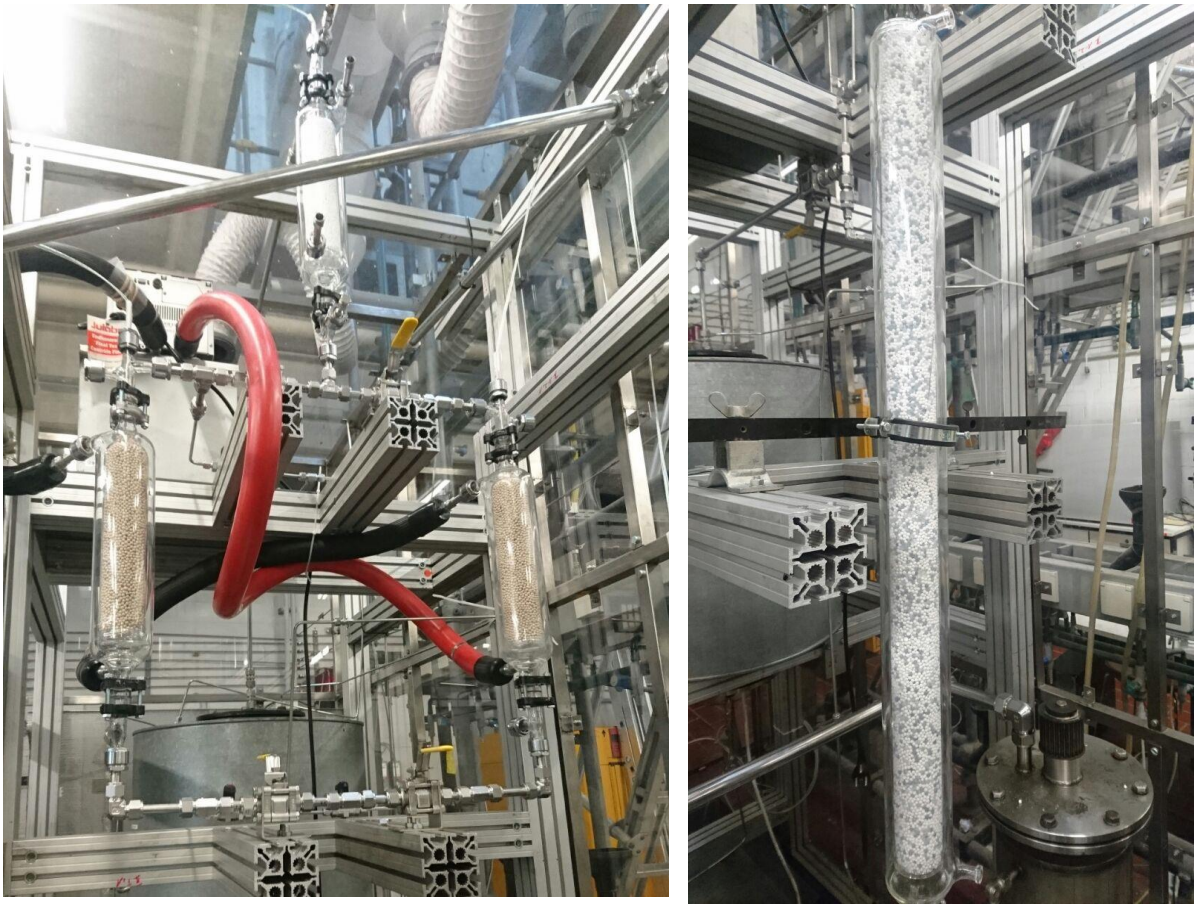


Figure 52: The bench-scale adsorptive reactor for Claus process: the spatially segregated arrangement/RAR (left), and the uniformly distributed functionalities/IAR (right)

The only explanation for this complete conversion is that the plant was operated at temperatures under the sulphur condensation point in which a continuous condensation of the sulphur occurs shifting the equilibrium, according to Le Chatelier's principle, towards the products side meaning more sulphur condensation until blockage in the plant occurred, Figure 53. Unfortunately, experiments at temperatures higher than 250 °C could not be carried out because of the seals used in the valves not to be damaged and the appropriate seals were beyond the available budget for this research study.

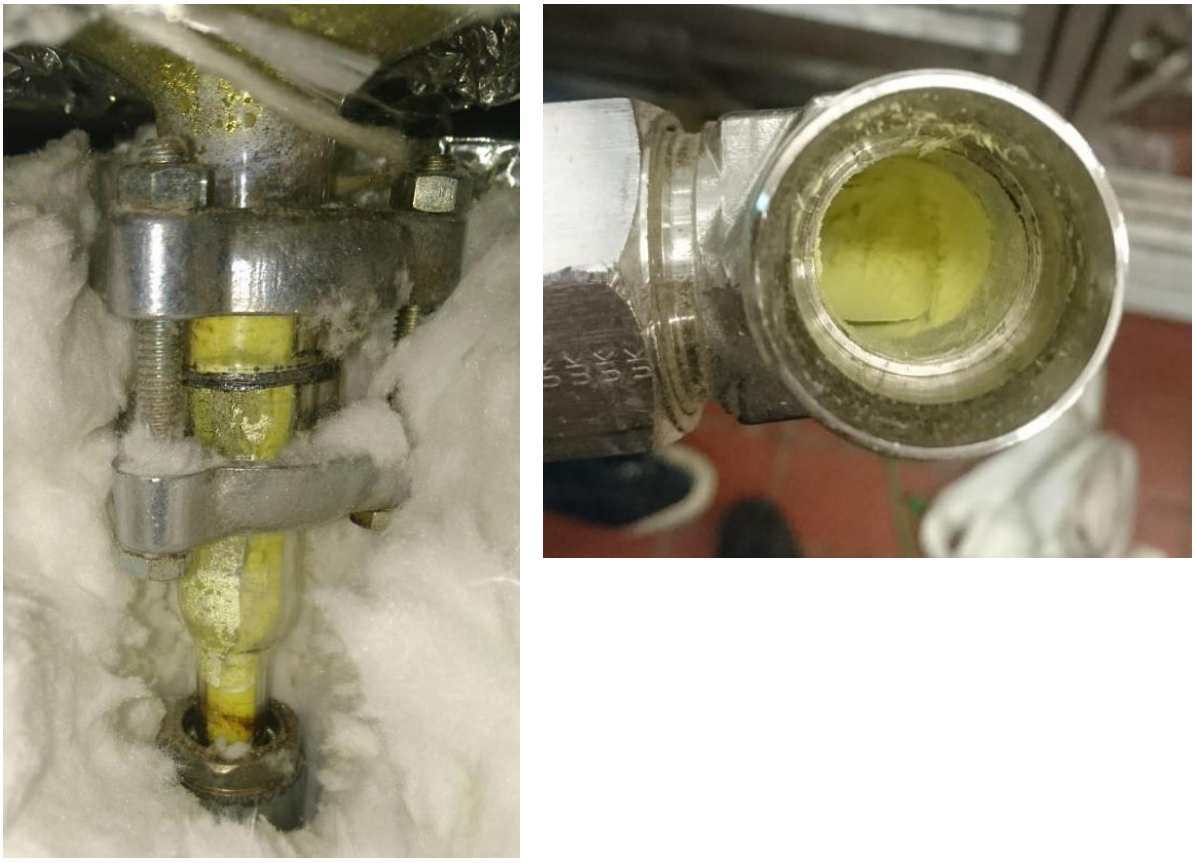


Figure 53: Blockage caused by sulphur condensation at the reactor outlet (left) and at the desublimator outlet (right)

5.2.2. Remarks on scale-up of adsorptive reactors

In addition to the successful experimental plant-scale verification of the proposed adsorptive reactor designs, the development of an appropriate scale-up methodology is necessary to commercialise the implementation of the adsorptive reactor concept. Due to the complex nature of the adsorptive reactor operation, keeping the decisive dimensionless numbers constant while scaling-up to ensure the geometrical and dynamic similarities of different sizes becomes very hard to establish. Instead, the so-called ‘horizontal scale-up’ or ‘scale-out’ would then be an appropriate option, in which numbering-up, or ‘scaling in parallel’, of the small spatially segregated adsorptive sandwich reactors to form a tube bundle (multitubular) reactor would efficiently match the industrial needs achieving at the same time a better performance prediction of the process and a better control of the scale effect.

The industrial multitubular reactor can contain several thousand, up to 20 000 packed bed tubes and can operate adiabatically or even under non-adiabatic conditions in a shell-and-tube arrangement with countercurrent or cross flow heat exchange [120, 121]. In the latter case, the ratio between the mass transfer area and reactor volume should be as high as possible to achieve efficient heat exchange. Therefore, the smallest possible diameter of the packed bed tubes is to be chosen [121]. The most important rules of thumb when designing and upscaling multitubular reactors are summarised in Table 13 [122].

Table 13: Rules of thumb for designing multi-tubular reactors

Characteristic parameter	Value
$\frac{L}{d_P}$	> 100
$\frac{d_P}{d_R}$	≤ 0.1
d_P	1 – 5 mm
$d_{R,max}$	6 cm
ΔP_{max}	$0.1 \times P$

REFERENCES CITED

[1] D. W. Agar, W. Ruppel, Multifunktionale Reaktoren für die heterogene Katalyse, Chem. –Ing. –Tech. (1988), 60, 731 – 741.

[2] U. Hoffmann, K. Sundmacher, Multifunktionale Reaktoren, Chem.-Ing.-Tech. (1997), 69, 613-622.

[3] D. W. Agar, Multifunctional reactors: Old preconceptions and new dimensions, Chem. Eng. Sci. (1999), 54, 1299-1305.

[4] D.W. Agar, Dos and Don'ts of adsorptive reactors, Ch. 7 in Integrated Chemical Processes - Synthesis, Operation, Analysis, and Control, Eds. K. Sundmacher, A. Kienle, A. Seidel-Morgenstern, Wiley-VCH, Weinheim 1st. Edition (2006), ISBN: 978-3-527-60555-2.

[5] M. Ghaedi, Adsorption: Fundamental Processes and Applications, Elsevier, 1st Edition (2021), ISBN: 9780128188057.

[6] C. Tien, Introduction to Adsorption, Elsevier, 1st Edition (2018), ISBN: 9780128164464.

[7] B. B. Fish, R. W. Carr, An experimental study of the countercurrent moving-bed chromatographic reactor, Chem. Eng. Sci. (1989), 44, 1773 – 1783.

[8] B. D. Unger, R. G. Rinker, Ammonia synthesis reaction in the chromatographic regime, Ind. & Eng. Chem. Fund. (1976), Vol. 15, No. 3, 225 – 227.

[9] A. L. Tonkovich, R. W. Carr, R. Aris, Enhanced C2 yields from methane oxidative coupling by means of a separative chemical reactor, Science (1993), 262(10), 221 – 223.

[10] E. Drioli, M. Romano, Progress and new perspectives on integrated membrane operations for sustainable industrial growth, Industrial & Engineering Chemistry Research (2001), 40, 1277.

[11] A. Julbe, D. Farrusseng, C. Guizzard, Porous ceramic membranes for catalytic reactors – overview and new ideas, Journal of Membrane Science (2001), 181, 3.

[12] M. Kuczynski, M.H. Oyevaar, R.T. Pieters, K.R. Westerterp, Methanol synthesis in a countercurrent gas-solid-solid trickle flow reactor: An experimental study, Chemical Engineering Science (1987), Vol. 42, No. 8, p. 1887 – 1898.

[13] J.R. Hufton, S. Mayorga, S. Sircar, Sorption-enhanced reaction process for hydrogen production, AIChE J. (1999), 45, 248–256.

[14] W. E. Waldron, J. R. Hufton, S. Sircar, Production of Hydrogen by cyclic sorption enhanced reaction process, AIChE Journal (2001), Vol. 47, No. 6, p. 1477-1479.

[15] R. C. Soares, J. M. Loureiro, C. Sereno, A. E. Rodrigues, Modelling and simulation of carbon mask adsorptive reactors, Ind. Eng. Chem. Res. (1995), 34 (8), 2762–2768.

[16] Z. P. Lu, A. E. Rodrigues, Pressure swing adsorption reactors: simulation of three-step one-bed process, AIChE J. (1994), 40 (7), 1118-1137.

[17] I. Yongsunthon, E. Alpay, Conversion-temperature trajectories for well-mixed adsorptive reactors, Chem. Eng. Sci. (1998), 53 (4), 691-696.

[18] I. Yongsunthon, E. Alpay, Total connectivity models for adsorptive reactor design, Chem. Eng. Sci. (2000), 55 (23), 5643-5656.

[19] Y. Ding, E. Alpay, Adsorption-enhanced steam-methane reforming, Chem. Eng. Sci. (2000), 55 (18), 3929-3940.

[20] Y. Ding, E. Alpay, Equilibria and kinetics of CO2 adsorption on hydrotalcite adsorbent, Chem. Eng. Sci. (2000), 55 (18), 3461-3474.

[21] G-H. Xiu, J. L. Soares, P. Li, A. E. Rodrigues, Simulation of five-step one-bed sorption-enhanced reaction process, AIChE J. (2002), 48, 2817–2832.

[22] G-H Xiu, P. Li, A.E. Rodrigues, New generalized strategy for improving sorption-enhanced reaction process, Chem. Eng. Sci. (2003), 58, 3425-3437.

[23] M. P. Elsner, Experimentelle und modellbasierte Studien zur Bewertung des adsorptiven Reaktorkonzeptes am Beispiel der CLAUS-Reaktion, Ph.D. Thesis (2004), Technische Universität Dortmund.

[24] P. S. Lawrence, M. Grünewald, W. Dietrich, D. W. Agar, Optimal Distribution of Catalyst and Adsorbent in an Adsorptive Reactor at the Reactor Level, Ind. Eng. Chem. Res. (2006), 45 (14), 4911–4917.

[25] M. Grünewald, D. W. Agar, Enhanced catalyst performance using integrated structured functionalities, Chem. Eng. Sci. (2004), 59 (22-23), 5519-5526.

[26] P. S. Lawrence, M. Grünewald and D. W. Agar, Spatial distribution of functionalities in an adsorptive reactor at the particle level, Catal. Today (2005), 105 (3-4), 582-588.

[27] W. Dietrich, P.S. Lawrence, M. Grünewald, D.W. Agar, Theoretical studies on multi-functional catalysts with integrated adsorption sites, Chem. Eng. Journal (2005), 107, p.103-111.

[28] P.S. Lawrence, Distribution of functionalities at different scales within a multifunctional reactor, PhD Thesis (2005), TU Dortmund, Germany.

[29] S. Reßler, D.W. Agar, Einsatz eines adsorptiven Reaktors am Beispiel der direkten Synthese von Dimethylether aus Synthesegas, Chemie Ingenieur Technik (2005), 77 (8), p. 1029.

[30] P. S. Lawrence, M. Grünewald, Multifunctionality at particle level - case studies for adsorptive catalysts, in Integrated reaction and separation operations (2006), Springer, Berlin, Hrsgbr. H. Schmidt-Traub, A. Gorak, ISBN: 3540301488.

[31] S. Reßler, M.P. Elsner, Chr. Dittrich, D.W. Agar, S. Geisler, O. Hinrichsen, Reactive gas adsorption, in Integrated reaction and separation operations: modelling and experimental validation (2006), Springer-Verlag, Berlin, Hrsgbr. H. Schmidt-Traub, A. Gorak, ISBN: 3540301488

[32] M. Richrath, S. Lohse, M. Grünewald, D.W. Agar, Particle-scale heat removal in fixed bed catalytic reactors: Modelling and optimisation of a desorptive cooling process, ESCAPE 15, Computer-Aided Chemical Engineering (2005), Volume 20A, 673 – 678.

[33] M. Richrath, Desorptive Kühlung chemischer Reaktoren: Untersuchungen zur Kopplung von Reaktions- und Desorptionsprozessen in katalytischen Festbetten, PhD Thesis (2007), TU Dortmund, Germany.

[34] M. Nau, Desorptive Kühlung chemischer Reaktoren: Untersuchungen zur Selektivitätssteigerung und Maßstabsvergrößerung, PhD Thesis (2011), TU Dortmund, Germany.

[35] M. Grünewald, Multifunktionale Katalysatoren: Studien zur Integration und Strukturierung von Funktionalitäten auf mikroskopischer Skala, Habilitationsschrift (2006), TU Dortmund, Germany.

[36] M. H. Halabi, M. H. J. M. de Croon, J. van der Schaaf, P. D. Cobden, J. C. Schouten, Reactor modeling of sorption-enhanced autothermal reforming of methane. Part I: Performance study of hydrotalcite and lithium zirconate-based processes, Chemical Engineering Journal (2011), 168, 872 – 882.

[37] M. H. Halabi, M. H. J. M. de Croon, J. van der Schaaf, P. D. Cobden, J. C. Schouten, Reactor modeling of sorption-enhanced autothermal reforming of methane. Part II: Effect of operational parameters, Chemical Engineering Journal (2011), 168, 883 – 888.

[38] B. Duo, C. Wang, Y. Song, H. Chen, B. Jiang, M. Yang, Y. Xu, Solid sorbents for in-situ CO_2 removal during sorption-enhanced steam reforming process: A review, Renewable and Sustainable Energy Reviews (2016), 53, 536 – 546.

[39] E. L. Lugo, B. A. Wilhite, A theoretical comparison of multifunctional catalyst for sorption-enhanced reforming process, Chemical Engineering Science (2016), 150, 1 – 15.

[40] Z. Chao, Y. Zhang, Y. Wang, J. P. Jakobsen, H. A. Jakobsen, Modelling of binary fluidized bed reactors for the sorption-enhanced steam methane reforming process, The Canadian Journal of Chemical Engineering (2017), 95(1), 157 – 169.

[41] G. Diglio, D. P. Hanak, P. Bareschino, E. Mancusi, F. Pepe, F. Montagnaro, V. Manovic, Techno-economic analysis of sorption-enhanced steam methane reforming in a fixed bed reactor network integrated with fuel cell, Journal of Power Sources (2017), 364, 41 – 51.

[42] Y. Wang, M. Z. Memon, M. A. Seelro, W. Fu, Y. Gao, Y. Dong, G. Ji, A review of CO_2 sorbents for promoting hydrogen production in the sorption-enhanced steam reforming process, International Journal of Hydrogen Energy (2021), 46(45), p. 23358 – 23379.

[43] Z. Chao, Y. Zhang, Y. Wang, J. P. Jakobsen, H. A. Jakobsen, Modelling of binary fluidized bed reactors for the sorption-enhanced steam methane reforming process, International Journal of Hydrogen Energy (2021), 46(45), 23358 – 23379.

[44] S. M. Soltani, A. Lahiri, H. Bahzad, P. Clough, M. Gorbounov, Y. Yan, Sorption-enhanced Steam Methane Reforming for Combined CO_2 Capture and Hydrogen Production: A State-of-the-Art Review, Carbon Capture Science & Technology (2021), 1, 1 – 17.

[45] A. Phuluanglue, W. Khaodee, S. Assabumrungrat, Simulation of Intensified Process of Sorption Enhanced Chemical-Looping Reforming of Methane: Comparison with Conventional Processes, Computers & Chemical Engineering (2017), 105.

[46] R. Habibi, F. Pourfayaz, M. Mehrpooya, H. Kamali, A natural gas-based eco-friendly polygeneration system including gas turbine, sorption-enhanced steam methane reforming, absorption chiller and flue gas CO_2 capture unit, Sustainable Energy Technologies and Assessments (2022), 52(A), p. 101984.

[47] M. Ayesha, A. H. Khoja, F. A. Butt, U. Sikandar, A. H. Javed, S. R. Naqvi, I. ud din, M. T. Mehran, Sorption enhanced steam reforming of methane over waste-derived CaO promoted MgNiAl hydrotalcite catalyst for sustainable H2 production, Journal of Environmental Chemical Engineering (2022), 10(3), p. 107651.

[48] A. Gunawan, A. K. Singh, A solar thermal sorption-enhanced steam methane reforming (SE-SMR) approach and its performance assessment, Sustainable Energy Technologies and Assessments (2022), 52(A), p. 102036.

[49] Z. Li, W. Cai, C. Dang, Sorption-enhanced steam reforming of CH4/CO2 synthetic mixture representing biogas over porous Ni–CaO–MgO microsphere via a surface modified carbon template, International Journal of Hydrogen Energy (2022), 47(77), p. 32776 – 32786.

[50] V. S. Sikarwar, C. Pfeifer, F. Ronsse, M. Pohorely, E. Meers, A. K. Kaviti, M. Jeremias, Progress in in-situ CO2- sorption for enhanced hydrogen production, Progress in Energy and Combustion Science (2022), 91, p. 101008.

[51] B. Li, S. Wang, X. Yang, Q. Wu, Y. He, Thermodynamic evaluation of sorption-enhanced chemical looping gasification with coal as fuel, International Journal of Hydrogen Energy (2020), 45(41), p. 21186 – 21194.

[52] Z. Cui, S. Sun, H. Zhang, B. Liu, W. Tian, Q. Guo, Comprehensive optimization of coal chemical looping gasification process for low CO_2 emission based on multi-scale simulation coupled experiment, Fuel (2022), 324(A), p. 124464.

[53] S. Wang, B. Li, X. Yang, Y. He, Performance of sorption-enhanced chemical looping gasification system coupled with solid oxide fuel cell using exergy analysis, International Journal of Hydrogen Energy (2021), 46(2), p. 1752 – 1761.

[54] B. Li, C. F. M. Mbeugang, D. Liu, S. Zhang, S. Wang, Q. Wang, Z. Xu, X. Hu, Simulation of sorption enhanced staged gasification of biomass for hydrogen production in the presence of calcium oxide, International Journal of Hydrogen Energy (2020), 45(51), p. 26855 – 26864.

[55] G. Liu, Y. Zhao, S. Heberlein, A. Veksha, A. Giannis, W. P. Chan, T. T. Lim, G. Lisak, Hydrogen and power co-production from autothermal biomass sorption enhanced chemical looping gasification: Thermodynamic modeling and comparative study, Energy Conversion and Management (2022), 269, p. 116087.

[56] A. Pitkäoja, J. Ritvanen, S. Hafner, T. Hyppänen, G. Scheffknecht, Simulation of a sorbent enhanced gasification pilot reactor and validation of reactor model, Energy Conversion and Management (2020), 204, p. 112318.

[57] A. Pitkäoja, J. Ritvanen, S. Hafner, T. Hyppänen, G. Scheffknecht, Numerical modelling of sorption-enhanced gasification: Development of a fuel decomposition model, Fuel (2021), 289, p. 119868.

[58] A. M. Parvez, S. Hafner, M. Hornberger, M. Schmid, G. Scheffknecht, Sorption enhanced gasification (SEG) of biomass for tailored syngas production with in-situ CO2 capture: Current status, process scale-up experiences and outlook, Renewable and Sustainable Energy Reviews (2021), 141, p. 110756.

[59] X. Yang, T. Kan, A. Kheradmand, H. Xu, V. Strezov, A. Yu, Y. Jiang, Tunable syngas production from two-stage sorption-enhanced steam gasification of sewage sludge, Chemical Engineering Journal (2021), 404, p. 126069.

[60] M. P. S. Santos, D. P. Hanak, Techno-economic feasibility assessment of sorption enhanced gasification of municipal solid waste for hydrogen production, International Journal of Hydrogen Energy (2022), 47(10), p. 6586 – 6604.

[61] J. Dai, K. J. Whitty, Chemical looping gasification and sorption enhanced gasification of biomass: A perspective, Chemical Engineering and Processing – Process Intensification (2022), 174, p. 108902.

[62] W.-H. Chen, C.-Y. Chen, Water gas shift reaction for hydrogen production and carbon dioxide capture: A review, Applied Energy (2020), 258, p. 114078.

[63] C. Zhang, Y. Li, Z. He, J. Zhao, D. Wang, Microtubular Fe/Mn-promoted CaO-Ca12Al14O33 bi-functional material for H_2 production from sorption enhanced water gas shift, Applied Catalysis B: Environmental (2022), 314, p. 121474.

[64] N. Saithong, S. Authayanun, Y. Patcharavorachot, A. Arpornwichanop, Thermodynamic analysis of the novel chemical looping process for two-grade hydrogen production with CO_2 capture. Energy Conversion and Management 180 (2019), 325 – 337.

[65] B. Li, C. F. M. Mbeugang, D. Liu, S. Zhang, S. Wang, Q. Wang, Z. Xu, X. Hu, Simulation of sorption enhanced staged gasification of biomass for hydrogen production in the presence of calcium oxide, International Journal of Hydrogen Energy (2020), 45(51), 26855 – 26864.

[66] H. Lee, B. Lee, M. Byun, H. Lim, Comparative techno-economic analysis for steam methane reforming in a sorption-enhanced membrane reactor: Simultaneous H2 production and CO_2 capture, Chemical Engineering Research and Design (2021), 171, 383 – 394.

[67] S. Rawadieh, V.G. Gomes, Steam reforming for hydrogen generation with in-situ adsorptive separation, Int. J. Hydrogen Energy 34 (2009) 343-355.

[68] A. Gorbach, G. Eigenberger, G. Kolios, General Approach for the Reduction of Detailed Models for Fast Cycling Processes, Ind. Eng. Chem. Res. (2005), 44, 2369–2381.

[69] G. Kolios, D. Luss, R. Garg, G. Viswanathan, Efficient computation of periodic state of cyclic fixed bed processes, Chem. Eng. Sci. (2013), 101, 90-98.

[70] A.G. Salinger, G. Eigenberger, The direct calculation of periodic states of the reverse flow reactor—I. Methodology and propane combustion results, Chem. Eng. Sci. (1996), 51, 4903-4913.

[71] D. T. Croft, M.D. LeVan, Direct Determination and Multiplicity of Periodic States of Adsorption Cycles, Stud. Surf. Sci. Catal. (1993), 80, 113-120.

[72] I. Yongsunthon, E. Alpay, Design of periodic adsorptive reactors for the optimal integration of reaction, separation and heat exchange, Chem. Eng. Sci. (1999), 54, 2647–2657.

[73] Y. Ding, M.D. LeVan, Periodic states of adsorption cycles III. Convergence acceleration for direct determination, Chem. Eng. Sci. (2001), 56, 5217-5230.

[74] M.D. LeVan, D.T. Croft, Determination of periodic states of pressure swing adsorption cycles, Gas Sep. Purif. (1995), 9, 13-16.

[75] L. Jiang, L.T. Biegler, V.G. Fox, Simulation and optimisation of pressure-swing adsorption systems for air separation, AIChE J. (2003), 49, 1140–1157.

[76] A.-A. Munera-Parra, C. Asmanoglo, D.W. Agar, Cyclic Steady-State Behaviour of a Fixed bed Adsorptive Reactor for Reverse Water-Gas Shift Reaction, Chem. Eng. Technol. (2017), 40, 915–926.

[77] T. J. Stadler, J-H. Knoop, S. Decker, P. Pfeifer, Numerical Simulation Approach for a Dynamically Operated Sorption-Enhanced Water-Gas Shift Reactor, Processes (2022),10(6), 1160.

[78] A. Arora, S.S. Iyer, M. M. F. Hasan, GRAMS: A General Framework Describing Adsorption, Reaction and Sorption-Enhanced Reaction Processes, Chemical Engineering Science (2018),192, 335 - 358.

[79] A. Arora, S. S. Iyer, I. Bajaj, M. M. F. Hasan, Optimal Methanol Production via Sorption-Enhanced Reaction Process, Industrial and Engineering Chemistry Research (2018), 57(42).

[80] A. Arora, I. Bajaj, S. S. Iyer, M. M. F. Hasan, Optimal synthesis of periodic sorption enhanced reaction processes with application to hydrogen production, Computers & Chemical Engineering (2018), 115, 89 - 111.

[81] A. Zachopoulos, E. Heracleous, Overcoming the equilibrium barriers of CO_2 hydrogenation to methanol via water sorption: A thermodynamic analysis, Journal of CO_2 Utilization (2017), 21, 360 - 367.

[82] Y. Tian S. M. Demirel, M. M. F. Hasan, E. N. Pistikopoulos, An overview of process systems engineering approaches for process intensification: State of the art, Chemical Engineering and Processing (2018), 133-C.

[83] S. Jung, S. Reining, S. Schindler, D.W. Agar, Anwendung von adsorptiven Reaktoren für die reverse Wassergas-Shift-Reaktion, Chemie-Ingenieur-Technik (2013), 85 (4), 484-488.

[84] S. Goto, T. Tagawa, T. Oomiya, Dehydrogenation of cyclohexane in a PSA reactor using hydrogen storage alloy, Kagaku-kogaku-ronbonshu (Chem. Eng. Essays) (1993), 19(6), 978 - 983 (Jap. Text, Engl. Abstract).

[85] Y.J. Wu, P. Li, J.G. Yu, A.F. Cunha, A.E. Rodrigues, Progress on Sorption-Enhanced Reaction Process for Hydrogen Production: A Review. Reviews in Chemical Engineering (2016), 32, 271-304.

[86] A.E. Rodrigues, L. M. Madeira, Y-J. Wu, R. Faria, Sorption Enhanced Reaction Processes, World Scientific Europe Ltd (2017), ISBN: 1786343568.

[87] G-H. Xiu, P. Li, A.E. Rodrigues, Sorption-enhanced reaction process with reactive re-generation, Chem. Eng. Sci. (2002), 57, p. 3893-3908.

[88] K. T. Coenen, F. Gallucci, P. Cobden, E. van Dijk, E. J. M. Hensen, M. van Sint Annaland, Chemisorption working capacity and kinetics of CO_2 and H_2O of hydrotalcite-based adsorbents for sorption-enhanced water-gas-shift applications, Chemical Engineering Journal (2016), 293, p. 9 -23.

[89] A.L. Tonkovich, R.W. Carr, R. Aris, Enhanced C_2 Yields from Methane Oxidative Coupling by Means of a Separative Chemical Reactor, Science (1993), Vol. 262, p. 221 - 223.

[90] M.P. Elsner, M. Menge, D.W. Agar, The Claus process: teaching an old dog new tricks, Catal. Today (2003), 79-80, p. 487-494.

[91] M.P. Elsner, C. Dittrich, D.W. Agar, Adsorptive reactors for enhancing equilibrium gas-phase reactions – Two case studies, Chem. Eng. Sci. (2002), 57(9), p. 1607 - 1619.

[92] C. Dittrich, Bewertung eines adsorptiven Reaktors für die direkte Cyanwasserstoff-Synthese aus Ammoniak und Kohlenmonoxid, PhD Thesis (2002), TU Dortmund, Germany.

[93] E. Alpay, D. Chatsiriwech, L.S. Kershenbaum, Combined reaction and separation in pressure swing processes, Chem. Eng. Sci. (1994), 49, 24B, 5845 – 5864.

[94] K. Iwanaga, T. Suzuta, K. Seki, M. Nakada, T. Hibi, Y. Mori, K. Issoh, T. Abe, The Development of Improved Hydrogen Chloride Oxidation Process, SUMITOMO KAGAKU (2004), vol. 2004-I.

[95] J. Pérez-Ramírez, C. Mondelli, T. Schmidt, O. F.-K. Schlüter, A. Wolf, L. Mleczko, T. Dreierb, Sustainable chlorine recycling via catalysed HCl oxidation: from fundamentals to implementation, Energy Environ. Sci. (2011), 4 (12), 4786-4799.

[96] A.P. Amrute, Deacon chemistry revisited: New catalysts for chlorine recycling, Ph.D. Thesis (2004), ETH Zürich.

[97] V. Scharl, F. Fischer, S. Herrmann, S. Fendt, H. Spliethoff, Applying Reaction Kinetics to Pseudohomogeneous Methanation Modeling in Fixed-Bed Reactors, Chemical Engineering & Technology (2020), 43(6), 1224 - 1233.

[98] J. F. Horstmeier, S. Heib, D. W. Agar, Simulation der prozessinternen Rückgewinnung von Adsorptionswärmen durch Latentwärmespeicher: Simulation of Internal Recycling of Adsorption Heat with Phase Change Materials, Chemie Ingenieur Technik 86 (2014), 97–105.

[99] D. Teschner, R. Farra, L. Yao, R. Schlögl, H. Soerijanto, R. Schomäcker, T. Schmidt, L. Szentmiklósi, A. P. Amrute, C. Mondelli, J. Pérez-Ramírez, G. Novell-Leruth, N. López., An integrated approach to Deacon chemistry on RuO2-based catalysts, J. Catal. (2012), 285 (1), 273-284.

[100] F.W. Dynys, J.W. Halloran, Alpha alumina formation in alum-derived gamma alumina, Journal of the American Ceramic Society (1982), 65, 442–448.

[101] G. Bercic, J. Levec, Intrinsic and global reaction rate of methanol dehydration over gamma-alumina pellets, Industrial & engineering chemistry research (1992), 31, 1035–1040.

[102] S.R. Chowdhury, R. Schmuhl, K. Keizer, ten Elshof, Johan E, D.H.A. Blank, Pore size and surface chemistry effects on the transport of hydrophobic and hydrophilic solvents through mesoporous γ-alumina and silica MCM-48, Journal of membrane science (2003), 225, 177–186.

[103] M. Simo, S. Sivashanmugam, C.J. Brown, V. Hlavacek, Adsorption/desorption of water and ethanol on 3A zeolite in near-adiabatic fixed bed, Ind. Eng. Chem. Res. (2009), 48, 9247–9260.

[104] D. M. Ruthven, Principles of Adsorption and Adsorption Processes, Wiley, (1984), ISBN: 9780471866060.

[105] R. T. Yang, Gas Separation by Adsorption Processes, Chemical Engineering - Imperial College Press, (1997), ISBN: 9781860940477.

[106] S.M. Auerbach, K.A. Carrado, P.K. Dutta, Handbook of zeolite science and technology, CRC press, (2003).

[107] M. Simo, Pressure Swing Adsorption Process for Ethanol Dehydration, in Proceedings (2008).

[108] M. S. Shafeeyan, W.M.A. Wan Daud, A. Shamiri, A review of mathematical modelling of fixed bed columns for carbon dioxide adsorption, Chem. Eng. Res. Des. (2014), 92(5), 961-988.

[109] N. Wakao, T. Funazkri, Effect of fluid dispersion coefficients on particle-to-fluid mass transfer coefficients in packed beds: Correlation of sherwood numbers, Chem. Eng. Sci. (1978), 33(10), 1375-1384.

[110] N. Wakao, S. Kaguei, T. Funazkri, Effect of fluid dispersion coefficients on particle-to-fluid heat transfer coefficients in packed beds: Correlation of Nusselt numbers, Chem. Eng. Sci. (1979), 34(3), 325-336.

[111] G.F. Froment, K.B. Bischoff, J. De Wilde, Chemical Reactor Analysis and Design, 3rd Edition, Wiley, (2010), ISBN: 978-0-470-56541-4.

[112] J.F. Horstmeier, S. Heib, J. Herrmann, D. Keseberg, D.W. Agar, Prozessinterne Rückgewinnung der Adsorptionswärme zur effizienten CO_2-Adsorption aus Industrieabgasen, DECHEMA ProcessNet Jahrestreffen Fach-gruppe Adsorption (2013).

[113] K.H. Stem, High temperature properties and decomposition of inorganic salts Part 3. Nitrates and nitrites, J. Phys. Chem. Ref. Data (1972), 1.

[114] M.M. Kenisarin, High-temperature phase change materials for thermal energy storage, Renewable and Sustainable energy reviews (2010), 14, 955–970.

[115] Kátia S. do Couto Aktay, R. Tamme, H. Müller-Steinhagen, Thermal conductivity of high-temperature multicomponent materials with phase change, International Journal of Thermophysics (2008), 29, 678–692.

[116] L.R. White, H.T. Davis, Thermal conductivity of molten alkali nitrates, The Journal of Chemical Physics (1967), 47, 5433–5439.

[117] M. N. Gibbs, D. J. MacKay, Variational gaussian process classifiers, IEEE Transactions on Neural Networks (2000), 11 (6), 1458 - 1464.

[118] Y. Lu, A. G. Dixon, W. R. Moser, Y. H. Ma, Analysis and optimization of cross-flow reactors with staged feed policies - isothermal operation with parallel-series, irreversible reaction systems, Chemical Engineering Science (1997), 52 (8), 1349 - 1363.

[119] Y. Lu, A. G. Dixon, W. R. Moser, Y. H. Ma, Analysis and optimization of cross-flow reactors with distributed reactant feed and product removal, Catalysis Today (1997), 35 (4), 443 - 450.

[120] K. Hertwig, L. Martens, Chemische Verfahrenstechnik, Berechnung, Auslegung und Betrieb chemischer Reaktoren, Köthen : Oldenburg Wissenschaftsverlag GmbH, (2012), ISBN: 978-3-487-70890-5.

[121] J. Hagen, Chemiereaktoren: Auslegung und Simulation, Weinheim: Wiley-VCH Verlag GmbH& Co. KG, (2004), ISBN: 3-527-30827-X.

[122] D. R. Woods, Rules of Thumb in Engineering Practice, Weinheim: Wiley-VCH Verlag GmbH & Co. KG, (2007), ISBN: 978-3-527-31220-7.

[123] C. W. Arnold, K. A. Kobe, Thermodynamics of the deacon process, Chemical Engineering Progress (1952), 48, p. 293 - 296.

Declaration on the reproduction of previously published content

The content of this research work was developed in terms of the DFG (Deutsche Forschungsgemeinschaft) project AG 26/18-1.

Parts of this work have already been published and presented by the author or are based on measurement data that arose as part of supervised student work at the Chair of Chemical Process Engineering (Faculty of Biological and Chemical Engineering, TU Dortmund). In particular, the data acquisition and programming are reproductions or are based on the explicitly mentioned contributions by other authors:

[a] M. Hussainy, D. W. Agar, Structural and Operational Optimality of Adsortive Reactors, Chemical Engineering and Technology (2016), Vol. 39 (11), 2135 – 2141.	*Ch. 1 - 5*
[b] M. Hussainy and D.W. Agar, Modelling and Optimisation of the Cyclic Steady State Operation of Adsorptive Reactors, Chinese Journal of Chemical Engineering (2018), Vol. 26 (6), 1321 – 1329.	*Ch. 1 - 5*
[c] M. Hussainy, D.W. Agar, Structural and Operational Optimality of Adsortive Reactors, ESCRE 2015 - European Symposium on Chemical Reaction Engineering, Fürstenfeldbruck, Germany (2015).	*Ch. 3 & 4*
[d] M. Hussainy, D. W. Agar, Multiscale Heat Integration Study of Periodically Operated Adsorptive Reactors, Jahrestreffen Reaktionstechnik, Würzburg (2016).	*Ch. 4: 4.2*
[e] M. Hussainy and D.W. Agar, Modelling and Optimisation of the Cyclic Steady State Operation of Adsorptive Reactors, CAMURE10&ISMR9, Qingdao-China (2017).	*Ch. 4: 4.5*
[f] M. Hussainy, C. Pouwels, and D.W. Agar, Multi-Dimensional and Multi-Scale Modelling and Optimisation of the Functionality Distribution in Adsorptive Reactors, CAMURE10&ISMR9, Qingdao-China (2017).	*Ch. 4: 4.4*

[g] M. Hussainy, O. Scholl, and D.W. Agar, Temporal Profiling Strategies for Enhanced Performance of Adsorptive Reactors, CAMURE10&ISMR9, Qingdao-China (2017). *Ch. 4: 4.3*

[h] B. Arfeen, Functionality distribution and micro-level heat integration strategies in adsorptive reactors, Master Thesis (2015), TU Dortmund, Germany. *Ch. 4: 4.2*

[i] O. Scholl, Improvement of the Cyclic Process Performance by Temporal Profiling Strategies Applied for the Adsorptive Claus Process, Master Thesis (2017), TU Dortmund, Germany. *Ch. 4: 4.3*

[j] C. Pouwels, Studien über strukturelle Maßnahmen zur Optimierung der Performance eines Adsorptivreaktors, Master Thesis (2017), TU Dortmund, Germany. *Ch. 4: 4.4*

[k] M. Kuhnigk, Experimentelle Untersuchungen am adsorptiven Reaktorkonzept für die Claus-Reaktion, Master Thesis (2017), TU Dortmund, Germany. *Ch. 5: 5.1.2.b*

APPENDIX A: MODEL CODE IN ASPEN CUSTOM MODELLER

The mathematical equations of the one-dimensional pseudo-homogeneous dispersed model presented in CHAPTER 2 are implemented in Aspen Custom Modeler. Examples of the written codes in the ACM simulator are given below:

A. One-dimensional non-isothermal adsorptive Claus reactor

```
Model ADR2D

//-------------------------------------------------------------------------------------------------------------
//================================BEGIN SIMULATION=======================================================//
//-------------------------------------------------------------------------------------------------------------

//========================= Defining ComponentLists =========================//
  CompAll   as StringSet(["H2S","O2S","S8","H2O","N2"]);
  CompR     as StringSet(["H2S","O2S","S8","H2O"]);//Reacting Species List
  CompRnA   as StringSet(["H2S","O2S","S8"]);//Reacting Not Adsorbing Species List
  CompA     as StringSet(["H2O"]);// Adsorbing Species List
  SumStoiC  as RealParameter(Description:"Sum of Stoichiometric Coefficients",Value:-5/8);

//========================= Defining Geometric Parameters Reactor Specific ========================//
  Length    as length       (description:"Reactor Length - Axial Co-ordinate [m]", value:1, spec:Fixed);
  Diameter  as length       (description:"Reactor Diameter - Radial Co-ordinate [m]", value:0.06, spec:Fixed);
  pi        as RealParameter(value:355/113);
  Acs       as area         (description:"Cross Sectional Area of the reactor-Normal to the flow [m²]");
  Dp        as length       (description:"Average Particle Diameter [m]", value:3.6E-3, spec:Fixed, scale:1e-3);

//========================= Numerical Handle For Geometry =========================//
  NoL       as IntegerParameter (Description:"Number Of Axial Nodes", 50);
  NoR       as IntegerParameter (Description:"Number Of Radial Nodes", 5);

//========================= Declaring Domain =========================//
  Axial as    LengthDomain(Length: Length, DiscretizationMethod:"BFD1", HighestOrderDerivative:2, SpacingPreference:(Length/NoL), Description:"Axis Of The Reactor");

//========================= Variable Decleration - Phase Ratios For Catalyst/Adsorbent/Gas ========================//
  eps       as  Voidage_Fraction (description:"Reactor Porosity-Catalyst Specific[-]", value:0.36, spec:Fixed);//
  Phi ([0:Axial.EndNode])       as  Voidage_Fraction (/*spec: fixed,*/ description:"Adsorbent Fraction in the solid Phase[-]");//,value:0.43, spec:free
phi1 as Voidage_Fraction (description:"Adsorbent Fraction in the first section of the solid Phase[-]", spec:Fixed, upper:1, lower: 0);
phi2 as Voidage_Fraction (description:"Adsorbent Fraction in the second section of the solid Phase[-]", spec:Fixed, upper:1, lower:0);
phi3 as Voidage_Fraction (description:"Adsorbent Fraction in the third section of the solid Phase[-]", spec:Fixed, upper:1, lower:0);

  f1, f2 as  RealVariable;
  f3([0:Axial.EndNode]) , Fc_cb([0:Axial.EndNode]) , fc_eb([0:Axial.EndNode]) , Fa_cb([0:Axial.EndNode]) , Fa_eb([0:Axial.EndNode])  as  RealVariable;
  //..... Calculating Constant Value Phase Ratios.....
```

```
//======================== Declaring Time-Invarient Physical Properties ========================//
  Phi([0:5])=0;
  Phi([6:45])=1;
  Phi([46:50])=0;



//BY Trapezoidal Function
/*
a as realvariable(15.5,Fixed);
b as realvariable(a+1e-10,Fixed);
c as realvariable(41,Fixed);
d as realvariable(c+1e-10,Fixed);
For X in [0:Axial.EndNode] do;
Phi([X])=Max(Min(((X-a)/(b-a)),1,((d-X)/(d-c))),0);
endfor
*/
  f1=1-eps; f2=f1/eps;
  For X in [0:Axial.EndNode]  do
 f3(X)=1-phi(X);
 Fc_cb(X)=f3(X)*f2;
 Fc_eb(X)=f1*f3(X);
 Fa_cb(X)=phi(X)*f2;
 Fa_eb(X)=phi(X)*f1;
  endFor
  Acs=(pi*Diameter*Diameter)/4;



//======================== Declaring Time-Invarient Properties ========================//
  Had       as RealVariable (description:"Heat of Adsorption[kJ/kmol]", value:-57950/*30500*/*1, spec
:Fixed);//
```

```
71    Hrx       as RealVariable (description:"Heat of Reaction[kJ/kmol]", value:-108600*1, spec:Fixed);//
72    Rho_s     as RealVariable(description:"solid phase density[kg/m³]", value:2400, spec:fixed);
73    Cp_S      as RealVariable (description:"Catalyst Mass Heat Capacity [kJ/kg/K]", value:1.050, spec:
      Fixed);
74    //Cond_s    as RealVariable(description:"Catalyst Mass Heat Capacity [kW/m/K]", value:0.004,
      spec:Fixed);// to be fixed by Rayleigh's Approximation
75
76    //========================= Aspen-Plus Physical Properties =========================//
77    Viscosity([0:Axial.EndNode])                as visc_vap (description:"Gas Phase Viscosity [cP]");
78    Visc_Bar([0:Axial.EndNode])                 as Viscosity_BarS (description:"Gas Phase Viscosity
      [Bar-s]");
79    D_ax ([0:Axial.EndNode])                    as Dispersion (description:"Axial Dispersion[m²/s]");
80    Rho_g([0:Axial.EndNode])                    as dens_mass_vap(description:"Mass Density Of the gas
      phase [kg/m³]");
81    Mw([0:Axial.EndNode])                       as molweight (description:"Molecular Weight Of the gas
      Phase[kg/kmol]");
82    Cp_Vap_Molar([0:Axial.EndNode])             as cp_mol_vap(description:"Vapor Heat
      Capacity[kJ/kmol/K]");// has to be divided by Mw to get Cp_Vap_Mass
83    Cp_Vap_Mass([0:Axial.EndNode])              as RealVariable(description:"Vapor Heat
      Capacity[kJ/kg/K]");
84    Cond_g_W([0:Axial.EndNode])                 as cond_vap (description:"Vapor Phase
      Conductivity[W/m/K]");
85    Cond_g([0:Axial.EndNode])                   as RealVariable (description:"Vapor Phase
      Conductivity[kW/m/K]");
86    D_mol(CompAll,CompAll)([0:Axial.EndNode]) as  hidden Diffus_vap;
87    D_mix([0:Axial.EndNode]) as  RealVariable;
88    D_mi(CompAll)([0:Axial.EndNode]) as hidden Diffusivity;
89    //========================= Declaring Distributed Variables =========================//
90    Ci(CompAll)                     as Distribution1D (XDomain is Axial,Description:"Gas Phase Conc
      [kmol/m³]") of conc_mole;
91    CFlux(CompAll)                  as Distribution1D (XDomain is Axial,Description:"Gas Phase Conc
      [kmol/m²/s]") of RealVariable;
92    u                               as Distribution1D (XDomain is Axial,Description:"Gas Phase Velocity
      [m/s]") of Velocity;
93    Pt                              as Distribution1D (XDomain is Axial,Description:"Gas Phase Pressure
      [Bar]") of Pressure;
94    Tk                              as Distribution1D (XDomain is Axial,Description:"Gas Phase
      Temperature") of Temperature_abs;
95    Ts                              as Distribution1D (XDomain is Axial,Description:"Solid Phase
      Temperature") of Temperature_abs;
96    Yi(CompAll)([0:Axial.EndNode])  as molefraction (description:"Component Mole Fractions");
97    Ct([0:Axial.EndNode])           as conc_mole;
98    Tc([0:Axial.EndNode])           as Temperature(Free);
99    //========================= Sub-Model Decleration =========================//
100   Reac([0:Axial.EndNode]) as Reaction;
101   Ads([0:Axial.EndNode])  as Adsorption;
102   TP([0:Axial.EndNode])   as TransPortProperties;
103
104    //========================= Sub-Model Connections =========================//
105    For X in [0:Axial.Endnode] do
106     Reac(X).Tk=Ts(X);Reac(X).Pt=Pt(X);Reac(X).Yi(CompAll)=Yi(CompAll)(X);Reac(X).Phi=Phi(X);
107     Ads(X).Tk=Ts(X);Ads(X).Pt=Pt(X);Ads(X).Yi(CompAll)=Yi(CompAll)(X);Ads(X).D_mi=D_mi("H2O")(X);Ads(X
      ).Phi=Phi(X);
108     TP(X).u=u(X);TP(X).D_mix=D_mix(X);TP(X).Visc_Bar=Visc_Bar(X);TP(X).rho_g=rho_g(X);TP(X).Cp_Vap_Mass
      =Cp_Vap_Mass(X);TP(X).Cond_g=Cond_g(X);TP(X).Eps=Eps;TP(X).Phi=Phi(X);
109   ENDFOR
110
111
112   For X in [0:Axial.Endnode] do
113     Call (Viscosity(X)) = pVisc_Vap(Tc(X), Pt(X), Yi(CompAll,X)) ;//[cP]
114     Visc_Bar(X)  =Viscosity(X)*(1e-8);//[Bar.s]
115     Call (D_mol(CompAll)(CompAll)(X)) = pDiffus_bin_Vap(Tc(X), Pt(X), Yi(CompAll,X));//[cm²/s]
116     D_mix(X)=SIGMA(Yi(CompAll,X)*(D_mol(CompAll)(CompAll)(X)))/10000;// [m²/s]. D_mol is in
      [cm²/s]---> 1[m²/s]=10000[cm²/s]
117     Call (Rho_g(X)) = pDens_Mass_Vap(Tc(X), Pt(X), Yi(CompAll,X));//[kg/m³]
118     Call (Mw(X)) = pMolweight(Yi(CompAll)(X));//[kg/kmol]
119     Call (Cp_Vap_Molar(X)) = pCp_Mol_Vap(Tc(X),Pt(X),Yi(CompAll)(X));//[kJ/kmol/K]
120     Cp_Vap_Mass(X)=Cp_Vap_Molar(X)/Mw(X);//[kJ/kg/K]
121     Call (Cond_g_W(X)) = pCond_Vap(Tc(X),Pt(X),Yi(CompAll)(X));//[W/m/K]
122     Cond_g(X)=Cond_g_W(X)/1000;//[kW/m/K]
123     Call (D_mi(CompAll)(X)) = pDiffus_Vap(Tc(X), Pt(X), Yi(CompAll)(X));
124   ENDFOR
125
126
127   //========================= Formulating Distributed variables =========================//
128   For X in [0:Axial.EndNode] do
129     Ct(X)=SIGMA(Ci(CompAll)(X));
130     CFlux(CompAll)(X)= Ci(CompAll)(X)*u(X);
131     Yi(CompAll)(X)=Ci(CompAll)(X)/(Ct(X)+1e-5);
132     Tc(X)=Tk(X)-273.15;
133   EndFor
134
135    //========================= PDE'S =========================//
136    // Component Balances:
137    For X in [Axial.Interior] do
```

```
138     $Ci(CompAll)(X)=  -CFlux(CompAll)(X).ddX  +   TP(X).D_ax*Ci(CompAll)(X).d2dX2 +  Fc_cb(X)*Reac(X).
    Rx(CompAll) + Fa_cb(X)*Ads(X).Rad(CompAll);
139    EndFor
140     //Energy Balance
141
142   For X in ([1:5]) do
143     ((1-eps)*rho_s*cp_s)*$Ts(X)            = + TP(X).Cond_s*Ts(X).d2dX2
144                                              - (1-eps)*TP(X).h_fs*(6/Dp)*(Ts(X)-Tk(X))
145                                              + (-Hrx*Fc_eb(X)*Reac(X).Rate + Had*Fa_eb(X)*Ads(X).Rad(
    "H2O"));
146
147   EndFor
148
149   For X in ([1:5]) do
150     (eps*Rho_g(X)*Cp_Vap_Mass(X))*$Tk(X)    = - eps*Rho_g(X)*Cp_Vap_Mass(X)*u(X)*Tk(X).ddX
151                                               + TP(X).L_ax*Tk(X).d2dX2
152                                               + (1-eps)*TP(X).h_fs*(6/Dp)*(Ts(X)-Tk(X));
153   EndFor
154
155   For X in ([6:45]) do
156     ((1-eps)*rho_s*cp_s)*$Ts(X)            = + TP(X).Cond_s*Ts(X).d2dX2*0
157                                              - (1-eps)*TP(X).h_fs*(6/Dp)*(Ts(X)-Tk(X))*0
158                                              + (-Hrx*Fc_eb(X)*Reac(X).Rate + Had*Fa_eb(X)*Ads(X).Rad(
    "H2O"))*0;
159
160   EndFor
161
162   For X in ([6:45]) do
163     (eps*Rho_g(X)*Cp_Vap_Mass(X))*$Tk(X)    = - eps*Rho_g(X)*Cp_Vap_Mass(X)*u(X)*Tk(X).ddX*0
164                                               + TP(X).L_ax*Tk(X).d2dX2*0
165                                               + (1-eps)*TP(X).h_fs*(6/Dp)*(Ts(X)-Tk(X))*0;
166   EndFor
167
168   For X in ([46:49]) do
169     ((1-eps)*rho_s*cp_s)*$Ts(X)            = + TP(X).Cond_s*Ts(X).d2dX2
170                                              - (1-eps)*TP(X).h_fs*(6/Dp)*(Ts(X)-Tk(X))
171                                              + (-Hrx*Fc_eb(X)*Reac(X).Rate + Had*Fa_eb(X)*Ads(X).Rad(
    "H2O"));
172
173   EndFor
174
175   For X in ([46:49]) do
176     (eps*Rho_g(X)*Cp_Vap_Mass(X))*$Tk(X)    = - eps*Rho_g(X)*Cp_Vap_Mass(X)*u(X)*Tk(X).ddX
177                                               + TP(X).L_ax*Tk(X).d2dX2
178                                               + (1-eps)*TP(X).h_fs*(6/Dp)*(Ts(X)-Tk(X));
179   EndFor
180
181    // Overall Balance(Velocity Calculation) and Darcy's Law(Pressure Drop):
182   For X in [Axial.Interior+Axial.EndNode] do
183     //Pt(X)=1;
184     Pt(X).ddX=-(180*Visc Bar(X))*((f1^2)/((eps^3)*(Dp^2)))*u(X);
185
    //-Pt(X).ddX=(150*Visc_Bar(X)*(1-eps)^2*u(X)/(Dp^2*eps^3)+1.75*rho_g(X)*(1-eps)*abs(u(X))*u(X)/(Dp*ep
    ^3))/*/100000*/;
186     Ct(X)*u(X).ddX= sumstoic*Fc_cb(X)*Reac(X).Rate  -Fa_cb(X)*Ads(X).Rad("H2O");
187   EndFor
188
189
190   //========================= Fixed Initial Conditions ========================//
191   Ci(CompR)(Axial.Interior):1e-10,Initial;
192   Ci("N2")(Axial.Interior):0.023295947,Initial;
193   //Tk(Axial.Interior):523.15,Initial;
194   //Ts(Axial.Interior):523.15,Initial;
195
196   Tk([1:5]):513,Initial;
197   Tk([6:45]):473,Initial;
198   Tk([46:49]):533,Initial;
199   Ts([1:5]):513,Initial;
200   Ts([6:45]):473,Initial;
201   Ts([46:49]):533,Initial;
202   //========================= Boundary Conditions ========================//
203   //Ci("H2S")(0)=0.002329595; Ci("O2S")(0)=0.001164797; Ci("S8")(0)=0;  Ci("H2O")(0)=0;
    Ci("N2")(0)=0.019801555;
204
205   -TP(0).D_ax*Ci("H2S")(0).ddx=u(0)*(0.002329595-Ci("H2S")(0));
206   -TP(0).D_ax*Ci("O2S")(0).ddx=u(0)*(0.001164797-Ci("O2S")(0));
207   -TP(0).D_ax*Ci("S8")(0).ddx=u(0)*(0.000000001-Ci("S8")(0));
208   -TP(0).D_ax*Ci("H2O")(0).ddx=u(0)*(0.000000001-Ci("H2O")(0));
209   -TP(0).D_ax*Ci("N2")(0).ddx=u(0)*(0.019801555-Ci("N2")(0));
210
211   Ci(CompAll)(Axial.EndNode).ddX=0;
212
213   Pt(0)=Ct(0)*(0.0814)*Tk(0);
214   u(0)=0.21;
215
```

```
//Tk(0)=273.15+250;
-TP(0).L_ax*Tk(0).ddX=rho_g(0)*Cp_Vap_Mass(0)*u(0)*(523-Tk(0));

Tk(Axial.EndNode).ddX=0;

Ts(0)=273.15+250;
Ts(Axial.EndNode).ddX=0;

//========================= Performance Indices =========================//
//Variables to plot
BreakH2o as RealVariable;
equllibrium([0:Axial.EndNode]) as Realvariable;
throughput as realvariable;
throughput:0, initial;
$throughput=Ci("S8")(Axial.EndNode)*Acs*u(axial.endnode);
//throughput=SIGMA(Ci("S8")(Axial.EndNode)*Time*Acs*u(axial.endnode)/eps);
purity as realvariable;
purity=Sigma((u(Axial.Endnode)*Ci("S8")(Axial.EndNode)*Time))/Sigma((u(Axial.Endnode)*(Ct(
Axial.EndNode)-Ci("N2")(Axial.EndNode)-Ci("H2O")(Axial.EndNode))*(Time+1e-5)));
  BreakH2o=Yi("H2O")(Axial.EndNode);
  XX as realvariable;
  XD([0:Axial.EndNode]) as Realvariable;
 //
XX=1-((Yi("H2S")(Axial.EndNode))*u(Axial.EndNode)*Ct(Axial.Endnode)/Tk(Axial.EndNode))/(Yi("H2S")(0)*
(0)*Ct(0)/Tk(0));
  XX=1-((Ci("H2S")(Axial.EndNode))*u(Axial.EndNode))/(Ci("H2S")(0)*u(0));
For X in [0:Axial.Endnode] do
  XD(X)=1-((Ci("H2S")(X))*u(x))/(Ci("H2S")(0)*u(0));
  equllibrium(X)=((Yi("H2O")(X))^2)*((Yi("S8")(X))^(3/8))/(((Yi("H2S")(X))^2)*(Yi("O2S")(X)));
EndFor


  XX_obj as RealVariable;
$XX_obj=XX-0.995;
XX_obj:0.005, Initial;

//-------------------------------------------------------------------------------------------------------
//=================================END SIMULATION=====================================================//
//-------------------------------------------------------------------------------------------------------

End
```

```
Model Reaction

//Connection Variables with ADR2D Model
  CompAll               as StringSet(["H2S","O2S","S8","H2O","N2"]);
  Pt                    as pressure(description:"Reactor Pressure[bar]");
  Tk                    as temperature_abs ;
  Yi(CompAll)           as molefraction (description:"Component Mole Fractions");
  Phi as RealVariable;

// Reaction Specific Variables
  Ea1         as hidden Activation_Energy (description:"Forward Reaction[J/mol]", value:49980, spec:
Fixed);
  Ea2         as hidden Activation_Energy (description:"Backward Reaction[J/mol]", value:86601, spec:
Fixed);
  rho_cat     as hidden dens_mass         (description:"Catalyst Density[kg/m³]", value:3600, spec:
Fixed);
  k1          as RealVariable;
  k2          as RealVariable;
  Approach_to_eq  as RealVariable;

// Reaction Rate Computation
  Rx(CompAll)     as  Reaction_Rate(Description:"Rate [kmol/m³/s]");
  Rate            as  Reaction_Rate(Description:"Rate [kmol/m³/s]");

// Claus Reaction Rate
/*k1=(5.292e-3)*EXP((-Ea1)/(8.314*Tk));
k2=(1.252)*EXP((-Ea2)/(8.314*Tk));//(1e-3)*
  //IF ((Yi("H2S")>=0) and (Yi("O2S")>=0) and (Yi("H2O")>=0) ) THEN//and (Phi==0)

Rate=(k1*((Yi("H2S")*Pt*1000)^0.95)*((Yi("O2S")*Pt*1000)^0.22)-k2*(Yi("H2O")*Pt*1000))*rho_cat;//[mol
(kg-catalyst)/s]*[kg-catalyst*m³]*[m³/mol]=[kmol/m³/s]
  //ELSE
    //Rate=0;
  //ENDIF
  */

  k1=(/*5.292e-3*/17.12458)*EXP((-Ea1)/(8.314*Tk));
k2=(/*1.252*/1168.434385)*EXP((-Ea2)/(8.314*Tk));//(1e-3)*
  //IF ((Yi("H2S")>=0) and (Yi("O2S")>=0) and (Yi("H2O")>=0) ) THEN//and (Phi==0)
    Rate=(k1*((Yi("H2S")*Pt/**1000*/)^0.95)*((Yi("O2S")*Pt/**1000*/)^0.22)-k2*(Yi("H2O")*Pt/**1000*/
))*rho_cat;//[mol/(kg-catalyst)/s]*[kg-catalyst*m³]*[m³/mol]=[kmol/m³/s]
  //ELSE
    //Rate=0;
  //ENDIF

  (k2*(Yi("H2O")*Pt)) * Approach_to_eq = (k1*((Yi("H2S")*Pt)^0.95)*((Yi("O2S")*Pt)^0.22));

// Component Specific Reaction Rates: 2H2S + SO2 + N2 <==> (3/8)S8 + 2H2O + N2
  Rx("H2S") =   -2*Rate;// Consumption
  Rx("O2S") =     -Rate;// Consumption
  Rx("H2O") =    2*Rate;// Generation
  Rx("S8")  =(3/8)*Rate;// Generation
  Rx("N2")  =0;

End
```

```
Model Adsorption



//Connection Variables with ADR2D Model
  CompAll       as StringSet(["H2S","O2S","S8","H2O","N2"]);
  Yi(CompAll)   as molefraction;
  Pt            as pressure       (description:"Reactor Pressure[bar]");
  Tk            as temperature abs(description:"Temperature[K]") ;
  Q             as conc mole      (description:"Adsorbate Concentartion in the bed[kmol/m³]");
  Pw            as Pressure       (description:"Partial Pressure Of Water[Bar]");
  Phi as RealVariable;

// Adsorption Specific Parameters
  rho_ad    as hidden dens_mass           (description:"Catalyst Density[kg/m³]", value:1199.3, spec:
Fixed);
  Binf      as RealVariable               (description:"Maximum Adsorption Affinity [K^0.5/bar]",
value:5.3126E-5, spec:Fixed,scale:1e-8);//--> 1[Pa]=0.01[mbar]
  T0        as  hidden  temperature abs   (description:"Adsorption Reference Temperature [K]", value:
300, spec:Fixed);
  q0w       as conc_mole                  (description:"Loading Correction [kmol/m³-Adsorbent]",
value:12.886, spec:Fixed);
  gamma     as RealVariable               (description:"Adsorption Parameter [-]", value:23.235, spec
:Fixed);
  delta     as hidden  RealVariable       (description:"Adsorption Parameter [-]", value:0.68792,
spec:Fixed);
  Dp        as hidden  length             (description:"Average Particle Diameter [m]", value:3.6E-3,
 spec:Fixed);
  Rg        as hidden  RealParameter      (description:"Regnault's Constant [J/K/mol]",Value:
8.3144621);
  r_pore as realvariable(3e-8,Fixed);//cm
  Mi as realvariable(18,fixed);
  D_k as realvariable;
  D_mi as RealVariable;
  Tau as RealVariable(1,Fixed);
  D_ad as realvariable;

// Adsorption Specific Variables
  Kldf      as RealVariable   (description:"Linear Driving Force Coefficient[1/s]");// has to be
changed later on
  qsw       as conc_mole      (description:"Saturation Loading Capacity [kmol/m³-Adsorbent]");
  B         as RealVariable   (description:"Adsorption Affinity[1/bar]",scale:0.1);
  Qstar     as conc_mole      (description:"Equillibrium Loading  [kmol/m³-Adsorbent]");
  qstr      as conc_mole      (description:"Equillibrium Loading  [kmol/m³-Adsorbent]");


// Adsorption Rate Computation ===> Equivalent To Reaction Rate
  Rad(CompAll)   as  Reaction Rate(Description:"Rate [kmol/m³/s]");
  RateAd         as  Reaction Rate(Description:"Rate [kmol/m³/s]");
  A              as RealVariable;
  F_dash         as RealVariable;


//Declaring Solvent Pore Diffusion Coefficient Variables for Zeolite 3A

D_k=9700*r_pore*sqrt(Tk/Mi);
D_ad*10000=(D_k*D_mi)/(Tau*(D_k+D_mi));

// Adsorption Rate
  Pw=Yi("H2O")*Pt;
  EqAD1:  qsw=q0w*EXP(delta*(1-(Tk/T0))); //[kmol/m³-Adsorbent] -> No change
  EqAD2:  B=(Binf/SQRT(Tk))*EXP(gamma*T0/Tk);// 1/bar  -> No change
  EqAD3:  qstr=B*qsw*Pw/(1+ B*Pw);//[mol/kg-Adsorbent]  -> No change

  EQAD4:  A=B*Pw;//Unitless
  EQAD5:  F_dash =qsw*B*Tk*(8.314e-5)/((1+A)^2);
  EqAD6:  Kldf=(60/(Dp^2)) * D_ad  * (1/(1+(1/0.5)*F_dash ));//[1/s]

  IF (0.9 < Phi ) then
  EqAD7:  $Q =(Kldf*(qstr-Q));//[kmol/(kg-adsorbent)/s]*[kg-adsorbent/m³]=[kmol/m³/s]
  Else
  $Q=0;
  ENDIF
  EqAD8:  RateAd=$Q;
  Rad("H2O")   =   -RateAd;// Consumption
  Rad("H2S")   =   0;// Consumption
  Rad("O2S")   =   0;// Consumption
  Rad("S8")    =   0;// Generation
  Rad("N2")    =   0;

  Q:1e-10,Initial;


End
```

```
Model TransportProperties


//Conection Parameters With The Model ADR2D
u          as velocity(description:"True Velocity");
D_mix as RealVariable;
Rho_g      as dens_mass_vap;
Visc Bar   as Viscosity BarS;
Visc Pas   as Realvariable(description:"Gas Phase Viscosity[kg/m/s]");
Cp Vap Mass        as RealVariable(description:"Mass Heat Capacity of Gas Phase[kJ/kg/K]");
Eps    as Voidage_Fraction;
Phi as Voidage_Fraction;
Cond_Cat   as RealVariable(description:"Catalyst Conductivity[kW/m/K]",0.00574,Fixed);
Cond_Ads   as RealVariable(description:"Adsorbent Conductivity[kW/m/K]",0.00012,Fixed);
Cond_s     as RealVariable;
Cond_g     as RealVariable(description:"Vapor Phase Conductivity[kW/m/K]");
//Declaring Reactor Scale Parameters

Dp         as length       (description:"Average Particle Diameter [m]", value:3.6E-3, spec:Fixed,
scale:1e-3);
D_Reactor as RealVariable (description:"Reactor Diameter [m]", value:0.06, spec:Fixed,scale:1e-2);



N          as RealVariable;
K          as realVariable;
M          as RealVariable(Description:"Der Parameter");
a1         as Realvariable;
a2         as Realvariable;
B          as RealVariable;

Pe_mz      as RealVariable(Description:"Axial Mass Peclet Number");
Pe_hz      as RealVariable(Description:"Axial Heat Peclet Number");
Re_p       as RealVariable(Description:"Particle Reynolds Number");
Sc         as RealVariable(Description:"Schmidt Number");
Pr         as RealVariable(Description:"Prandtl Number");

D_ax       as Dispersion;
L_ax       as Dispersion;
L_ax1      as RealVariable;
L_ax2      as RealVariable;
D_ar       as Realvariable;
L_ar       as Realvariable;
Kwall      as Realvariable;
h_fs       as Realvariable;
Nu         as Realvariable;
Visc_Pas=Visc_Bar*1e5;

//Calculating Solid Phase Conductivity - Rayleigh's Law
if(phi==0) then
  Cond_s=Cond_cat;
  elseif(phi==1) then
    Cond_s=Cond_Ads;
    else
Cond_s/Cond_Cat=1+(3*Phi)/(((Cond_Ads+2*Cond_Cat)/(Cond_Ads-Cond_Cat))-(Phi)+1.569*((Cond_Ads-
Cond_Cat)/(3*Cond_Ads-4*Cond_Cat))*(Phi^(10/3)));
endif
//Calculating DimensionLess Numbers
Pr      = Visc Pas*Cp Vap Mass/(Cond g);
Re_p    = rho_g*Dp*u/Visc_Pas;
Pe_hz   = Re_p*Pr;
Nu      = (2+1.1*(Pr^(1/3))*(Re_p^(0.6)));
// Axial Mass Dispersion Coefficient - D_ax
D_ax    = 20*D_mix + 0.5*u*Dp;

// Axial Heat Dispersion Coefficient - L_ax
L_ax *1000    = ((0.73/(Re_p*Pr))+(0.5/(1+(9.7/(Re_p*Pr)))))*(u*Rho_g*Cp_Vap_Mass*1000*Dp);// because
this has to be kept in [Kw/m/K]

// Fluid To Solid Heat Transfer Coefficient - H_fs
h_fs=(2+1.1*(Pr^(1/3))*(Re_p^(0.6)))*Cond_g/Dp;

End
```

B. One-dimensional isothermal adsorptive Deacon reactor

```
Model ADR2D

//-------------------------------------------------------------------------------------------------------------
//==================================BEGIN SIMULATION=====================================================//
//-------------------------------------------------------------------------------------------------------------

//========================= Defining ComponentLists =========================//
  CompAll   as StringSet(["HCl","O2","Cl2","H2O","N2"]);
  CompR     as StringSet(["HCl","O2","Cl2","H2O"]);//Reacting Species List
  CompRnA   as StringSet(["HCl","O2","Cl2"]);//Reacting Not Adsorbing Species List
  CompA     as StringSet(["H2O"]);// Adsorbing Species List
  SumStoiC  as RealParameter(Description:"Sum of Stoichiometric Coefficients",Value:-1/2);

//========================= Defining Geometric Parameters Reactor Specific ========================//
  Length    as length       (description:"Reactor Length - Axial Co-ordinate [m]", value:1, spec:Fixed);
  Diameter  as length       (description:"Reactor Diameter - Radial Co-ordinate [m]", value:0.06, spec:Fixed);
  pi        as RealParameter(value:355/113);
  Acs       as area         (description:"Cross Sectional Area of the reactor-Normal to the flow [m²]");
  Dp        as length       (description:"Average Particle Diameter [m]", value:3.6E-3, spec:Fixed, scale:1e-3);
  //========================= Numerical Handle For Geometry =========================//
  NoL       as IntegerParameter (Description:"Number Of Axial Nodes", 50);
  NoR       as IntegerParameter (Description:"Number Of Radial Nodes", 5);

//========================= Declaring Domain =========================//
  Axial as    LengthDomain(Length: Length, DiscretizationMethod:"BFD1", HighestOrderDerivative:2, SpacingPreference:(Length/NoL), Description:"Axis Of The Reactor");



//========================= Variable Decleration - Phase Ratios For Catalyst/Adsorbent/Gas ========================//
  eps       as  Voidage Fraction (description:"Reactor Porosity-Catalyst Specific[-]", value:0.36, spec:Fixed);//
  Phi ([0:Axial.EndNode])       as  Voidage_Fraction (/*spec: fixed,*/ description:"Adsorbent Fraction in the solid Phase[-]");//,value:0.43, spec:free
/*phi1 as Voidage_Fraction (description:"Adsorbent Fraction in the first section of the solid Phase[-]", spec:Fixed, upper:1, lower: 0);
phi2 as Voidage_Fraction (description:"Adsorbent Fraction in the second section of the solid Phase[-]", spec:Fixed, upper:1, lower:0);
phi3 as Voidage_Fraction (description:"Adsorbent Fraction in the third section of the solid Phase[-]", spec:Fixed, upper:1, lower:0);*/
//n as Integerparameter(5);
//m as Integerparameter(46);

  Ts ([0:Axial.EndNode]) as Temperature_abs /*(spec: fixed, value: 573)*/;
/*T1 as Temperature_abs (description:"Temperature in the first section of the solid Phase[-]", spec:Fixed, upper:573, lower: 473);
T2 as Temperature_abs (description:"Temperature in the second section of the solid Phase[-]", spec:Fixed, upper:573, lower:473);
T3 as Temperature_abs (description:"Temperature in the third section of the solid Phase[-]", spec:Fixed, upper:573, lower:473);*/
  f1, f2 as  RealVariable;
  f3([0:Axial.EndNode]) , Fc_cb([0:Axial.EndNode]) , fc_eb([0:Axial.EndNode]) , Fa_cb([0:Axial.EndNode]) , Fa_eb([0:Axial.EndNode])  as  RealVariable;
  //............ Calculating Constant Value Phase Ratios.....................

//========================= Declaring Time-Invarient Physical Properties ========================//
Phi([0:5])=0;
  Phi([6:45])=1;
  Phi([46:50])=0;

Ts([0:5])=543;
  Ts([6:45])=473;
  Ts([46:50])=533;

/*
n1 as integerparameter(5);
n2 as integerparameter(50);
  Phi([0:n1])=0;
  //Phi([n1+1:50])=0.96;
  Phi([n1+1:n2])=0.95;
  //Phi([n2+1:50])=0;
//BY Trapezoidal Function
*/

/*
a as realvariable(13,Fixed);
b as realvariable(a+1e-10,Fixed);
c as realvariable(45,Fixed);
```

```
d as realvariable(c+1e-10,Fixed);
For X in [0:Axial.EndNode] do;
Phi([X])=Max(Min(((X-a)/(b-a)),1,((d-X)/(d-c))),0);
endfor
*/
  f1=1-eps; f2=f1/eps;
  For X in [0:Axial.EndNode]  do
 f3(X)=1-phi(X);
 Fc_cb(X)=f3(X)*f2;
 Fc_eb(X)=f1*f3(X);
 Fa_cb(X)=phi(X)*f2;
 Fa_eb(X)=phi(X)*f1;
  endFor
  Acs=(pi*Diameter*Diameter)/4;


//========================= Declaring Time-Invarient Properties =========================//
  Had       as RealVariable (description:"Heat of Adsorption[kJ/kmol]", value:-57950/*30500*/*0, spec:Fixed);//
  Hrx       as RealVariable (description:"Heat of Reaction[kJ/kmol]", value:-57200*0, spec:Fixed);//
  Rho_s     as RealVariable(description:"solid phase density[kg/m³]", value:3800, spec:fixed);
  Cp_S      as RealVariable (description:"Catalyst Mass Heat Capacity [kJ/kg/K]", value:0.519, spec:Fixed);//0.789
  //Cond_s    as RealVariable(description:"Catalyst Mass Heat Capacity [kW/m/K]", value:0.004, spec:Fixed);// to be fixed by Rayleigh's Approximation

//========================= Aspen-Plus Physical Properties =========================//
Viscosity([0:Axial.EndNode])                as visc_vap (description:"Gas Phase Viscosity [cP]");
Visc_Bar([0:Axial.EndNode])                 as ViscosityBarS (description:"Gas Phase Viscosity [Bar-s]");
D_ax ([0:Axial.EndNode])                    as Dispersion (description:"Axial Dispersion[m²/s]");
Rho_g([0:Axial.EndNode])                    as dens_mass_vap(description:"Mass Density Of the gas phase [kg/m³]");
Mw([0:Axial.EndNode])                       as molweight (description:"Molecular Weight Of the gas Phase[kg/kmol]");
Cp_Vap_Molar([0:Axial.EndNode])             as cp_mol_vap(description:"Vapor Heat Capacity[kJ/kmol/K]");// has to be divided by Mw to get Cp_Vap_Mass
Cp_Vap_Mass([0:Axial.EndNode])              as RealVariable(description:"Vapor Heat Capacity[kJ/kg/K]");
Cond_g_W([0:Axial.EndNode])                 as cond_vap (description:"Vapor Phase Conductivity[W/m/K]");
Cond_g([0:Axial.EndNode])                   as RealVariable (description:"Vapor Phase Conductivity[kW/m/K]");
D_mol(CompAll,CompAll)([0:Axial.EndNode]) as  hidden Diffus_vap(upper:2);
D_mix([0:Axial.EndNode]) as  RealVariable;
D_mi(CompAll)([0:Axial.EndNode]) as hidden Diffusivity(upper:2);
//========================= Declaring Distributed Variables =========================//
Ci(CompAll)                     as Distribution1D (XDomain is Axial,Description:"Gas Phase Conc [kmol/m³]") of conc_mole;
CFlux(CompAll)                  as Distribution1D (XDomain is Axial,Description:"Gas Phase Conc [kmol/m²/s]") of RealVariable;
u                               as Distribution1D (XDomain is Axial,Description:"Gas Phase Velocity [m/s]") of Velocity;
Pt                              as Distribution1D (XDomain is Axial,Description:"Gas Phase Pressure [Bar]") of Pressure;
//Tk                              as Distribution1D (XDomain is Axial,Description:"Gas Phase Temperature") of Temperature_abs;
//Ts                              as Distribution1D (XDomain is Axial,Description:"Solid Phase Temperature") of Temperature_abs;
Yi(CompAll)([0:Axial.EndNode])  as molefraction (description:"Component Mole Fractions");
Ct([0:Axial.EndNode])           as conc_mole;
Tc([0:Axial.EndNode])           as Temperature(Free);
//========================= Sub-Model Decleration =========================//
Reac([0:Axial.EndNode]) as Reaction;
Ads([0:Axial.EndNode])  as Adsorption;
TP([0:Axial.EndNode])   as TransPortProperties;

 //========================= Sub-Model Connections =========================//
 For X in [0:Axial.Endnode] do
  Reac(X).Tk=Ts(X);Reac(X).Pt=Pt(X);Reac(X).Yi(CompAll)=Yi(CompAll)(X);Reac(X).Phi=Phi(X);
  Ads(X).Tk=Ts(X);Ads(X).Pt=Pt(X);Ads(X).Yi(CompAll)=Yi(CompAll)(X);Ads(X).D_mi=D_mi("H2O")(X);Ads(X).Phi=Phi(X);
  TP(X).u=u(X);TP(X).D_mix=D_mix(X);TP(X).Visc_Bar=Visc_Bar(X);TP(X).rho_g=rho_g(X);TP(X).Cp_Vap_Mass=Cp_Vap_Mass(X);TP(X).Cond_g=Cond_g(X);TP(X).Eps=Eps;TP(X).Phi=Phi(X);
ENDFOR


For X in [0:Axial.Endnode] do
  Call (Viscosity(X)) = pVisc_Vap(Tc(X), Pt(X), Yi(CompAll,X)) ;//[cP]
  Visc_Bar(X)  =Viscosity(X)*(1e-8);//[Bar.s]
  Call (D_mol(CompAll)(CompAll)(X)) = pDiffus_bin_Vap(Tc(X), Pt(X), Yi(CompAll,X));//[cm²/s]
  D_mix(X)=SIGMA(Yi(CompAll,X)*(D_mol(CompAll)(CompAll)(X)))/10000;// [m²/s]. D_mol is in [cm²/s]---> 1[m²/s]=10000[cm²/s]
  Call (Rho_g(X)) = pDens_Mass_Vap(Tc(X), Pt(X), Yi(CompAll,X));//[kg/m³]
  Call (Mw(X)) = pMolweight(Yi(CompAll)(X));//[kg/kmol]
```

```
135     Call (Cp_Vap_Molar(X)) = pCp_Mol_Vap(Tc(X),Pt(X),Yi(CompAll)(X));//[kJ/kmol/K]
136     Cp_Vap_Mass(X)=Cp_Vap_Molar(X)/Mw(X);//[kJ/kg/K]
137     Call (Cond_g_W(X)) = pCond_Vap(Tc(X),Pt(X),Yi(CompAll)(X));//[W/m/K]
138     Cond_g(X)=Cond_g_W(X)/1000;//[kW/m/K]
139     Call (D_mi(CompAll)(X)) = pDiffus_Vap(Tc(X), Pt(X), Yi(CompAll)(X));
140   ENDFOR
141
142
143   //========================= Formulating Distributed variables =========================//
144   For X in [0:Axial.EndNode] do
145     Ct(X)=SIGMA(Ci(CompAll)(X));
146     CFlux(CompAll)(X)= Ci(CompAll)(X)*u(X);
147     Yi(CompAll)(X)=Ci(CompAll)(X)/(Ct(X)+1e-5);
148     Tc(X)=Ts(X)-273.15;
149   EndFor
150
151    //========================= PDE'S =========================//
152    // Component Balances:
153    For X in [Axial.Interior] do
154     $Ci(CompAll)(X)=  -CFlux(CompAll)(X).ddX  +   TP(X).D ax*Ci(CompAll)(X).d2dX2 +  Fc cb(X)*Reac(X).
      Rx(CompAll) + Fa_cb(X)*Ads(X).Rad(CompAll);
155    EndFor
156     //Energy Balance
157    /*
158   For X in [Axial.Interior] do
159     ((1-eps)*rho_s*cp_s)*$Ts(X)            = + TP(X).Cond_s*Ts(X).d2dX2*0
160                                              - (1-eps)*TP(X).h_fs*(6/Dp)*(Ts(X)-Tk(X))*0
161                                              + (-Hrx*Fc eb(X)*Reac(X).Rate +
      Had*Fa eb(X)*Ads(X).Rad("H2O"));
162
163   EndFor
164
165   For X in [Axial.Interior] do
166     (eps*Rho_g(X)*Cp_Vap_Mass(X))*$Tk(X)    = - eps*Rho_g(X)*Cp_Vap_Mass(X)*u(X)*Tk(X).ddX*0
167                                               + TP(X).L_ax*Tk(X).d2dX2*0
168                                               + (1-eps)*TP(X).h_fs*(6/Dp)*(Ts(X)-Tk(X))*0;
169   EndFor
170   */
171    // Overall Balance(Velocity Calculation) and Darcy's Law(Pressure Drop):
172   For X in [Axial.Interior+Axial.EndNode] do
173     //Pt(X)=1.01325;
174     Pt(X).ddX=-(180*Visc_Bar(X))*((fl^2)/((eps^3)*(Dp^2)))*u(X);
175
      //-Pt(X).ddX=(150*Visc_Bar(X)*(1-eps)^2*u(X)/(Dp^2*eps^3)+1.75*rho_g(X)*(1-eps)*abs(u(X))*u(X)/(Dp*ep
      ^3))/*/100000*/;
176     Ct(X)*u(X).ddX= sumstoic*Fc_cb(X)*Reac(X).Rate  -Fa_cb(X)*Ads(X).Rad("H2O");
177   EndFor
178
179
180   //========================= Fixed Initial Conditions =========================//
181   Ci(CompR)(Axial.Interior):1e-10,Initial;
182   Ci("N2")(Axial.Interior):0.023295947,Initial;
183   /*
184   Tk([1:5]):573.15,Initial;
185   Tk([6:49]):503.15,Initial;
186   Ts([1:5]):573.15,Initial;
187   Ts([6:49]):503.15,Initial;
188
189
190
191   Tk([1:5]):573.15,Initial;
192   Tk([6:45]):473.15,Initial;
193   Tk([46:49]):573.15,Initial;
194   Ts([1:5]):573.15,Initial;
195   Ts([6:45]):473.15,Initial;
196   Ts([46:49]):573.15,Initial;
197   */
198
199   //========================= Boundary Conditions =========================//
200   //Ci("HCl")(0)=0.002329595; Ci("O2")(0)=0.002329595; Ci("Cl2")(0)=0;  Ci("H2O")(0)=0;
      Ci("N2")(0)=0.019801555;
201   -TP(0).D_ax*Ci("HCl")(0).ddx=u(0)*(0.002329595-Ci("HCl")(0));
202   -TP(0).D_ax*Ci("O2")(0).ddx=u(0)*(0.002329595-Ci("O2")(0));
203   -TP(0).D_ax*Ci("Cl2")(0).ddx=u(0)*(0.000000001-Ci("Cl2")(0));
204   -TP(0).D_ax*Ci("H2O")(0).ddx=u(0)*(0.000000001-Ci("H2O")(0));
205   -TP(0).D_ax*Ci("N2")(0).ddx=u(0)*(0.019801555-Ci("N2")(0));
206
207   Ci(CompAll)(Axial.EndNode).ddX=0;
208   Pt(0)=Ct(0)*(0.0814)*Ts(0);
209   u(0)=0.2;
210   /*Tk(0)=273.15+300;
211   Tk(Axial.EndNode).ddX=0;
212   Ts(0)=273.15+300;
213   Ts(Axial.EndNode).ddX=0;*/
214
```

```
//========================= Performance Indices =========================//
//Variables to plot
BreakH2o as RealVariable;
Impurity as realvariable;
throughput as realvariable;
throughput:0, initial;
$throughput=Ci("Cl2")(Axial.EndNode)*Acs*u(axial.endnode);

equllibrium([0:Axial.EndNode]) as Realvariable;
  BreakH2o=Yi("H2O")(Axial.EndNode);
  XX as realvariable;
  XD([0:Axial.EndNode]) as Realvariable;
  //XX=1-((Ci("HCl")(Axial.EndNode))*u(Axial.EndNode))/(Ci("HCl")(0)*u(0));
  //XX=1-((Ci("HCl")(Axial.EndNode))*1)/(Ci("HCl")(0)*1);
  XX=XD(Axial.EndNode);
  Impurity=((Yi("HCl")(Axial.EndNode)))/((Yi("HCl")(Axial.EndNode))+(Yi("Cl2")(Axial.EndNode)));
For X in [0:Axial.Endnode] do
  //XD(X)=1-((Yi("HCl")(X))*u(X))/(Yi("HCl")(0)*u(0));
  //XD(X)=1-((Ci("HCl")(X))*1)/(Ci("HCl")(0)*1);
  XD(X)=1-((Ci("HCl")(X))*u(X))/(Ci("HCl")(0)*u(0));
  equllibrium(X)=((Yi("H2O")(X))^2)*((Yi("Cl2")(X))^(2))/(((Yi("HCl")(X))^4)*(Yi("O2")(X)));

EndFor

 XX_obj as RealVariable;
$XX_obj=XX-0.95;
XX_obj:0.05, Initial;


//-------------------------------------------------------------------------------------------------------
//=================================END SIMULATION=====================================================//
//-------------------------------------------------------------------------------------------------------

End
```

```
Model Reaction

//Connection Variables with ADR2D Model
  CompAll   as StringSet(["HCl","O2","Cl2","H2O","N2"]);
  Pt                    as pressure(description:"Reactor Pressure[bar]");
  Tk                    as temperature_abs ;
  Yi(CompAll)           as molefraction (description:"Component Mole Fractions");
  Phi as Realvariable;
// Reaction Specific Variables
  K_eq as realVariable;  K_ as realVariable;  K as realVariable;

  Ea1         as hidden Activation_Energy (description:"Forward Reaction[J/mol]", value:91552, spec:Fixed);//91552
  Ea2         as hidden Activation_Energy (description:"Backward Reaction[J/mol]", value:52331, spec:Fixed);//52331

  K01         as hidden RealVariable (description:"Forward Reaction[J/mol]", value:2.4768e5, spec:Fixed);
  K02         as hidden RealVariable (description:"Backward Reaction[J/mol]", value:0.0020075, spec:Fixed);

  rho_cat     as hidden dens_mass         (description:"Catalyst Density[kg/m³]", value:6950, spec:Fixed);
  k1          as RealVariable;
  k2          as RealVariable;
  Approach_to_eq  as RealVariable;

// Reaction Rate Computation
  Rx(CompAll)     as  Reaction Rate(Description:"Rate [kmol/m³/s]");
  Rate            as  Reaction_Rate(Description:"Rate [kmol/m³/s]");


  Coefficient_K:      K_=(10^(((5881.7/Tk)-(0.93035*LOG10(Tk))+((1.3704/10000)*Tk)-((1.758/100000000)*(Tk^2))-(4.1744))));
                      K=sqrt(K_/1);
                    K_eq=K;


// Deacon Reaction Rate
k1=(K01)*EXP((-Ea1)/(8.314*Tk));
k2=(K02)*EXP((Ea2)/(8.314*Tk));//(1e-3)*

  IF ((Yi("HCl")>=0) and (Yi("O2")>=0) ) THEN//and (Yi("H2O")>=0)
    Rate/(rho_cat*k1)=(((Yi("HCl")*Pt*1)^0.5)*((Yi("O2")*Pt*1))-((Yi("H2O")*Pt*1)*(Yi("Cl2")*Pt*1)/SQRT(K_eq*((Yi("O2")*Pt*1)^-0.5)*((Yi("HCl")*Pt*1)^1.5))))/(1+0.5*k2*(Yi("H2O")*Pt*1 + Yi("Cl2")*Pt*1));
  ELSE
    Rate=0;
  ENDIF

  (k1*(Yi("H2O")*Pt*1)*(Yi("Cl2")*Pt*1)/SQRT(K_eq*((Yi("O2")*Pt*1)^-0.5)*((Yi("HCl")*Pt*1)^1.5)))/(1+0.5*k2*(Yi("H2O")*Pt*1 + Yi("Cl2")*Pt*1)) * Approach_to_eq = k1*(((Yi("HCl")*Pt*1)^0.5)*((Yi("O2")*Pt*1)));

//Component Specific Reaction Rates: 2H2S + SO2 + N2 <==> (3/8)S8 + 2H2O + N2
  Rx("HCl") =   -2*Rate;// Consumption
  Rx("O2") =     -0.5*Rate;// Consumption
  Rx("Cl2") =    Rate;// Generation
  Rx("H2O")  =Rate;// Generation
  Rx("N2")  =0;

End
```

```
Model Adsorption



//Connection Variables with ADR2D Model
  CompAll    as StringSet(["HCl","O2","Cl2","H2O","N2"]);
  Yi(CompAll)    as molefraction;
  Pt             as pressure        (description:"Reactor Pressure[bar]");
  Tk             as temperature abs(description:"Temperature[K]") ;
  Q              as conc mole       (description:"Adsorbate Concentartion in the bed[kmol/m³]");
  Pw             as Pressure        (description:"Partial Pressure Of Water[Bar]");
  Phi as RealVariable;

// Adsorption Specific Parameters
  rho_ad     as hidden dens_mass           (description:"Catalyst Density[kg/m³]", value:1199.3, spec:
Fixed);
  Binf       as RealVariable               (description:"Maximum Adsorption Affinity [K^0.5/bar]",
value:5.3126E-5, spec:Fixed,scale:1e-8);//--> 1[Pa]=0.01[mbar]
  T0         as  hidden  temperature abs   (description:"Adsorption Reference Temperature [K]", value:
300, spec:Fixed);
  q0w        as conc_mole                  (description:"Loading Correction [kmol/m³-Adsorbent]",
value:12.886, spec:Fixed);
  gamma      as RealVariable               (description:"Adsorption Parameter [-]", value:23.235, spec
:Fixed);
  delta      as hidden  RealVariable       (description:"Adsorption Parameter [-]", value:0.68792,
spec:Fixed);
  Dp         as hidden  length             (description:"Average Particle Diameter [m]", value:3.6E-3,
 spec:Fixed);
  Rg         as hidden  RealParameter      (description:"Regnault's Constant [J/K/mol]",Value:
8.3144621);
  r_pore as realvariable(3e-8,Fixed);//cm
  Mi as realvariable(18,fixed);
  D_k as realvariable;
  D_mi as RealVariable;
  Tau as RealVariable(1,Fixed);
  D_ad as realvariable;

// Adsorption Specific Variables
  Kldf       as RealVariable   (description:"Linear Driving Force Coefficient[1/s]");// has to be
changed later on
  qsw        as conc_mole      (description:"Saturation Loading Capacity [kmol/m³-Adsorbent]");
  B          as RealVariable   (description:"Adsorption Affinity[1/bar]",scale:0.1);
  Qstar      as conc_mole      (description:"Equillibrium Loading  [kmol/m³-Adsorbent]");
  qstr       as conc_mole      (description:"Equillibrium Loading  [kmol/m³-Adsorbent]");


// Adsorption Rate Computation ===> Equivalent To Reaction Rate
  Rad(CompAll)    as  Reaction Rate(Description:"Rate [kmol/m³/s]");
  RateAd          as  Reaction Rate(Description:"Rate [kmol/m³/s]");
  A               as RealVariable;
  F_dash          as RealVariable;


//Declaring Solvent Pore Diffusion Coefficient Variables for Zeolite 3A

D_k=9700*r_pore*sqrt(Tk/Mi);
D_ad*10000=(D_k*D_mi)/(Tau*(D_k+D_mi));

// Adsorption Rate
  Pw=Yi("H2O")*Pt;
  EqAD1:  qsw=q0w*EXP(delta*(1-(Tk/T0))); //[kmol/m³-Adsorbent] -> No change
  EqAD2:  B=(Binf/SQRT(Tk))*EXP(gamma*T0/Tk);// 1/bar  -> No change
  EqAD3:  qstr=B*qsw*Pw/(1+ B*Pw);//[mol/kg-Adsorbent]  -> No change

  EQAD4:  A=B*Pw;//Unitless
  EQAD5:  F_dash =qsw*B*Tk*(8.314e-5)/((1+A)^2);
  EqAD6:  Kldf=(60/(Dp^2)) * D_ad  * (1/(1+(1/0.5)*F_dash ));//[1/s]
  //EqAD7:  $Q =(Kldf*(qstr-Q));//[kmol/(kg-adsorbent)/s]*[kg-adsorbent/m³]=[kmol/m³/s]

  IF (0.1< Phi ) then
  EqAD7:  $Q =(Kldf*(qstr-Q));//[kmol/(kg-adsorbent)/s]*[kg-adsorbent/m³]=[kmol/m³/s]
  Else
  $Q=0;
  ENDIF

  EqAD8:  RateAd=$Q;


  Rad("H2O")  =   -RateAd;// Consumption
  Rad("HCl")  =    0;// Consumption
  Rad("O2")   =   0;// Consumption
  Rad("Cl2")   =    0;// Generation
  Rad("N2")   =   0;

  Q:1e-10,Initial;



End
```

```
Model TransportProperties


//Conection Parameters With The Model ADR2D
u          as velocity(description:"True Velocity");
D_mix as RealVariable;
Rho_g      as dens_mass_vap;
Visc Bar   as Viscosity BarS;
Visc Pas   as Realvariable(description:"Gas Phase Viscosity[kg/m/s]");
Cp Vap Mass        as RealVariable(description:"Mass Heat Capacity of Gas Phase[kJ/kg/K]");
Eps    as Voidage_Fraction;
Phi as Voidage_Fraction;
Cond_Cat   as RealVariable(description:"Catalyst Conductivity[kW/m/K]",0.008385,Fixed);//0.00296
Cond_Ads   as RealVariable(description:"Adsorbent Conductivity[kW/m/K]",0.00012,Fixed);
Cond_s     as RealVariable;
Cond_g     as RealVariable(description:"Vapor Phase Conductivity[kW/m/K]");
//Declaring Reactor Scale Parameters

Dp         as length       (description:"Average Particle Diameter [m]", value:3.6E-3, spec:Fixed,
scale:1e-3);
D_Reactor as RealVariable (description:"Reactor Diameter [m]", value:0.06, spec:Fixed,scale:1e-2);



N          as RealVariable;
K          as realVariable;
M          as RealVariable(Description:"Der Parameter");
a1         as Realvariable;
a2         as Realvariable;
B          as RealVariable;

Pe_mz      as RealVariable(Description:"Axial Mass Peclet Number");
Pe_hz      as RealVariable(Description:"Axial Heat Peclet Number");
Re_p       as RealVariable(Description:"Particle Reynolds Number");
Sc         as RealVariable(Description:"Schmidt Number");
Pr         as RealVariable(Description:"Prandtl Number");

D ax       as Dispersion;
L ax       as Dispersion;
L_ax1      as RealVariable;
L_ax2      as RealVariable;
D_ar       as Realvariable;
L_ar       as Realvariable;
Kwall      as Realvariable;
h_fs       as Realvariable;
Nu         as Realvariable;
Visc_Pas=Visc_Bar*1e5;

//Calculating Solid Phase Conductivity - Rayleigh's Law
Cond_s/Cond_Cat=1+(3*Phi)/(((Cond_Ads+2*Cond_Cat)/(Cond_Ads-Cond_Cat))-(Phi)+1.569*((Cond_Ads-
Cond_Cat)/(3*Cond_Ads-4*Cond_Cat))*(Phi^(10/3)));

//Calculating DimensionLess Numbers
Pr      = Visc_Pas*Cp_Vap_Mass/(Cond_g);
Re_p    = rho_g*Dp*u/Visc_Pas;
Pe_hz   = Re_p*Pr;
Nu      = (2+1.1*(Pr^(1/3))*(Re_p^(0.6)));
// Axial Mass Dispersion Coefficient - D ax
D ax    = 20*D mix + 0.5*u*Dp;

// Axial Heat Dispersion Coefficient - L_ax
L_ax *1000    = ((0.73/(Re_p*Pr))+(0.5/(1+(9.7/(Re_p*Pr)))))*(u*Rho_g*Cp_Vap_Mass*1000*Dp);// because
this has to be kept in [Kw/m/K]

// Fluid To Solid Heat Transfer Coefficient - H_fs
h_fs=(2+1.1*(Pr^(1/3))*(Re_p^(0.6)))*Cond_g/Dp;

End
```

APPENDIX B: MULTISCALE FUNCTIONALITY DISTRIBUTION

Some studies were conducted using a heterogeneous dispersed model, where the axial length of the reactor and radial axis of the solid pellets were discretised. An example of the ACM code for these simulations is attached here:

```
Model Reactor
//-------------------------------------------------------------------------------------------------------------
//==================================BEGIN SIMULATION=======================================================//
//-------------------------------------------------------------------------------------------------------------

//-----------------------------------Components-------------------------------------------------------------
/

 CompAll                    as StringSet(["H2S","O2S","S8","H2O","N2"]);
  CompR                     as StringSet(["H2S","O2S","S8","H2O"]);
  H2S                       as StringSet(["H2S"]);//Reacting Species List
  //CompRnA                 as StringSet(["H2S","O2S","S8"]);//Reacting Not Adsorbing Species List
  //CompA                   as StringSet(["H2O"]);// Adsorbing Species List
  SumStoiC                  as RealParameter(Description:"Sum of Stoichiometric Coefficients",Value:-5/8);

//---------------------------------- Parameters ----------------------------------------------------//

  NoL                       as IntegerParameter (Description:"Number Of Axial Nodes", 50);
  NoR                       as IntegerParameter (Description:"Number Of Radial Nodes", 11);
  NumSec                    as IntegerParameter (Description:"Number Of Axial Domain Sections", 1);
  Length                    as length           (description:"Reactor Length - Axial Co-ordinate [m]", value:1, spec:Fixed);
  Diameter                  as length           (description:"Reactor Diameter - Radial Co-ordinate [m]", value:0.06, spec:Fixed);
  pi                        as RealParameter    (value:355/113);
  //Acs                     as area             (description:"Cross Sectional Area of the reactor-Normal to the flow [m²]");
  Dp                        as length           (description:"Average Particle Diameter [m]", value:3.57E-3/*4.44E-3*/, spec:Fixed,scale:1e-3);
  Rp_cat                    as RealParameter    (value:1.785E-3/*2.22E-3*/, Description:"Radius of Catalyst Pellet");
  Rc                        as RealParameter    (Value:0.5E-6, Description:"Mean Zeolite Crystal Radius");
  SwitchParameter           as Realvariable     (Fixed, Value:1);

  NumAxialStateVar          as IntegerParameter (27);
  NumStateVar               as IntegerParameter ((NoL+1)*NumAxialStateVar);
  NumCycles                 as IntegerParameter (400);

//---------------------------------- Domains ----------------------------------------------------//

  Axial                     as LengthDomain(Length: Length, DiscretizationMethod:"BFD1", HighestOrderDerivative:2,SpacingPreference:(Length/NoL), NumSections:NumSec, Description:"Axis Of The Reactor");
  Pellet                    as LengthDomain(Length: Rp cat, DiscretizationMethod:"BFD1", HighestOrderDerivative:2,SpacingPreference:(Rp_cat/NoR), Description:"Radial Pellet Coordinate");
  //Crystal                 as   LengthDomain(Length: Rc, DiscretizationMethod:"BFD1", HighestOrderDerivative:2,SpacingPreference:(Rc/NoR), Description:"Radial Crystal Coordinate");


//---------------------------------- Variables ----------------------------------------------------//

  eps bed                                   as Voidage Fraction (description:"Reactor Porosity-Catalyst Specific[-]", value:0.36, spec:Fixed);
```

```
46      Viscosity([0:Axial.EndNode])                as visc_vap (description:"Gas Phase Viscosity [cP]");
47      Visc_Bar([0:Axial.EndNode])                 as Viscosity_BarS (description:"Gas Phase Viscosity
    [Bar-s]");
48      Visc Pas([0:Axial.EndNode])                 as RealVariable (description:"Gas Phase Viscosity
    [Bar-s]");
49      //Rho_s                                       as RealVariable(description:"solid phase
    density[kg/m³]", value:750, spec:fixed);
50      Rho_g([0:Axial.EndNode])                    as dens_mass_vap(description:"Mass Density Of the gas
    phase [kg/m³]");
51      Cond_g_W([0:Axial.EndNode])                 as cond_vap (description:"Vapor Phase
    Conductivity[W/m/K]");
52      Cond_g([0:Axial.EndNode])                   as RealVariable (description:"Vapor Phase
    Conductivity[kW/m/K]");
53      //Had                                         as RealVariable (description:"Heat of
    Adsorption[kJ/kmol]", value:-30500*1, spec:Fixed);//
54      //Hrx                                         as RealVariable (description:"Heat of
    Reaction[kJ/kmol]", value:-108600*1, spec:Fixed);//
55      Mw([0:Axial.EndNode])                       as molweight (description:"Molecular Weight Of the gas
    Phase[kg/kmol]");
56      Cp_Vap_Molar([0:Axial.EndNode])             as cp_mol_vap(description:"Vapor Heat
```

```
Capacity[kJ/kmol/K]");// has to be divided by Mw to get Cp_Vap_Mass
  Cp_Vap_Mass([0:Axial.EndNode])              as RealVariable(description:"Vapor Heat
Capacity[kJ/kg/K]");
  //Cp_S                                        as RealVariable (description:"Catalyst Mass Heat
Capacity [kJ/kg/K]", value:1.050, spec:Fixed);

  u                                           as Distribution1D (XDomain is Axial,Description:"Gas
Phase Velocity [m/s]") of RealVariable;
  Pt                                          as Distribution1D (XDomain is Axial,Description:"Gas
Phase Pressure [Bar]") of Pressure;
  Yi(CompAll)([0:Axial.EndNode])              as molefraction (description:"Component Mole Fractions
in the gaseous bulk phase");
  Ct([0:Axial.EndNode])                       as conc_mole;
  Tc([0:Axial.EndNode])                       as Temperature(Free);
  Cinlet(CompAll)                             as conc_mole; // Inlet concentrations of the
components for Danckwerts boundary conditions
  Tk_inlet                                    as Temperature; // Inlet temperature of the gas for
Dankcwerts boundary conditions

  Ci(CompAll)                                 as Distribution1D (XDomain is Axial,Description:"Gas
Phase Conc [kmol/m³]",Integrals:"idx") of conc_mole;
  Tk                                          as Distribution1D (XDomain is Axial,Description:"Gas
Phase Temperature") of Temperature_abs;
  //Ts                                        as Distribution1D (XDomain is Axial,Description:"Solid
Phase Temperature") of Temperature_abs;
  //T_PCM_avg                                 as Temperature_abs;
  Tk_pellet_average                           as Temperature_abs;
  //TK_pellet_average                         as Temperature_abs;

  D_mi(CompAll)([0:Axial.EndNode])            as RealVariable;
  Dmi(CompAll)([0:Axial.EndNode])             as Diffusivity;
 // D_mix([0:Axial.EndNode])                   as RealVariable;
 // D_mol(CompAll,CompAll)([0:Axial.EndNode]) as hidden Diffus_vap;

  //DaI_cat([0:Axial.EndNode])                as RealVariable;
  //DaI_ads([0:Axial.EndNode])                as RealVariable;
  //DaII_cat([0:Axial.EndNode])               as RealVariable;
  //DaII ads([0:Axial.EndNode])               as RealVariable;

  Total_Loading_Reactor                       as RealVariable;
  Average_Loading_Reactor                     as RealVariable;
  Total_Loading_Pellet                        as Distribution1D (XDomain is Axial,Description:
"multipellet Pellet Loading [kmol/m³ multipellet]",Integrals:"idx") of conc_mole;
  Amount_adsorbed                             as Distribution1D (XDomain is Axial,Description:
"Amount of water adsorbed [kmol]",Integrals:"idx") of RealVariable;


//=============================================================================================
==========================================================================//

alpha([0:Axial.EndNode])                      as Voidage_Fraction (spec: Free);
/*beta([0:Axial.EndNode])                       as Voidage_Fraction (spec: Free);
gamma([0:Axial.EndNode])                      as Voidage_Fraction (spec: Free);
delta([0:Axial.EndNode])                      as Voidage_Fraction (spec: Free);
epsilon([0:Axial.EndNode])                    as Voidage_Fraction (spec: Free);
zeta([0:Axial.EndNode])                       as Voidage_Fraction (spec: Free);
*/

//Initial Values for better changing these values in Variable List
alpha_ini                      as Voidage_Fraction (spec: Fixed, value: 0);
beta_ini                       as Voidage_Fraction (spec: Fixed, value: 1);
gamma_ini                      as Voidage_Fraction (spec: Fixed, value: 1);
delta_ini                      as Voidage_Fraction (spec: Fixed, value: 1);
epsilon_ini                    as Voidage_Fraction (spec: Fixed, value: 1);
zeta_ini                       as Voidage_Fraction (spec: Fixed, value: 1);

alpha_sec1                      as Voidage_Fraction (spec: Fixed, value: 0);
alpha_sec2                      as Voidage_Fraction (spec: Fixed, value: 0);
alpha_sec3                      as Voidage_Fraction (spec: Fixed, value: 0);
alpha_sec4                      as Voidage_Fraction (spec: Fixed, value: 0);
alpha_sec5                      as Voidage_Fraction (spec: Fixed, value: 0);
alpha_sec6                      as Voidage_Fraction (spec: Fixed, value: 0);
alpha_sec7                      as Voidage_Fraction (spec: Fixed, value: 0);
alpha_sec8                      as Voidage_Fraction (spec: Fixed, value: 0);
alpha_sec9                      as Voidage_Fraction (spec: Fixed, value: 0);
alpha_sec10                     as Voidage_Fraction (spec: Fixed, value: 0);

/*
beta_sec1                       as Voidage_Fraction (spec: Fixed, value: 1);
gamma_sec1                      as Voidage_Fraction (spec: Fixed, value: 0.5);
delta_sec1                      as Voidage_Fraction (spec: Fixed, value: 1);
epsilon_sec1                    as Voidage_Fraction (spec: Fixed, value: 1);
zeta_sec1                       as Voidage_Fraction (spec: Fixed, value: 1);


```

```
128
129  alpha_sec2                      as Voidage_Fraction (spec: Fixed, value: 0);
130  beta_sec2                       as Voidage_Fraction (spec: Fixed, value: 1);
131  gamma_sec2                      as Voidage_Fraction (spec: Fixed, value: 1);
132  delta_sec2                      as Voidage_Fraction (spec: Fixed, value: 1);
133  epsilon_sec2                    as Voidage_Fraction (spec: Fixed, value: 0.8);
134  zeta_sec2                       as Voidage_Fraction (spec: Fixed, value: 1);
135
136  alpha sec3                      as Voidage Fraction (spec: Fixed, value: 0);
137  beta sec3                       as Voidage Fraction (spec: Fixed, value: 1);
138  gamma_sec3                      as Voidage_Fraction (spec: Fixed, value: 1);
139  delta_sec3                      as Voidage_Fraction (spec: Fixed, value: 1);
140  epsilon_sec3                    as Voidage_Fraction (spec: Fixed, value: 1);
141  zeta_sec3                       as Voidage_Fraction (spec: Fixed, value: 1);
142
143  ReactorFrac1                      as Voidage_Fraction (spec: fixed; value: 0.1);
144  ReactorFrac2                      as Voidage_Fraction (spec: fixed; value: 0.9);
145
146  Sec12                      as integerparameter (value: round(50*ReactorFrac1));
147  Sec23                      as integerparameter (value: round(50*ReactorFrac2));
148
149  Rfdummy1                     as realvariable (spec: free);
150  Rfdummy2                      as realvariable (spec: free);
151
152  Rfdummy1=ReactorFrac1+1;
153  Rfdummy2=ReactorFrac2+1;
154
155
156
157
158  alpha_sec4                      as Voidage_Fraction (spec: Fixed, value: 0.2);
159  beta_sec4                       as Voidage_Fraction (spec: Fixed, value: 1);
160  gamma_sec4                      as Voidage_Fraction (spec: Fixed, value: 1);
161  delta_sec4                      as Voidage_Fraction (spec: Fixed, value: 1);
162  epsilon_sec4                    as Voidage_Fraction (spec: Fixed, value: 1);
163  zeta_sec4                       as Voidage_Fraction (spec: Fixed, value: 1);
164
165  alpha sec5                      as Voidage Fraction (spec: Fixed, value: 0);
166  beta sec5                       as Voidage Fraction (spec: Fixed, value: 0);
167  gamma_sec5                      as Voidage_Fraction (spec: Fixed, value: 1);
168  delta_sec5                      as Voidage_Fraction (spec: Fixed, value: 1);
169  epsilon_sec5                    as Voidage_Fraction (spec: Fixed, value: 1);
170  zeta_sec5                       as Voidage_Fraction (spec: Fixed, value: 0.4);
171  */
172   For X in [0:4] do
173        alpha(x)=alpha_sec1;
174   EndFor
175    For X in [5:41] do
176        alpha(x)=alpha sec2;
177   EndFor
178    For X in [42:Axial.Endnode] do
179        alpha(x)=alpha_sec3;
180   EndFor
181
182
183  /*
184     For X in [31:40] do
185        alpha(x)=alpha sec4;
186        beta(x)=beta sec4;
187        gamma(x)=gamma_sec4;
188        delta(x)=delta_sec4;
189        epsilon(x)=epsilon_sec4;
190        zeta(x)=zeta_sec4;
191    EndFor
192
193     For X in [41:50] do
194        alpha(x)=alpha_sec5;
195        beta(x)=beta sec5;
196        gamma(x)=gamma sec5;
197        delta(x)=delta_sec5;
198        epsilon(x)=epsilon_sec5;
199        zeta(x)=zeta_sec5;
200    EndFor
201  */
202
203
204
205  //=============================================================================================================
     ==========================================================================//
206
207    Total_Amount_Adsorbed                        as RealVariable;
208
209    V_reactor                                    as RealVariable;
210
211    State_Vector_Initial([1:NumStateVar])        as RealVariable(Fixed);
```

```
212     State_Vector_Final([1:NumStateVar])         as RealVariable(Fixed);
213
214     Error([1:NumStateVar])                      as RealVariable(Free);
215
216     Overall_Error                               as RealVariable(Free);
217     Overall_Errorr([1:NumCycles])               as RealVariable(Fixed,0);
218
219     Dummy([1:NumStateVar])                      as RealVariable(Free);
220
221     //ErrorYi_H2O                               as RealVariable(Free);
222     ErrorCi_H2O                                 as RealVariable(Free);
223     ErrorTk                                     as RealVariable(Free);
224     ErrorYi_pellet                              as RealVariable(Free);
225     //ErrorYi_cat                               as RealVariable(Free);
226     ErrorTK_pellet                              as RealVariable(Free);
227     //ErrorTk_cat                               as RealVariable(Free);
228     //ErrorPCM                                  as RealVariable(Free);
229     ErrorPt                                     as RealVariable(Free);
230
231     //Error_Yi_H2O([1:NumCycles])               as RealVariable(Fixed,0);
232     Error_Ci_H2O([1:NumCycles])                 as RealVariable(Fixed,0);
233     Error_Tk([1:NumCycles])                     as RealVariable(Fixed,0);
234     Error_Yi_pellet([1:NumCycles])              as RealVariable(Fixed,0);
235     //Error_Yi_cat([1:NumCycles])               as RealVariable(Fixed,0);
236     Error_TK_pellet([1:NumCycles])              as RealVariable(Fixed,0);
237     //Error_Tk_cat([1:NumCycles])               as RealVariable(Fixed,0);
238     //Error_PCM([1:NumCycles])                  as RealVariable(Fixed,0);
239     Error_Pt([1:NumCycles])                     as RealVariable(Fixed,0);
240     CycleTimee([1:NumCycles])                   as RealVariable(Fixed,0);
241     CycleTime([1:NumCycles])                    as RealVariable;
242
243     ReactionTimee([1:NumCycles])                as RealVariable(Fixed,0);
244     ReactionTime([1:NumCycles])                 as RealVariable;
245
246     DesorptionTimee([1:NumCycles])              as RealVariable(Fixed,0);
247     DesorptionTime([1:NumCycles])               as RealVariable;
248
249     CoolingTimee([1:NumCycles])                 as RealVariable(Fixed,0);
250     CoolingTime([1:NumCycles])                  as RealVariable;
251   /*
252     MBError(CompAll)                            as RealVariable;
253     HtError                                     as RealVariable;
254     Qdotads([0:Axial.EndNode])                  as RealVariable;
255     Qdotcat([0:Axial.EndNode])                  as RealVariable;
256     QdotPCM([0:Axial.EndNode])                  as RealVariable;
257     Qdotgas([0:Axial.EndNode])                  as RealVariable;
258     Qdottotal([0:Axial.Endnode])                as RealVariable;
259   */
260
261   //=================================== OPTIMIERUNG
      ====================================================//
262
263     Tmax                                        as Temperature_abs;
264     Tmin                                        as Temperature_abs;
265     Tdummy                                      as Temperature_abs (value: 523.15, spec:fixed);
266     OPT                                         as Realvariable(Initial,0);
267
268     Tmax = max(Tk(Axial.Interior));
269     Tmin = min(Tk(Axial.Interior));
270     $OPT = Tmax;
271
272     XX                                          as realvariable;
273     XX_obj                                      as RealVariable(spec:Free, value:0);
274     XX_obj=100*XX;
275     //$XX_obj=abs(XX-0.995);
276
277   //=================================== EQUATIONS
      ====================================================//
278   /*
279     For i in [1:NumSec] do
280       Phi_ads(i)+Phi_cat(i)+Phi_PCM(i)=1;
281     EndFor
282   */
283
284     For X in [0:Axial.EndNode] do
285       Ct(X)=SIGMA(Ci(CompAll)(X));
286       //CFlux(CompAll)(X)= u(X)*Ci(CompAll)(X);
287       Yi(CompAll)(X)=Ci(CompAll)(X)/(Ct(X)+1e-10);
288       Tc(X)=Tk(X)-273.15;
289     EndFor
290
291   Cycletime(1)=Cycletimee(1);
292   Reactiontime(1)=Reactiontimee(1);
293   Desorptiontime(1)=Desorptiontimee(1)-Reactiontimee(1);
294   Coolingtime(1)=Coolingtimee(1)-Desorptiontimee(1);
```

```
295
296   For i in [2:NumCycles] do
297
298   Cycletime(i)=Cycletimee(i)-Cycletimee(i-1);
299   Reactiontime(i)=Reactiontimee(i)-Cycletimee(i-1);
300   Desorptiontime(i)=Desorptiontimee(i)-Reactiontimee(i);
301   Coolingtime(i)=Coolingtimee(i)-Desorptiontimee(i);
302
303   EndFor
304
305   //--------------------------------- Sub-Model Declaration ---------------------------------------/
306
307     //Reac([0:Axial.EndNode])                    as Reaction;
308     //Ads([0:Axial.EndNode])                     as Adsorption;
309     TP([0:Axial.EndNode])                        as TransPortProperties;
310     //Catalyst([0:Axial.Endnode])                as Catalyst;
311     //PCM([0:Axial.Endnode])                     as PCM;
312     multipellet([0:Axial.Endnode])               as multipellet;
313
314   //--------------------------------- Sub-Model Connections ---------------------------------------/
315
316
317     For X in [0:Axial.Endnode] do
318
319       TP(X).u=abs(eps_bed*u(X));
320       TP(X).D_mi(CompAll)=abs(D_mi(CompAll)(X));
321       TP(X).Visc_Bar=abs(Visc_Bar(X));
322       TP(X).rho_g=abs(rho_g(X));
323       TP(X).Cp_Vap_Mass=abs(Cp_Vap_Mass(X));
324       TP(X).Cond_g=abs(Cond_g(X));
325       TP(X).Eps=Eps_bed;
326      // TP(X).D_mix=D_mix(X);
327   /*
328       For r in [0:Pellet.Endnode] do
329         Catalyst(X).D_mol(CompAll)(CompAll)(r)=D_mol(CompAll)(CompAll)(X);
330       EndFor
331   */
332       //Catalyst(X).D_mi(CompAll)=abs(D_mi(CompAll)(X));
333       //Catalyst(X).Cond_g_W=Cond_g_W(X);
334       multipellet(X).D_mi(CompAll)=abs(D_mi(CompAll)(X));
335       multipellet(X).Cond_g_W=Cond_g_W(X);
336       //multipellet(X).Zeolite([0:multipellet(X).Pellet.Endnode]).SwitchParameter=SwitchParameter;
337
338       /*PCM(X).h_fs=TP(X).h_fs;
339       //PCM(X).Tk=Tk(X);
340
341       IF (TP(X).Phi_PCM<1E-4) THEN
342         PCM(X).Tk=523.15;
343       ELSE
344         PCM(X).Tk=Tk(X);
345       ENDIF
346         */
347     EndFor
348   /*
349   For i in [1:NumSec] do
350     For X in [Axial.Section(i).Interior] do
351       TP(X).Phi_cat=Phi_cat(i);
352       TP(X).Phi_ads=Phi_ads(i);
353       TP(X).Phi_PCM=Phi_PCM(i);
354     EndFor
355   EndFor
356
357   For i in [1:NumSec] do
358     TP(Axial.Section(i).Base).Phi_cat=Phi_cat(i);
359     TP(Axial.Section(i).Base).Phi_ads=Phi_ads(i);
360     TP(Axial.Section(i).Base).Phi_PCM=Phi_PCM(i);
361   EndFor
362
363   TP(Axial.Endnode).Phi_cat=Phi_cat(NumSec);
364   TP(Axial.Endnode).Phi_ads=Phi_ads(NumSec);
365   TP(Axial.Endnode).Phi_PCM=Phi_PCM(NumSec);
366   */
367   For i in [1:NumSec] do
368     For X in [Axial.Section(i).Interior] do
369       Total_Loading_Pellet(X)=multipellet(X).Total_Loading_Pellet;
370     EndFor
371   EndFor
372
373   Total_Loading_Pellet(0)=multipellet(0).Total_Loading_Pellet;
374   Total_Loading_Pellet(Axial.Endnode)=multipellet(Axial.Endnode).Total_Loading_Pellet;
375
376     For X in [0:Axial.Endnode] do
377         alpha(X)=multipellet(X).alpha;
378         /*beta(X)=multipellet(X).beta;
379         gamma(X)=multipellet(X).gamma;
```

```
380         delta(X)=multipellet(X).delta;
381         epsilon(X)=multipellet(X).epsilon;
382         zeta(X)=multipellet(X).zeta;*/
383       EndFor
384
385     //---------------------------------- Property Calculations
        ----------------------------------------//
386
387     For X in [0:Axial.Endnode] do
388       Call (Viscosity(X)) = pVisc Vap(Tc(X), Pt(X), Yi(CompAll,X)) ;//[cP]
389           Visc_Bar(X)   = Viscosity(X)*(1e-8);//[Bar.s]
390           Visc_Pas(X)   = Visc_Bar(X)*1e5;// [Pa.s]
391
392       Call (Rho_g(X)) = pDens_Mass_Vap(Tc(X), Pt(X), Yi(CompAll,X));//[kg/m³]
393
394       Call (Cond_g_W(X)) = pCond_Vap(Tc(X),Pt(X),Yi(CompAll)(X));//[W/m/K]
395           Cond_g(X)=Cond_g_W(X)/1000;//[kW/m/K]
396
397       Call (Mw(X)) = pMolweight(Yi(CompAll)(X));//[kg/kmol]
398
399       Call (Cp_Vap_Molar(X)) = pCp_Mol_Vap(Tc(X),Pt(X),Yi(CompAll)(X));//[kJ/kmol/K]
400           Cp_Vap_Mass(X)=abs(Cp_Vap_Molar(X)/Mw(X));//[kJ/kg/K]
401
402       Call (Dmi(CompAll)(X)) = pDiffus_Vap(Tc(X), Pt(X), Yi(CompAll)(X));//[cm²/s]
403           D_mi(CompAll)(X)=abs(Dmi(CompAll)(X))/10000;// [m²/s]. D_mol is in [cm²/s]--->
    1[m²/s]=10000[cm²/s]
404
405     //  Call (D mol(CompAll)(CompAll)(X)) = pDiffus bin Vap(Tc(X), Pt(X), Yi(CompAll,X));//[cm²/s]
406     //      D mix(X)=SIGMA(Yi(CompAll,X)*(D mol(CompAll)(CompAll)(X)))/10000;// [m²/s]. D mol is in
        [cm²/s]---> 1[m²/s]=10000[cm²/s]
407     EndFor
408
409     //--------------------------------- Dimensionless Numbers ----------------------------------------/
410     /*
411     For X in [0:Axial.Endnode] do
412
        DaI_cat(X)=abs(Catalyst(X).Reaction(Pellet.Endnode).Rx("H2S"))/(pi*0.03^2*u(X)*(Ci("H2S")(X)+1E-10));
413
        DaI ads(X)=multipellet(X).Zeolite(Pellet.Endnode).Rate ads("H2O")/(pi*0.03^2*u(X)*(Ci("H2O")(X)+1E-10
        );
414
        DaII_cat(X)=abs(Catalyst(X).Reaction(Pellet.Endnode).Rx("H2S"))/abs(TP(X).k_fs("H2S")*6/Dp*(1-eps_bed
        /eps_bed*TP(X).Phi_cat*(Ci("H2S")(X)-Catalyst(X).C_catpellet("H2S")(Pellet.Endnode)));
415
        //DaII_ads(X)=multipellet(X).Zeolite(Pellet.Endnode).Rate_ads("H2O")/abs(TP(X).k_fs("H2O")*6/Dp*(1-ep
        _bed)/eps_bed*TP(X).Phi_ads*(Ci("H2O")(X)-multipellet(X).C_adspellet("H2O")(Pellet.Endnode)));
416     EndFor
417     */
418     //---------------------------------------- PDES
        ----------------------------------------------------//
419
420     //---------------------------------- Component Balances
        ----------------------------------------//
421
422     //For i in [1:NumSec] do
423       For X in [Axial/*.Section(i)*/.Interior] do
424       //  D_axdCidx(CompAll)(X)=TP(X).D_ax(CompAll)*Ci(CompAll)(X).ddX;
425         $Ci(CompAll)(X)=  - u(X)*Ci(CompAll)(X).ddX - Ci(CompAll)(X)*u(X).ddX  +   TP(X).D ax(CompAll)*Ci
        (CompAll)(X).d2dX2
426                           - TP(X).k_fs(CompAll)*
        /*(4*355/113*(Dp/2)^2-multipellet(X).n_contacts*355/113*(1/(2/(Dp/2)))^2)/(4/3*355/133*(Dp/2)^3)*/6/
        Dp*(1-eps_bed)/eps_bed*/*Phi_cat(i)*/(Ci(CompAll)(X)-multipellet(X).C_pellet(CompAll)(Pellet.Endnode
        ));
427                           //-
        TP(X).k_fs(CompAll)*/*(4*355/113*(Dp/2)^2-multipellet(X).n_contacts*355/113*(1/(2/(Dp/2)))^2)/(4/3*35
        /133*(Dp/2)^3)*/6/Dp*(1-eps_bed)/eps_bed*Phi_ads(i)*(Ci(CompAll)(X)-multipellet(X).C_adspellet(CompAl
        )(Pellet.Endnode));
428       //EndFor
429     EndFor
430
431     //---------------------------------- Heat Balance - Solid Phase ----------------------------------/
432
433     /* Probably don't need this, since you're doing heat balances on each individual solid phase
434
435     For i in [1:NumSec] do
436       For X in [Axial.Section(i).Interior] do
437          ((1-eps_bed)*rho_s*cp_s)*$Ts(X)      = + TP(X).Cond_s*Ts(X).d2dX2
438                                                 - (1-eps bed)*TP(X).h fs*(6/Dp)*(Ts(X)-Tk(X))
439                                                 + (-Hrx*(1-eps bed)*phi cat(i)*Reac(X).Rate
440                                                 + Had*(1-eps_bed)*Phi_ads(i)*Ads(X).Rad("H2O"));
441       EndFor
442     EndFor
443     */
444
445     //---------------------------------- Heat Balance - Gas Phase ------------------------------------//
```

```
446   /* Remove this when you finish your multipellet and PCM balances!*/
447   //For i in [1:NumSec] do
448     For X in [Axial/*.Section(i)*/.Interior] do
449     (Rho_g(X)*Cp_Vap_Mass(X))*$Tk(X) = - Rho_g(X)*Cp_Vap_Mass(X)*u(X)*Tk(X).ddX
450                                        + TP(X).L_ax*Tk(X).d2dX2
451                                        - (1-eps_bed)/eps_bed*/*Phi_cat(i)*/TP(X).h_fs*
      /*(4*355/113*(Dp/2)^2-multipellet(X).n_contacts*355/113*(1/(2/(Dp/2)))^2)/(4/3*355/133*(Dp/2)^3)*/6/
      Dp*(Tk(X)-multipellet(X).Tk pellet(Pellet.Endnode));
452                                        //-
      (1-eps bed)/eps bed*Phi ads(i)*TP(X).h fs*/*(4*355/113*(Dp/2)^2-multipellet(X).n contacts*355/113*(1/
      2/(Dp/2)))^2)/(4/3*355/133*(Dp/2)^3)*/6/Dp*(Tk(X)-multipellet(X).TK_pellet(Pellet.Endnode))
453                                        //-
      (1-eps_bed)/eps_bed*Phi_PCM(i)*TP(X).h_fs*(6/Dp)*(Tk(X)-PCM(X).T_PCM);
454     //EndFor
455   EndFor
456   /*
457   Qdotcat([0:Axial.EndNode]) = -
      (1-eps bed)/eps bed*TP([0:Axial.EndNode]).Phi cat*TP([0:Axial.EndNode]).h fs*6/Dp*(Tk([0:Axial.EndNod
      ])-Catalyst([0:Axial.EndNode]).Tk cat(Pellet.Endnode));
458
459   Qdotads([0:Axial.EndNode]) = -
      (1-eps_bed)/eps_bed*TP([0:Axial.EndNode]).Phi_ads*TP([0:Axial.EndNode]).h_fs*6/Dp*(Tk([0:Axial.EndNod
      ])-multipellet([0:Axial.EndNode]).TK_pellet(Pellet.Endnode));
460
461   For i in [1:NumSec] do
462     For X in [Axial.Section(i).Interior] do
463
464     Qdotgas(X) =  rho g(X)*Cp Vap Mass(X)*$Tk(X)-rho g(X)*Cp Vap Mass(X)*u(X)*Tk(X).ddX-
      Rho g(X)*Cp Vap Mass(X)*u(X).ddX*Tk(X)- TP(X).L ax*Tk(X).d2dX2;
465     QdotPCM(X) = - (1-eps_bed)/eps_bed*TP(X).Phi_PCM*TP(X).h_fs*6/Dp*(Tk(X)-PCM(X).T_PCM);
466
467     Endfor
468   EndFor
469
470   For i in [2:NumSec] do
471
472
      Qdotgas(Axial.Section(i).Base)=Qdotgas(Axial.Section(i).Base-1);//=(rho g(Axial.Section(i).Base)*Cp V
      p Mass(Axial.Section(i).Base)*$Tk(Axial.Section(i).Base)-rho g(Axial.Section(i).Base)*Cp Vap Mass(Axi
      l.Section(i).Base)*u(Axial.Section(i).Base)*Tk(Axial.Section(i).Base).ddX -
      Rho_g(Axial.Section(i).Base)*Cp_Vap_Mass(Axial.Section(i).Base)*u(Axial.Section(i).Base).ddX*Tk(Axial
      Section(i).Base))/*-TP(Axial.Section(i).Base).L_ax*Tk(Axial.Section(i).Base).d2dX2)*/;
473
      //Qdotgas(Axial.Section(i).Base):=Qdotgas(Axial.Section(i).Base-1);//(rho_g(Axial.Section(i).Base)*Cp_
      Vap_Mass(Axial.Section(i).Base)*$Tk(Axial.Section(i).Base)-rho_g(Axial.Section(i).Base)*Cp_Vap_Mass(A
      ial.Section(i).Base)*u(Axial.Section(i).Base)*Tk(Axial.Section(i).Base).ddX -
      Rho_g(Axial.Section(i).Base)*Cp_Vap_Mass(Axial.Section(i).Base)*u(Axial.Section(i).Base).ddX*Tk(Axial
      Section(i).Base))/*-TP(Axial.Section(i).Base).L ax*Tk(Axial.Section(i).Base).d2dX2)*/,Initial;
474
475       QdotPCM(Axial.Section(i).Base)=QdotPCM(Axial.Section(i).Base-1);
476
477   EndFor
478
479   Qdotgas(0) = (rho_g(0)*Cp_Vap_Mass(0)*$Tk(0)+rho_g(0)*Cp_Vap_Mass(0)*u(0)*Tk(0).ddX-
      Rho_g(0)*Cp_Vap_Mass(0)*u(0).ddX*Tk(0));
480   Qdotgas(0) :  (rho_g(0)*Cp_Vap_Mass(0)*$Tk(0)+rho_g(0)*Cp_Vap_Mass(0)*u(0)*Tk(0).ddX-
      Rho_g(0)*Cp_Vap_Mass(0)*u(0).ddX*Tk(0)),Initial;
481   Qdotgas(Axial.Endnode) =
      (rho g(Axial.EndNode)*Cp Vap Mass(Axial.EndNode)*$Tk(Axial.EndNode)+rho g(Axial.EndNode)*Cp Vap Mass(
      xial.EndNode)*u(Axial.EndNode)*Tk(Axial.EndNode).ddX-
      Rho_g(Axial.Endnode)*Cp_Vap_Mass(Axial.Endnode)*u(Axial.Endnode).ddX*Tk(Axial.Endnode));
482   Qdotgas(Axial.Endnode) :
      (rho_g(Axial.EndNode)*Cp_Vap_Mass(Axial.EndNode)*$Tk(Axial.EndNode)+rho_g(Axial.EndNode)*Cp_Vap_Mass(
      xial.EndNode)*u(Axial.EndNode)*Tk(Axial.EndNode).ddX-
      Rho_g(Axial.Endnode)*Cp_Vap_Mass(Axial.Endnode)*u(Axial.Endnode).ddX*Tk(Axial.Endnode)),Initial;
483
484   QdotPCM(0)= - (1-eps_bed)/eps_bed*TP(0).Phi_PCM*TP(0).h_fs*6/Dp*(Tk(0)-PCM(0).T_PCM);
485   QdotPCM(Axial.Endnode)= -
      (1-eps bed)/eps bed*TP(Axial.Endnode).Phi PCM*TP(Axial.Endnode).h fs*6/Dp*(Tk(Axial.Endnode)-PCM(Axia
      .Endnode).T_PCM);
486
487
488   For X in [0:Axial.Endnode] do
489
490     Qdottotal(X)=Qdotgas(X)+Qdotcat(X)+Qdotads(X)+QdotPCM(X);
491
492   Endfor
493
494   */
495   //-------------------------------- Overall Momentum and Mass Balances ----------------------------//
496
497   //For i in [1:NumSec] do
498     For X in [Axial/*.Section(i)*/.Interior] do
499
500       -Pt(X).ddX=(150*Visc_Pas(X)*(1-eps_bed)^2*u(X)/(Dp^2*eps_bed^3)+1.75*rho_g(X)*(1-eps_bed)*abs(u(x
```

```
))*u(x)/(Dp*eps_bed^3))/100000;

    /*u(X)/(0.08314*Tk(X))*Pt(X).ddX*/-u(X)*Pt(X)/(0.08314*Tk(X)^2)*Tk(X).ddX+Pt(X)/(0.08314*Tk(X))*u(X).ddX=/*-(1/(0.08314*Tk(X))*$Pt(X)-Pt(X)/(0.08314*Tk(x)^2)*$Tk(X))*/-(1-eps_bed)/eps_bed*6/Dp*(/*Phi_cat(i)*/(TP(X).k_fs("H2S")*(Ci("H2S")(X)-multipellet(X).C_pellet("H2S")(Pellet.Endnode))
                                                +TP(X).k_fs("O2S")*(Ci("O2S")(X)-multipellet(X).C_pellet("O2S")(Pellet.Endnode))
                                                +TP(X).k fs("H2O")*(Ci("H2O")(X)-multipellet(X).C pellet("H2O")(Pellet.Endnode))
                                                +TP(X).k fs("S8")*(Ci("S8")(X)-multipellet(X).C_pellet("S8")(Pellet.Endnode))
                                                +TP(X).k_fs("N2")*(Ci("N2")(X)-multipellet(X).C_pellet("N2")(Pellet.Endnode))));

//+Phi_ads(i)*(TP(X).k_fs("H2S")*(Ci("H2S")(X)-multipellet(X).C_adspellet("H2S")(Pellet.Endnode))
                                                //
+TP(X).k_fs("O2S")*(Ci("O2S")(X)-multipellet(X).C_adspellet("O2S")(Pellet.Endnode))
                                               //
+TP(X).k fs("H2O")*(Ci("H2O")(X)-multipellet(X).C adspellet("H2O")(Pellet.Endnode))
                                              //
+TP(X).k_fs("S8")*(Ci("S8")(X)-multipellet(X).C_adspellet("S8")(Pellet.Endnode))
                                                //
+TP(X).k_fs("N2")*(Ci("N2")(X)-multipellet(X).C_adspellet("N2")(Pellet.Endnode))));

  //EndFor
EndFor

//------------------------------- Performance Indices ----------------------------------------//
//Variables to plot
//BreakH2o                                      as RealVariable;
//Impurity                                      as realvariable;
//equllibrium([0:Axial.EndNode])                as Realvariable;
  //BreakH2o=Yi("H2O")(Axial.EndNode);

  //XD([0:Axial.EndNode])                       as Realvariable;
  Ndot(CompR)                                   as RealVariable;
  Ndottotal                                     as RealVariable;
  IntegralNdotS8                                as RealVariable;
  IntegralNdottotal                             as RealVariable;
  Purity                                        as RealVariable;

  IntegralNdotS8:0, Initial;
  IntegralNdottotal:0, Initial;


    (Ci("H2S")(0)*u(0))*XX=abs((Ci("H2S")(0)*u(0))-(Ci("H2S")(Axial.EndNode)*u(Axial.EndNode)));
    Ndot(CompR)=pi*Diameter^2/4*u(Axial.Endnode)*Ci(CompR)(Axial.Endnode);
    Ndottotal=pi*Diameter^2/4*u(Axial.Endnode)*(Ci("H2S")(Axial.Endnode)+Ci("S8")(Axial.Endnode)+Ci("O2S")(Axial.Endnode)+Ci("H2O")(Axial.Endnode));
    $IntegralNdotS8=Ndot("S8");
    $IntegralNdottotal=Ndottotal;
    Purity*IntegralNdottotal=IntegralNdotS8;


//Impurity=((Yi("H2S")(Axial.EndNode))+(Yi("O2S")(Axial.EndNode)))/((Yi("H2S")(Axial.EndNode))+(Yi("OS")(Axial.EndNode))+(Yi("S8")(Axial.EndNode)));
/*For X in [0:Axial.Endnode] do
  (Ci("H2S")(0)*u(0))*XD(x)=((Ci("H2S")(0)*u(0))-(Ci("H2S")(X)*u(X)));
  //equllibrium(X)=((Yi("H2O")(X))^2)*((Yi("S8")(X))^(3/8))/(((Yi("H2S")(X))^2)*(Yi("O2S")(X)));
EndFor*/

/*
tbreakintavg as RealVariable;
//tbreakavg as RealVariable;
//iavgconc as RealVariable;
integralavgconc as RealVariable;
//iavgconc=Sigma(Ci("H2O"))/NoL;
integralavgconc=SIGMA(Ci("H2O").idx)/Length;
$tbreakintavg=1-Ci("H2O")(Axial.Endnode)/integralavgconc;
//$tbreakavg=1-Ci("H2O")(Axial.Endnode)/iavgconc;
tbreakintavg:0,Initial;
//tbreakavg:0.00001,Initial;
sPhi_ads([1:NumSec]) as SlackVariable;
//sphi2 as SlackVariable;
stbreakintavg as SlackVariable;
/*For i in [1:NumSec] do
  Phi_ads(i)+sPhi_ads(i)=1;
EndFor*/
//phi2+sphi2=1;
sXX as Slackvariable;
XX-sXX=0.995;
tbreakintavg-stbreakintavg=600;
*/
/*
sPhi_ads([1:NumSec]) as SlackVariable;
```

```
656   ENDSWITCH
657
658   Overall_obj as RealVariable;
659   Overall_obj=XX_obj-Amt_obj-Temp_obj;
660   XX_obj:0.005, Initial;*/
661   //Amt_obj:0,Initial;
662   //Temp_obj:0,Initial;
663
664   /*
665       $XX obj=XX-0.995;
666       XX_obj:0.005, Initial;
667
668       $Amt_obj=Total_Amount_Adsorbed-1E-3;
669       Amt_obj:0,Initial;
670
671       $Temp_obj=T_PCM_avg-523.15;
672       Temp_obj:0,Initial;
673
674       Overall obj=XX obj-Amt obj-Temp obj;
675   */
676
677     Absolute_rate_reaction([0:Axial.Endnode]) as RealVariable;
678     Absolute_rate_adsorption([0:Axial.Endnode]) as RealVariable;
679
680   Total_Loading_Reactor=SIGMA(Total_Loading_Pellet/*.idx*/);
681
682   For i in [1:NumSec] do
683     For X in [Axial.Section(i).Interior] do
684      Amount adsorbed(X)=Total Loading Pellet(X)*(1-eps bed)*/*TP(X).Phi ads*/V reactor;
685      //Absolute_rate_reaction(X)=Catalyst(X).Total_reacrate_Pellet*(1-eps_bed)*TP(X).Phi_cat*V_reactor;
686
      //Absolute_rate_adsorption(X)=multipellet(X).Total_adsrate_Pellet*(1-eps_bed)*TP(X).Phi_ads*V_reactor
687     EndFor
688   EndFor
689
690   Amount_adsorbed(0)=Total_Loading_Pellet(0)*(1-eps_bed)*/*TP(0).Phi_ads*/V_reactor;
691   Amount adsorbed(Axial.Endnode)=Total Loading Pellet(Axial.Endnode)*(1-eps bed)*
      /*TP(Axial.Endnode).Phi ads*/V reactor;
692   Absolute_rate_reaction([0:Axial.Endnode])=multipellet([0:Axial.Endnode]).Total_reacrate_Pellet*(1-
      eps_bed)*/*TP([0:Axial.Endnode]).Phi_cat*/V_reactor;
693   Absolute_rate_adsorption([0:Axial.Endnode])=multipellet([0:Axial.Endnode]).Total_adsrate_Pellet*(1-
      eps_bed)*/*TP([0:Axial.Endnode]).Phi_ads*/V_reactor;
694   //Absolute_rate_reaction(0)=Catalyst(0).Total_reacrate_Pellet*(1-eps_bed)*TP(0).Phi_cat*V_reactor;
695   //Absolute_rate_adsorption(0)=multipellet(0).Total_adsrate_Pellet*(1-eps_bed)*TP(0).Phi_ads*V_reactor
696   //Absolute_rate_reaction(Axial.Endnode)=Catalyst(Axial.Endnode).Total_reacrate_Pellet*(1-eps_bed)*TP(
      xial.Endnode).Phi_cat*V_reactor;
697   //Absolute rate adsorption(Axial.Endnode)=multipellet(Axial.Endnode).Total adsrate Pellet*(1-eps bed)
      TP(Axial.Endnode).Phi ads*V reactor;
698
699   Total_Amount_Adsorbed=SIGMA(Amount_Adsorbed/*.idx*/);
700
701   Average_Loading_Reactor=SIGMA(multipellet([0:Axial.Endnode]).Average_Loading_Pellet)/NoL;
702
703   V_reactor=pi*(Diameter/2)^2*Length;
704
705   //T_PCM_avg=SIGMA(PCM([1:Axial.Endnode]).T_PCM)/NoL;
706
707   //Tk cat average=SIGMA(Catalyst([1:Axial.Endnode]).Tk cat avg)/NoL;
708
709   Tk_pellet_average=SIGMA(multipellet([1:Axial.Endnode]).Tk_pellet_avg)/NoL;
710
711
712
713
714   /*
715   MBError("H2S") =
      u(0)*pi*Diameter^2/4*Ci("H2S")(0)-u(Axial.Endnode)*pi*Diameter^2/4*Ci("H2S")(Axial.Endnode)
716
      -pi*Diameter^2/4*(1-eps_bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k_fs("H2S")*TP(Axial.Interior).Phi_
      at*(Ci("H2S")(Axial.Interior)-Catalyst(Axial.Interior).C_catpellet("H2S")(Pellet.Endnode)))
717
      -pi*Diameter^2/4*(1-eps_bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k_fs("H2S")*TP(Axial.Interior).Phi_
      ds*(Ci("H2S")(Axial.Interior)-multipellet(Axial.Interior).C_adspellet("H2S")(Pellet.Endnode)))
718                   -pi*Diameter^2/4*eps_bed*Length*SIGMA($Ci("H2S")(Axial.Interior));
719
720   MBError("H2O") =
      u(0)*pi*Diameter^2/4*Ci("H2O")(0)-u(Axial.Endnode)*pi*Diameter^2/4*Ci("H2O")(Axial.Endnode)
721
      -pi*Diameter^2/4*(1-eps_bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k_fs("H2O")*TP(Axial.Interior).Phi_
      at*(Ci("H2O")(Axial.Interior)-Catalyst(Axial.Interior).C_catpellet("H2O")(Pellet.Endnode)))
722
      -pi*Diameter^2/4*(1-eps_bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k_fs("H2O")*TP(Axial.Interior).Phi_
      ds*(Ci("H2O")(Axial.Interior)-multipellet(Axial.Interior).C_adspellet("H2O")(Pellet.Endnode)))
723                   -pi*Diameter^2/4*eps_bed*Length*SIGMA($Ci("H2O")(Axial.Interior));
```

```
724
725   MBError("S8") =
      u(0)*pi*Diameter^2/4*Ci("S8")(0)-u(Axial.Endnode)*pi*Diameter^2/4*Ci("S8")(Axial.Endnode)
726
      -pi*Diameter^2/4*(1-eps_bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k_fs("S8")*TP(Axial.Interior).Phi_c
      t*(Ci("S8")(Axial.Interior)-Catalyst(Axial.Interior).C_catpellet("S8")(Pellet.Endnode)))
727
      -pi*Diameter^2/4*(1-eps bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k fs("S8")*TP(Axial.Interior).Phi a
      s*(Ci("S8")(Axial.Interior)-multipellet(Axial.Interior).C adspellet("S8")(Pellet.Endnode)))
728                 -pi*Diameter^2/4*eps bed*Length*SIGMA($Ci("S8")(Axial.Interior));
729
730   MBError("O2S") =
      u(0)*pi*Diameter^2/4*Ci("O2S")(0)-u(Axial.Endnode)*pi*Diameter^2/4*Ci("O2S")(Axial.Endnode)
731
      -pi*Diameter^2/4*(1-eps_bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k_fs("O2S")*TP(Axial.Interior).Phi_
      at*(Ci("O2S")(Axial.Interior)-Catalyst(Axial.Interior).C_catpellet("O2S")(Pellet.Endnode)))
732
      -pi*Diameter^2/4*(1-eps bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k fs("O2S")*TP(Axial.Interior).Phi
      ds*(Ci("O2S")(Axial.Interior)-multipellet(Axial.Interior).C adspellet("O2S")(Pellet.Endnode)))
733                 -pi*Diameter^2/4*eps bed*Length*SIGMA($Ci("O2S")(Axial.Interior));
734
735   MBError("N2") =
      u(0)*pi*Diameter^2/4*Ci("N2")(0)-u(Axial.Endnode)*pi*Diameter^2/4*Ci("N2")(Axial.Endnode)
736
      -pi*Diameter^2/4*(1-eps_bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k_fs("N2")*TP(Axial.Interior).Phi_c
      t*(Ci("N2")(Axial.Interior)-Catalyst(Axial.Interior).C_catpellet("N2")(Pellet.Endnode)))
737
      -pi*Diameter^2/4*(1-eps bed)*Length*6/Dp*SIGMA(TP(Axial.Interior).k fs("N2")*TP(Axial.Interior).Phi a
      s*(Ci("N2")(Axial.Interior)-multipellet(Axial.Interior).C adspellet("N2")(Pellet.Endnode)))
738                 -pi*Diameter^2/4*eps_bed*Length*SIGMA($Ci("N2")(Axial.Interior));
739
740   HtError =
      rho_g(0)*Cp_Vap_Mass(0)*pi*Diameter^2/4*Length*u(0)*Tk(0)-rho_g(Axial.Endnode)*Cp_Vap_Mass(Axial.Endn
      de)*pi*Diameter^2/4*Length*u(Axial.Endnode)*Tk(Axial.Endnode)
741               -
      pi*Diameter^2/4*Length*(1-eps_bed)*6/Dp*SIGMA(TP(Axial.Interior).h_fs*TP(Axial.Interior).Phi_cat*(Tk(
      xial.Interior)-Catalyst(Axial.Interior).Tk_cat(Pellet.Endnode)))
742               -
      pi*Diameter^2/4*Length*(1-eps bed)*6/Dp*SIGMA(TP(Axial.Interior).h fs*TP(Axial.Interior).Phi ads*(Tk(
      xial.Interior)-multipellet(Axial.Interior).TK_pellet(Pellet.Endnode)))
743               -
      pi*Diameter^2/4*Length*(1-eps_bed)*6/Dp*SIGMA(TP(Axial.Interior).h_fs*TP(Axial.Interior).Phi_PCM*(Tk(
      xial.Interior)-PCM(Axial.Interior).T_PCM))
744               -
      pi*Diameter^2/4*Length*(eps_bed)*SIGMA(rho_g(Axial.Interior)*Cp_Vap_Mass(Axial.Interior)*$Tk(Axial.In
      erior));
745   */
746   //-------------------------------- Initial Conditions ----------------------------------------//
747
748     For X in [Axial.Interior] do
749       //Ci(CompR)(x):0,Initial;
750       Ci("H2S")(x):0, Initial;
751       Ci("H2O")(x):0, Initial;
752       Ci("O2S")(x):0, Initial;
753       Ci("S8")(x):1E-4, Initial;
754       Ci("N2")(x):0.023295947,Initial;
755       Tk(x):523.15,Initial;
756       //Ts(x):523.15,Initial;
757     EndFor
758
759
760   //-------------------------------- Non-switching BCs -----------------------------//
761   /*
762   Pt(0)=Ct(0)*(0.08314)*Tk(0);
763   -Pt(Axial.Endnode).ddX=(150*Visc_Pas(Axial.Endnode)*(1-eps_bed)^2*u(Axial.Endnode)/(Dp^2*eps_bed^3)+1
      75*rho_g(Axial.Endnode)*(1-eps_bed)*abs(u(Axial.Endnode))*u(Axial.Endnode)/(Dp*eps_bed^3))/100000;
764
765   u(Axial.Endnode).ddX=0;
766
767   rho_g(0)*Cp_Vap_Mass(0)*u(0)*Tk_inlet=rho_g(0)*Cp_Vap_Mass(0)*u(0)*Tk(0)+TP(0).L_ax*Tk(0).ddX;
768   Tk(Axial.EndNode).ddX=0;
769
770   u(0)*Ci(CompAll)(0)=u(0)*Cinlet(CompAll)+TP(0).D_ax(CompAll)*Ci(CompAll)(0).ddX;
771   Ci(CompAll)(Axial.EndNode).ddX=0;
772   */
773   //-------------------------------- Switching BCs ---------------------------------------//
774   //--------------------------------BC SET 7 Countercurrent cooling 15.03.2015 Reaction, Desorption,
      Precooling, countercurrent cooling, prereaction ----------------------------------------//
775
776   SWITCH
777
778     Initial state REACTIONN
779
780       u(0)=0.2;
781
```

```
782       /*u(Axial.Endnode)/(0.08314*Tk(Axial.Endnode))*Pt(Axial.Endnode).ddX*/-u(Axial.Endnode)*Pt(
      Axial.Endnode)/(0.08314*Tk(Axial.Endnode)^2)*Tk(Axial.Endnode).ddX+Pt(Axial.Endnode)/(0.08314*Tk(
      Axial.Endnode))*u(Axial.Endnode).ddX=
      /*-(1/(0.08314*Tk(Axial.Endnode))*$Pt(Axial.Endnode)-Pt(Axial.Endnode)/(0.08314*Tk(Axial.Endnode)^2)*
      Tk(Axial.Endnode))*/-(1-eps_bed)/eps_bed*6/Dp*(/*Phi_cat(NumSec)*/(TP(Axial.Endnode).k_fs("H2S")*(Ci(
      "H2S")(Axial.Endnode)-multipellet(Axial.Endnode).C_pellet("H2S")(Pellet.Endnode))
783                                                 +TP(Axial.Endnode).k_fs("O2S")*(Ci("O2S")(
      Axial.Endnode)-multipellet(Axial.Endnode).C pellet("O2S")(Pellet.Endnode))
784                                                 +TP(Axial.Endnode).k fs("H2O")*(Ci("H2O")(
      Axial.Endnode)-multipellet(Axial.Endnode).C pellet("H2O")(Pellet.Endnode))
785                                                 +TP(Axial.Endnode).k_fs("S8")*(Ci("S8")(
      Axial.Endnode)-multipellet(Axial.Endnode).C_pellet("S8")(Pellet.Endnode))
786                                                 +TP(Axial.Endnode).k_fs("N2")*(Ci("N2")(
      Axial.Endnode)-multipellet(Axial.Endnode).C_pellet("N2")(Pellet.Endnode))));
787
      //+Phi_ads(NumSec)*(TP(Axial.Endnode).k_fs("H2S")*(Ci("H2S")(Axial.Endnode)-multipellet(Axial.Endnode
      .C_adspellet("H2S")(Pellet.Endnode))
788                                             //
      +TP(Axial.Endnode).k fs("O2S")*(Ci("O2S")(Axial.Endnode)-multipellet(Axial.Endnode).C adspellet("O2S"
      (Pellet.Endnode))
789                                               //
      +TP(Axial.Endnode).k_fs("H2O")*(Ci("H2O")(Axial.Endnode)-multipellet(Axial.Endnode).C_adspellet("H2O"
      (Pellet.Endnode))
790                                                 //
      +TP(Axial.Endnode).k_fs("S8")*(Ci("S8")(Axial.Endnode)-multipellet(Axial.Endnode).C_adspellet("S8")(P
      llet.Endnode))
791                                                   //
      +TP(Axial.Endnode).k fs("N2")*(Ci("N2")(Axial.Endnode)-multipellet(Axial.Endnode).C adspellet("N2")(P
      llet.Endnode))));
792
793       Cinlet("H2S")=0.002329595; Cinlet("O2S")=0.001164797; Cinlet("S8")=0;  Cinlet("H2O")=0; Cinlet(
      "N2")=0.019801555;
794       u(0)*Ci(CompAll)(0)=u(0)*Cinlet(CompAll)+TP(0).D_ax(CompAll)*Ci(CompAll)(0).ddX;
795       Ci(CompAll)(Axial.EndNode).ddX=0;
796
797       Tk_inlet=Tdummy;//273.15+250;
798       rho_g(0)*Cp_Vap_Mass(0)*u(0)*Tk(0)=rho_g(0)*Cp_Vap_Mass(0)*u(0)*Tk_inlet+TP(0).L_ax*Tk(0).ddX;
799       Tk(Axial.EndNode).ddX=0;
800
801       Pt(0)=Ct(0)*(0.08314)*Tk(0);
802       -Pt(Axial.Endnode).ddX=(150*Visc_Pas(Axial.Endnode)*(1-eps_bed)^2*u(Axial.Endnode)/(Dp^2*eps_bed^
      3)+1.75*rho_g(Axial.Endnode)*(1-eps_bed)*abs(u(Axial.Endnode))*u(Axial.Endnode)/(Dp*eps_bed^3))/
      100000;
803
804       IF (SwitchParameter > 0 and SwitchParameter <= 1) STATE : REACTIONN;
805
806     ENDSTATE
807   ENDSWITCH
808
809
810     // FOR THE CATALYST
811
812   //For n in [1:NumSec] do
813     For i in [0:Axial.Endnode] do
814       //Catalyst(i).C_catpellet(CompAll)(0).ddX=0; //Symmetry
815       TP(i).k_fs(CompAll)*(Ci(CompAll)(i)-multipellet(i).C_pellet(CompAll)(Pellet.Endnode))=multipellet
      (i).D_eff_pellet(CompAll)(Pellet.Endnode)*multipellet(i).C_pellet(CompAll)(Pellet.Endnode).ddX; //
      continuity of flux
816
817
      //Catalyst(i).Cond_cat/*_eff_W*/*Catalyst(i).Tk_cat(Pellet.Endnode).ddX=TP(i).h_fs*(Tk(i)-Catalyst(i)
      Tk_cat(Pellet.Endnode));
818
819
820       //Catalyst(i).Tk_cat(0).ddX=0;
821     //EndFor
822   //EndFor
823
824     // FOR THE multipellet
825
826   //For n in [1:NumSec] do
827     //For i in [0:Axial.Endnode] do
828       multipellet(i).C_pellet(CompAll)(0).ddX=0; //Symmetry
829
      //TP(i).k_fs(CompAll)*(Ci(CompAll)(i)-multipellet(i).C_adspellet(CompAll)(Pellet.Endnode))=multipelle
      (i).D_ads(CompAll)(Pellet.Endnode)*multipellet(i).C_adspellet(CompAll)(Pellet.Endnode).ddX; //
      continuity of flux
830
831
832       multipellet(i).Cond_pellet/*_eff_W*//*(Pellet.Endnode)*/*multipellet(i).TK_pellet(Pellet.Endnode
      ).ddX=TP(i).h_fs*(Tk(i)-multipellet(i).TK_pellet(Pellet.Endnode));
833
834
835       multipellet(i).TK_pellet(0).ddX=0;
836
```

```
837     EndFor
838
839   n_S8 as realvariable(initial);
840   if XX>0.995 then
841     $n_S8=Ci("S8")(Axial.Endnode)*Diameter^2/4*pi*u(Axial.Endnode);
842   else
843     $n_S8=0;
844   endif
845
846   END
```

```
Model multipellet
//-----------------------------------Components-------------------------------------------------------
/

  CompAll   as StringSet(["H2S","O2S","S8","H2O","N2"]);


//---------------------------------- Parameters
-------------------------------------------------------//

  NoR       as IntegerParameter (Description:"Number Of Radial Nodes", 11);

  Rp_ads    as RealParameter(value:1.785E-3/*2.22E-3*/, Description:"Radius of Adsorbent Pellet");
  Rc        as RealParameter(Value:0.5E-6, Description:"Mean Zeolite Crystal Radius");

  Rads_pore as RealParameter(value:3E-6/*FIND VALUE FOR THIS!*/, Description:"Mean radius of
Adsorbent Pores");
  Rcat_pore as RealParameter(value:1.635E-9, Description:"Mean radius of Catalyst Pores");

  alpha                     as Voidage Fraction;
  /*beta                      as Voidage_Fraction;
  gamma                     as Voidage_Fraction;
  delta                     as Voidage_Fraction;
  epsilon                   as Voidage_Fraction;
  zeta                      as Voidage_Fraction;*/


//---------------------------------- Domains
-------------------------------------------------------//

  Pellet                                            as   LengthDomain(Length: Rp_ads,
DiscretizationMethod:"BFD1", HighestOrderDerivative:2,SpacingPreference:(Rp_ads/NoR), Description:
"Radial Pellet Coordinate");
  Crystal                                           as   LengthDomain(Length: Rc,
DiscretizationMethod:"BFD1", HighestOrderDerivative:2,SpacingPreference:(Rc/NoR), Description:
"Radial Crystal Coordinate");

//---------------------------------- Variables
-------------------------------------------------------//

  Phi_ads1([0:Pellet.Endnode])                      as Voidage_Fraction(Spec:Free, Description:
"Volume Fraction of Adsorbent in the Pellet");
  Phi_cat1([0:Pellet.Endnode])                      as Voidage_Fraction(Spec:Free, Description:
"Volume Fraction of Catalyst in the Pellet");
  Phi_PCM1                                          as Voidage_Fraction(Spec:Fixed, value:0,
Description:"Volume Fraction of Adsorbent in the Pellet");

  q c(CompAll)                                      as Distribution1D (XDomain is Pellet,
Description: "Adsorbent Loading [kmol/m³ Adsorbent]", Integrals: "idx") of conc mole;
  //Pt_ads([0:Pellet.EndNode])                      as /*Distribution1D (XDomain is Pellet,
Description:"Pressure in pores [Bar]") of*/ Pressure;
  //Pt_cat                                          as Distribution1D (XDomain is Pellet,
Description:"Pressure in pores [Bar]") of Pressure;
  //Yi_ads(CompAll)([0:Pellet.EndNode])             as molefraction (description:"Component Mole
Fractions in the catalyst phase");
  //Yi_cat(CompAll)([0:Pellet.EndNode])             as molefraction (description:"Component Mole
Fractions in the catalyst phase");
  eps bed                                           as Voidage Fraction (description:"Reactor
Porosity-Catalyst Specific[-]", value:0.36, spec:Fixed);
  //eps_cat                                         as Voidage_Fraction (description:"Catalyst
Porosity[-]", value:0.53, spec:Fixed);
  rho_ads                                           as dens_mass (description:"Adsorbent
Density[kg/m³]", value:/*734*/1199.3, spec:Fixed);
  rho_cat                                           as dens_mass (description:"Cataylst
Density[kg/m³]", value:/*734*/3600, spec:Fixed);
  //eps_ads                                         as Voidage_Fraction (description:"Adsorbent
Porosity[-]", value:0.37, spec:Fixed); //same for both adsorbent and catalyst
  tort cat                                          as RealVariable(description:"Catalyst
Tortuosity[-]", value:5, spec:Fixed);
  tort_ads                                          as RealVariable(description:"Adsorbent
Tortuosity[-]", value:2, spec:Fixed);
  cond_ads                                          as RealVariable(description:"Adsorbent
Conductivity[kW/m/K]",0.00012,Fixed);
  cond_ads_eff/*([0:Pellet.EndNode])*/              as RealVariable(description:"Effective
Adsorbent Conductivity[BTU/hr/sq.ft/F]",Free);
  cond_ads_eff_W/*([0:Pellet.EndNode])*/            as RealVariable(description:"Effective
Adsorbent Conductivity[W/m/K]",Free);
  cond cat                                          as RealVariable(description:"Catalyst
Conductivity[kW/m/K]",0.00574,Fixed);
  cond_cat_eff/*([0:Pellet.EndNode])*/              as RealVariable(description:"Effective
Catalyst Conductivity[BTU/hr/sq.ft/F]",Free);
  cond_cat_eff_W/*([0:Pellet.EndNode])*/            as RealVariable(description:"Effective
Catalyst Conductivity[W/m/K]",Free);
  cp_ads                                            as RealVariable(description:"Adsorbent
Specific Heat Capacity [J/kg*K]",1.045,Fixed);
```

```
55    cp_cat                                          as RealVariable(description:"Catalyst
      Specific Heat Capacity [J/kg*K]",1.050,Fixed);
56    Cond_g_W/*([0:Pellet.EndNode])*/                as RealVariable(description:"Vapor Phase
      Conductivity[W/m/K]"); //this valus id same in adsorbent and catalyst model
57    Cond_g_BTU/*([0:Pellet.EndNode])*/              as RealVariable(description:"Vapor Phase
      Conductivity[BTU/hr/sq.ft/F]");//this valus id same in adsorbent and catalyst model
58    Cond_ads_BTU/*([0:Pellet.EndNode])*/            as RealVariable(description:"Adsorbent
      Conductivity[BTU/hr/sq.ft/F]");
59    Cond cat BTU/*([0:Pellet.EndNode])*/            as RealVariable(description:"Catalyst
      Conductivity[BTU/hr/sq.ft/F]");
60
61    //This value is taken from adsorbent and consider different variable for cataylst i.e. psi3,psi4
      ans so on
62    psi/*([0:Pellet.Endnode])*/                     as RealVariable(Free);
63    psi1/*([0:Pellet.Endnode])*/                    as RealVariable(Free);
64    psi2/*([0:Pellet.Endnode])*/                    as RealVariable(Free);
65    psi3                                            as RealVariable(Free);
66    psi4                                            as RealVariable(Free);
67    psi5                                            as RealVariable(Free);
68    //Theta                                         as RealVariable(Free);
69    n_contacts                                      as RealVariable(Free);
70
71    //this valus is same in both adorbent and catalyst model so there is no need to add variable again
      from catalyst model
72    D_mi(CompAll)                                   as RealVariable(Free, lower:0);
73    D_dustyeff(CompAll)([0:Pellet.EndNode])         as RealVariable(Free, lower:0);
74    D_dusty(CompAll)([0:Pellet.EndNode])            as RealVariable(Free, lower:0);
75
76    //Please consider different variable for catalyst
77    D_Knud(CompAll)([0:Pellet.Endnode])             as RealVariable(Free, lower:0);
78    D_ads(CompAll)([0:Pellet.Endnode])              as RealVariable;
79
80    D_Knud1(CompAll)([0:Pellet.Endnode])            as RealVariable(Free, lower:0); //for cat
81    D_cat(CompAll)([0:Pellet.Endnode])              as RealVariable;
82
83    D_eff([0:Pellet.Endnode])                       as RealVariable;
84
85    MolWeights(CompAll)                             as molweight (description:"Molecular Weights
      of the individual components[kg/kmol]", Spec:Fixed);
86
87    DeltaH_adsorption                               as RealVariable (description:"Heat of
      Adsorption[kJ/kmol]", value:-57950*1, spec:Fixed);//
88    DeltaH_reaction                                 as RealVariable (description:"Heat of
      Reaction[kJ/kmol]", value:-108600*1, spec:Fixed);//
89
90    Phi_crystals                                    as Voidage_Fraction(Spec:Fixed, Value: 0.8,
      Description:"Volume fraction of zeolite crystals in the adsorbent solid");
91
92    Total Loading Pellet                            as RealVariable;
93
94    Cond_eff                                        as RealVariable(description:"Pellet
      Conductivity[kW/m/K]",0.00011754,Fixed);
95
96    Total_reacrate_pellet                           as RealVariable;
97    Average_Loading_Pellet                          as RealVariable;
98
99    Total_adsrate_pellet                            as RealVariable;
100   //new variable include for one pellet and some paramters are assumeed
101   tort pellet                                     as RealVariable (description:"pellet
      Tortuosity[-]", value:5, spec:Fixed);
102   C_pellet(CompAll)                               as Distribution1D (XDomain is Pellet,
      Description: "Pellet Macropore Concentration [kmol/m³]") of conc_mole; //in this model
      C_adspellet=C_catpellet
103   Tk_pellet                                       as Distribution1D (XDomain is Pellet,
      Description:"pellet Temperature [K]") of Temperature_abs;
104   cond_pellet/*([0:Pellet.Endnode])*/             as RealVariable(description:
      "Adsorbent/catalyst mixture Conductivity[kW/m/K]",value:0.00041786,Fixed);
105   eps pellet                                      as Voidage Fraction (description:"Pellet
      Porosity[-]", value:0.37, spec:Fixed);
106   rho_pellet/*([0:Pellet.Endnode])*/              as dens_mass (description:
      "Adsorbent/cataylst mixture Density[kg/m³]",value:1326.5371, spec:Fixed);
107   cp_pellet/*([0:Pellet.Endnode])*/               as RealVariable(description:
      "Adsorbent/catalyst mixture Specific Heat Capacity [J/kg*K]", value:1.045265,Fixed);
108   D_eff_pellet(CompAll)([0:Pellet.Endnode])       as RealVariable;
109   Ct_pellet([0:Pellet.Endnode])                   as conc_mole;
110   Tc_pellet([0:Pellet.EndNode])                   as /*Distribution1D (XDomain is Pellet,
      Description:"Catalyst Temperature [C]") of*/ Temperature;
111   Pt_pellet([0:Pellet.EndNode])                   as /*Distribution1D (XDomain is Pellet,
      Description:"Pressure in pores [Bar]") of*/ Pressure;
112   Yi_pellet(CompAll)([0:Pellet.EndNode])          as molefraction (description:"Component Mole
      Fractions in the catalyst phase");
113   Tk_pellet_avg                                   as RealVariable;
114
115   Theile_Modulus([0:Pellet.EndNode])              as Realvariable;
116   Effectiveness_Factor([0:Pellet.EndNode])        as Realvariable;
```

```
117     Rate([0:Pellet.EndNode])                                as Realvariable;
118     C_pelletmix([0:Pellet.EndNode])                         as RealVariable;
119
120   //=================================== EQUATIONS
      ================================================//
121
122   //Phi_ads1=0.43;
123
124
125   Phi ads1([0:11])=alpha;
126   /*
127   Phi_ads1([2:3])=beta;
128   Phi_ads1([4:5])=gamma;
129   Phi_ads1([6:7])=delta;
130   Phi_ads1([8:9])=epsilon;
131   Phi_ads1([10:11])=zeta;
132   */
133
134   //Phi PCM1([0:5])=0;
135
136
137   For r in ([0:Pellet.EndNode]) do
138     Phi_ads1(r)+Phi_cat1(r)/*+Phi_PCM1(r)*/=1;
139   EndFor
140
141   For r in [0:Pellet.Endnode] do
142
      //Yi ads(CompAll)(r)=(Phi ads1*C pellet(CompAll)(r))/(SIGMA(phi ads1*C pellet(CompAll)(r))+1E-5)+1E-1
      ;
143       Yi_pellet(CompAll)(r)=C_pellet(CompAll)(r)/(SIGMA(C_pellet(CompAll)(r))+1E-5)+1E-10;
144       //Pt_ads(r)=SIGMA(Phi_ads1*C_pellet(CompAll)(r))*1000*8.314E-5*Tk_pellet(r);
145       Pt_pellet(r)=SIGMA(C_pellet(CompAll)(r))*1000*8.314E-5*Tk_pellet(r);
146       Tc_pellet(r)=abs(Tk_pellet(r))-273.15;
147       //Yi_pellet(CompAll)(r)=C_pellet(CompAll)(r)/(SIGMA(C_pellet(CompAll)(r))+1E-5)+1E-10;
148       //Yi_cat(CompAll)(r)=(Phi_cat1*C_pellet(CompAll)(r))/(Ct_pellet(r)+1E-5)+1E-10;
149       //Pt_ads(r)=SIGMA(C_pellet(CompAll)(r))*1000*8.314E-5*Tk_ads(r);
150       //Pt_cat(r)=Ct_pellet(r)*1000*8.314E-5*Tk_pellet(r);
151       //Tc_ads(r)=abs(Tk_ads(r))-273.15;
152       //Ct_pellet(r)=Phi_cat1*SIGMA(C_pellet(CompAll)(r));
153       //Tc_cat(r)=Tk_cat(r)-273.15;
154       //Yi_cat(CompAll)(r)=C_pellet(CompAll)(r)/(Ct_pellet(r)+1E-5)+1E-10;
155       //Pt_cat(r)=Ct_pellet(r)*1000*8.314E-5*Tk_pellet(r);
156       //Ct_pellet(r)=SIGMA(C_pellet(CompAll)(r));
157       //Tc_cat(r)=Tk_cat(r)-273.15;
158       //C_pellet(CompAll)(r)=(Phi_ads1*C_adspellet(CompAll)(r))+(Phi_cat1*C_catpellet(CompAll)(r));
159       //Tk_pellet(r)=(Phi_ads1*Tk_ads(r))+(Phi_cat1*Tk_cat(r));
160       //C_adspelletmix(r)=Sigma(Yi_ads(CompAll)(r)*C_adspellet(CompAll)(r));
161       //C_catpelletmix(r)=Sigma(Yi_cat(CompAll)(r)*C_catpellet(CompAll)(r));
162       C_pelletmix(r)=Sigma(Yi_pellet(CompAll)(r)*C_pellet(CompAll)(r));
163       //Theile_Modulus(r)= (r/3)*sqrt((Reaction(r).Rate)/((D_eff_pellet(r))*C_pelletmix(r)));
164     EndFor
165
166     MolWeights("H2S"):34.0809;
167     MolWeights("O2S"):64.066;
168     MolWeights("S8"):8*32;
169     MolWeights("H2O"):18;
170     MolWeights("N2"):14.0067;
171
172   //--------------------------------- Sub-Model Declaration ----------------------------------------/
173
174     Zeolite([0:Pellet.Endnode])                             as Zeolite;
175     Reaction([0:Pellet.Endnode])                            as Reaction;
176
177   //--------------------------------- Sub-Model Connections ----------------------------------------/
178
179   For r in [0:Pellet.Endnode] do
180     /*
181       For c in [0:Crystal.Endnode] do
182         Zeolite(r).Adsorption(c).Tk_ads=Tk_ads(r);
183         Zeolite(r).Adsorption(c).Pt_ads=Pt_ads(r);
184         Zeolite(r).Adsorption(c).Yi_ads(CompAll)=Yi_ads(CompAll)(r);
185       EndFor
186   */
187       Zeolite(r).Tk_pellet=abs(Tk_pellet(r));
188       Zeolite(r).Pt_pellet=Pt_pellet(r);
189       Zeolite(r).Yi_pellet(CompAll)=Yi_pellet(CompAll)(r);
190
191     EndFor
192
193   For r in [0:Pellet.Endnode] do
194
195       Reaction(r).Tk_pellet=Tk_pellet(r);
196       Reaction(r).Pt_pellet=Pt_pellet(r);
197       Reaction(r).Yi_pellet(CompAll)=Yi_pellet(CompAll)(r);
198       Reaction(r).Rate=Rate(r);
```

```
199       Reaction(r).Phi_cat1=Phi_cat1(r);
200
201     EndFor
202
203
204   //---------------------------------- Property Calculations
      ----------------------------------------//
205
206   //this is same for both
207   Cond g BTU=1/1.7307*Cond g W;
208
209   //adsorbent calc
210   Cond_ads_BTU=Cond_ads*1000*1/1.7307;
211   Cond_ads_eff=Cond_g_BTU*(eps_bed+(0.895*(1-eps_bed))/(psi+2/3*(Cond_g_BTU/Cond_ads_BTU)));
212   psi=psi2+(psi1+psi2)*(eps_bed-0.26)/0.216;
213   psi1=0.5*2/3*((Cond_ads_BTU/Cond_g_BTU-1)/(Cond_ads_BTU/Cond_g_BTU))^2*(sin(0.9953))^2/((Cond_ads_BTU
      /Cond_g_BTU*LOGe(Cond_ads_BTU/Cond_g_BTU-(Cond_ads_BTU/Cond_g_BTU-1)*cos(0.9953))-((Cond_ads_BTU/
      Cond g BTU-1)/(Cond ads BTU/Cond g BTU)*(1-cos(0.9953)))));
214   psi2=0.5*2/3*((Cond ads BTU/Cond g BTU-1)/(Cond ads BTU/Cond g BTU))^2*(sin(0.3897))^2/((Cond ads BTU
      /Cond g BTU*LOGe(Cond ads BTU/Cond g BTU-(Cond ads BTU/Cond g BTU-1)*cos(0.3897))-((Cond ads BTU/
      Cond_g_BTU-1)/(Cond_ads_BTU/Cond_g_BTU)*(1-cos(0.3897)))));
215   Cond_ads_eff_W=Cond_ads_eff/(1000*1/1.7307); //[kW/m/K]
216
217   //catalyst calc
218   Cond_cat_BTU=Cond_cat*1000*1/1.7307;
219   Cond_cat_eff=Cond_g_BTU*(eps_bed+(0.895*(1-eps_bed))/(psi+2/3*(Cond_g_BTU/Cond_cat_BTU)));
220   psi3=psi5+(psi4+psi5)*(eps_bed-0.26)/0.216;
221   psi4=0.5*2/3*((Cond cat BTU/Cond g BTU-1)/(Cond cat BTU/Cond g BTU))^2*(sin(0.9953))^2/((Cond cat BTU
      /Cond g BTU*LOGe(Cond cat BTU/Cond g BTU-(Cond cat BTU/Cond g BTU-1)*cos(0.9953))-((Cond cat BTU/
      Cond_g_BTU-1)/(Cond_cat_BTU/Cond_g_BTU)*(1-cos(0.9953)))));
222   psi5=0.5*2/3*((Cond_cat_BTU/Cond_g_BTU-1)/(Cond_cat_BTU/Cond_g_BTU))^2*(sin(0.3897))^2/((Cond_cat_BTU
      /Cond_g_BTU*LOGe(Cond_cat_BTU/Cond_g_BTU-(Cond_cat_BTU/Cond_g_BTU-1)*cos(0.3897))-((Cond_cat_BTU/
      Cond_g_BTU-1)/(Cond_cat_BTU/Cond_g_BTU)*(1-cos(0.3897)))));
223   Cond_cat_eff_W=Cond_cat_eff/(1000*1/1.7307); //[W/m/K]
224
225
226
227   //-------------------- Calculating effective diffusion coefficients in multipellet
      ----------------------//
228
229   For r in [0:Pellet.Endnode] do
230
231   //this values remain same for both
232
233   D_dusty("H2S")(r)=(D_mi("H2S"))/*(1-Yi_ads("H2S")(r))*(Yi_ads("O2S")(r)/(D_mol("H2S")("O2S")(r))
234                                          + Yi_ads("S8")(r)/(D_mol("H2S")("S8")(r))
235                                          + Yi_ads("H2O")(r)/(D_mol("H2S")("H2O")(r))
236                                          + Yi_ads("N2")(r)/(D_mol("H2S")("N2")(r)))*/;
237
238    D_dusty("O2S")(r)=(D_mi("O2S"))/*(1-Yi_ads("O2S")(r))*(Yi_ads("H2S")(r)/(D_mol("O2S")("H2S")(r))
239                                          + Yi_ads("S8")(r)/(D_mol("O2S")("S8")(r))
240                                          + Yi_ads("H2O")(r)/(D_mol("O2S")("H2O")(r))
241                                          + Yi_ads("N2")(r)/(D_mol("O2S")("N2")(r)))*/;
242
243     D_dusty("S8")(r)=(D_mi("S8"))/*(1-Yi_ads("S8")(r))*(Yi_ads("O2S")(r)/(D_mol("S8")("O2S")(r))
244                                          + Yi_ads("H2S")(r)/(D_mol("S8")("H2S")(r))
245                                          + Yi_ads("H2O")(r)/(D_mol("S8")("H2O")(r))
246                                          + Yi_ads("N2")(r)/(D_mol("S8")("N2")(r)))*/;
247
248     D_dusty("H2O")(r)=(D_mi("H2O"))/*(1-Yi_ads("H2O")(r))*(Yi_ads("O2S")(r)/(D_mol("H2O")("O2S")(r))
249                                          + Yi_ads("S8")(r)/(D_mol("H2O")("S8")(r))
250                                          + Yi_ads("H2S")(r)/(D_mol("H2O")("H2S")(r))
251                                          + Yi_ads("N2")(r)/(D_mol("H2O")("N2")(r)))*/;
252
253    D_dusty("N2")(r)=(D_mi("N2"))/*(1-Yi_ads("N2")(r))*(Yi_ads("O2S")(r)/(D_mol("N2")("O2S")(r))
254                                          + Yi_ads("S8")(r)/(D_mol("N2")("S8")(r))
255                                          + Yi_ads("H2O")(r)/(D_mol("N2")("H2O")(r))
256                                          + Yi_ads("H2S")(r)/(D_mol("N2")("H2S")(r)))*/;
257
258     D_dustyeff(CompAll)(r)=D_dusty(CompAll)(r);
259
260   //please consider different variable for catalyst.
261     D_Knud(CompAll)(r)=9700*(Rads_pore*100)*SQRT(abs(Tk_pellet(r))/MolWeights(CompAll))/10000;
262
263   //1/D_ads(CompAll)(r)=1/D_knud(CompAll)(r)+1/D_dustyeff(CompAll)(r);
264     D_ads(CompAll)(r)=eps_pellet/tort_ads*1/((D_knud(CompAll)(r)+D_dustyeff(CompAll)(r))/(D_knud(
      CompAll)(r)*D_dustyeff(CompAll)(r)));
265
266   //for catalyst
267
268     D_Knud1(CompAll)(r)=9700*(Rcat_pore*100)*SQRT(abs(Tk_pellet(r))/MolWeights(CompAll))/10000*100; //
      Parameter sensitivity check : Removing order-of-magnitude effect on overall diffusivity
269     D_cat(CompAll)(r)=eps_pellet/tort_cat*1/((D_knud1(CompAll)(r)+D_dustyeff(CompAll)(r))/(D_knud1(
      CompAll)(r)*D_dustyeff(CompAll)(r)));
270     D_eff_pellet(CompAll)(r)= (eps_pellet/tort_pellet)*(1/((/*Phi_ads1*/1/(D_Knud(CompAll)(r)))+(
```

```
    /*Phi_cat1*/1/(D_Knud1(CompAll)(r)))+ (1/(D_dustyeff(CompAll)(r)+1E-10))));
271
272     //D_eff_pellet(CompAll)(r)=
    /*(eps_pellet/tort_pellet)*/(1/((/*Phi_ads1*/1/(D_ads(CompAll)(r)))+(/*Phi_cat1*/1/(D_cat(CompAll)(r)
    )/*+ (1/(D_dustyeff(CompAll)(r)+1E-10)))*/));
273   D_eff(r)=Sigma((Yi_pellet(CompAll)(r))*D_eff_pellet(CompAll)(r));
274   EndFor
275
276   //---------------------------------------- PDES
    ----------------------------------------------------//
277
278   //---------------------------------- Component Balances
    -------------------------------------------//
279
280   For r in [Pellet.Interior] do
281   //modify this equation after adding catlyst model variable in this model
282   //$C_adspellet(CompAll)(r)=D_ads(CompAll)(r)*(C_adspellet(CompAll)(r).d2dX2+2/r*C_adspellet(CompAll)(
    ).ddX)-((1-eps pellet)/eps pellet)*Phi crystals*Phi ads1*Zeolite(r).Rate ads(CompAll);
283   //$C catpellet(CompAll)(r)=D cat(CompAll)(r)*(C catpellet(CompAll)(r).d2dX2+2/r*C catpellet(CompAll)(
    ).ddX)+((1-eps pellet)/eps pellet)*Phi cat1*Reaction(r).Rx(CompAll);
284     $C_pellet(CompAll)(r)=D_eff_pellet(CompAll)(r)*(C_pellet(CompAll)(r).d2dX2+2/r*C_pellet(CompAll)(r
    ).ddX)-(1-eps_pellet)/eps_pellet*(Phi_crystals*Phi_ads1(r)*Zeolite(r).Rate_ads(CompAll)- Phi_cat1(r)*
    Reaction(r).Rx(CompAll));
285   EndFor
286
287   //---------------------------------------- Heat Balance
    ------------------------------------------------//
288
289   //For r in [0:Pellet.EndNode] do
290   //rho_pellet=(/*phi_ads1*rho_ads)+(phi_cat1(r)*rho_cat);
291   //Cp_pellet= (phi_ads1*Cp_ads)+(phi_cat1(r)*Cp_cat);
292   //Cond_pellet=(phi_ads1*Cond_ads)+(phi_cat1(r)*Cond_cat);
293
294   //EndFor
295     For r in [Pellet.Interior] do
296
    //rho_ads*Cp_ads*$Tk_ads(r)=Cond_ads*(2/r*Tk_ads(r).ddX+Tk_ads(r).d2dX2)-Phi_crystals*Phi_ads1*Zeolit
    (r).Rate ads("H2O")*DeltaH adsorption;
297
    //rho_cat*Cp_cat*$Tk_cat(r)=Cond_cat*(2/r*Tk_cat(r).ddX+Tk_cat(r).d2dX2)-Phi_cat1*Reaction(r).Rate*De
    taH_reaction;
298       (rho_pellet*Cp_pellet)*$Tk_pellet(r)=Cond_pellet*(2/r*Tk_pellet(r).ddX+Tk_pellet(r).d2dX2)-(
    Phi_crystals*Phi_ads1(r)*Zeolite(r).Rate_ads("H2O")*DeltaH_adsorption + Phi_cat1(r)*Reaction(r).Rate*
    DeltaH_reaction);
299     EndFor
300
301   //-------------------------------- Performance Indices ----------------------------------------//
302
303   For r in [0:Pellet.Endnode] do
304   //include functionalties here and change variable in the graph too
305     q_c(CompAll)(r)=/*Phi_ads1*/Zeolite(r).q_c(CompAll);
306   EndFor
307
308   For r in [0:Pellet.Endnode] do  //Calculation of the radial location is better than inserting it as
    fixed parameter and calculate 6 equation (below)
309     Theile_Modulus(r)= (((Rp_ads/5*r)/3)*sqrt(abs(Rate(r))/((D_eff(r))*C_pelletmix(r))))+1E-10;
310     Effectiveness_Factor(r)= (3/(Theile_Modulus(r)*Theile_Modulus(r)))*(((Theile_Modulus(r))*(1/tanh(
    Theile Modulus(r))))-1);
311   EndFor
312
313   //For r in [Pellet.Interior] do
314      /* Theile_Modulus(0)= ((0/3)*sqrt(abs(Rate(0))/((D_eff(0))*C_pelletmix(0))))+1E-10;
315       Theile_Modulus(1)= (3.57E-4/3)*sqrt(abs(Rate(1))/((D_eff(1))*C_pelletmix(1)));
316       Theile_Modulus(2)= (7.14E-4/3)*sqrt(abs(Rate(2))/((D_eff(2))*C_pelletmix(2)));
317       Theile_Modulus(3)= (0.001071/3)*sqrt(abs(Rate(3))/((D_eff(3))*C_pelletmix(3)));
318       Theile_Modulus(4)= (0.001428/3)*sqrt(abs(Rate(4))/((D_eff(4))*C_pelletmix(4)));
319       Theile_Modulus(5)= (0.001785/3)*sqrt(abs(Rate(5))/((D_eff(5))*C_pelletmix(5)));
320       Effectiveness Factor(0)=
    (3/(Theile Modulus(0)*Theile Modulus(0)))*(((Theile Modulus(0))*(1/tanh(Theile Modulus(0))))-1);
321       Effectiveness_Factor(1)=
    (3/(Theile_Modulus(1)*Theile_Modulus(1)))*(((Theile_Modulus(1))*(1/tanh(Theile_Modulus(1))))-1);
322       Effectiveness_Factor(2)=
    (3/(Theile_Modulus(2)*Theile_Modulus(2)))*(((Theile_Modulus(2))*(1/tanh(Theile_Modulus(2))))-1);
323       Effectiveness_Factor(3)=
    (3/(Theile_Modulus(3)*Theile_Modulus(3)))*(((Theile_Modulus(3))*(1/tanh(Theile_Modulus(3))))-1);
324       Effectiveness_Factor(4)=
    (3/(Theile_Modulus(4)*Theile_Modulus(4)))*(((Theile_Modulus(4))*(1/tanh(Theile_Modulus(4))))-1);
325       Effectiveness Factor(5)=
    (3/(Theile Modulus(5)*Theile Modulus(5)))*(((Theile Modulus(5))*(1/tanh(Theile Modulus(5))))-1); */
326   //EndFor
327
328     Total_Loading_Pellet=SIGMA(Zeolite([0:Pellet.EndNode]).q_c("H2O")/*.idx*/);
329     Average_Loading_Pellet=Total_Loading_Pellet/NoR;
330
331     Total_reacrate_Pellet=SIGMA(Reaction([0:Pellet.Endnode]).Rate);
```

```
332
333     Total_adsrate_Pellet=SIGMA(Zeolite([0:Pellet.Endnode]).Rate_ads("H2O"));
334     Tk_pellet_avg=SIGMA(Tk_pellet)/(NoR+1);
335
336
337
338   //------------------------------- Initial Conditions
      ------------------------------------------------//
339
340     For r in [Pellet.Interior] do
341       C_pellet("H2S")(r):0, Initial;
342       C_pellet("H2O")(r):0, Initial;
343       C_pellet("O2S")(r):0, Initial;
344       C_pellet("S8")(r):0, Initial;
345       C_pellet("N2")(r):0.023295947, Initial;
346       /*C_catpellet("H2S")(r):0, Initial;
347       C_catpellet("H2O")(r):0, Initial;
348       C_catpellet("O2S")(r):0, Initial;
349       C_catpellet("S8")(r):0, Initial;
350       C_catpellet("N2")(r):0.023295947, Initial;
351       Tk_ads(r):523.15, Initial;*/
352       Tk_pellet(r):523.15, Initial;
353
354     EndFor
355
356   End
```

```
Model Zeolite

//----------------------------------Components-----------------------------------------------------
/
  CompAll   as StringSet(["H2S","O2S","S8","H2O","N2"]);
  CompA     as StringSet(["H2O"]);// Adsorbing Species List

//---------------------------------- Parameters
-----------------------------------------------------//
   NoR       as IntegerParameter (Description:"Number Of Radial Nodes", 5);
   Rc        as RealParameter(Value:0.5E-6, Description:"Mean Zeolite Crystal Radius [m]");

//---------------------------------- Domains
-----------------------------------------------------//
  //Crystal as   LengthDomain(Length: Rc, DiscretizationMethod:"BFD1",
HighestOrderDerivative:2,SpacingPreference:(Rc/NoR), Description:"Radial Crystal Coordinate");

//---------------------------------- Variables
-----------------------------------------------------//
  //q_c(CompAll)                              as Distribution1D(XDomain is Crystal,
description:"Zeolite Crystal Loading [kmol/m³]") of RealVariable;//conc_mole;
  q_c(CompAll)                              as RealVariable;
  //q_ctimescsquared([0:Crystal.Endnode]) as /*Distribution1D(XDomain is Crystal,
description:"Zeolite Crystal Loading [kmol/m³]", integrals:"idx", HighestOrderXDerivative: 0) of*/
RealVariable;
  Rate_ads(CompAll)                         as RealVariable;
  D_mu0                                     as Diffus_vap(Value:6.3825E-14, description:"Micropore
diffusivity of water [m^2/s]", spec:Fixed);
  Ea_diff                                   as RealVariable (description:"Activation energy for
micropore diffusion [kJ/kmol]", value:27600*1, spec:Fixed);//

  Yi_pellet(CompAll)                          as molefraction;
  Pt_pellet                                   as pressure         (description:"Reactor
Pressure[bar]");
  TK pellet                                   as temperature abs  (description:"Temperature[K]") ;
  Pw                                        as Pressure         (description:"Partial Pressure Of
Water[Bar]");
  Binf                                      as RealVariable     (description:"Maximum Adsorption
Affinity [K^0.5/bar]", value:5.3126E-5, spec:Fixed,scale:1e-8);//--> 1[Pa]=0.01[mbar]
  T0                                        as  temperature_abs (description:"Adsorption Reference
Temperature [K]", value:300, spec:Fixed);
  q0w                                       as conc_mole        (description:"Loading Correction
[kmol/m³-Adsorbent]", value:12.88599878/*8.05845*/, spec:Fixed);
  gamma                                     as RealVariable     (description:"Adsorption Parameter
[-]", value:23.235, spec:Fixed);
  delta                                     as  RealVariable    (description:"Adsorption Parameter
[-]", value:0.68792, spec:Fixed);
  qsw                                       as conc_mole        (description:"Saturation Loading
Capacity [kmol/m³-Adsorbent]");
  B                                         as RealVariable     (description:"Adsorption
Affinity[1/bar]",scale:0.1);
  //qstr                                      as conc_mole      (description:"Equillibrium Loading
[kmol/m³-Adsorbent]");

  SwitchParameter                           as RealVariable(Fixed, Value:1);

  Approach_to_ads_eq                        as RealVariable;

```

```
86
87        $q_c("H2O")=15/Rc^2*D_mu0*exp(-Ea_diff/(8.314*TK_pellet))*(qsw/(qsw-q_c("H2O")))*(B*qsw*Pw/(1+
      B*Pw)-q_c("H2O"));
88
89
90        $q_c("H2S")=0;
91        $q_c("O2S")=0;
92        $q_c("N2")=0;
93        $q_c("S8")=0;
94
95        Rate_ads("H2O")=$q_c("H2O");
96        Rate_ads("H2S")=0;
97        Rate_ads("O2S")=0;
98        Rate_ads("N2")=0;
99        Rate_ads("S8")=0;
100
101       EqAD0:  Pw=Yi_pellet("H2O")*Pt_pellet;
102       EqAD1:  qsw=q0w*EXP(delta*(1-(TK_pellet/T0))); //[kmol/m³-Adsorbent] -> No change
103       EqAD2:  B=(Binf/SQRT(TK_pellet))*EXP(gamma*T0/TK_pellet);// 1/bar  -> No change
104       //EqAD3:  qstr=B*qsw*Pw/(1+ B*Pw);//[mol/kg-Adsorbent]  -> No change
105       //EqAD3:
      qstr=((Binf/SQRT(TK_pellet))*EXP(gamma*T0/TK_pellet))*(q0w*EXP(delta*(1-(TK_pellet/T0))))*(Yi_pellet(
      H2O")*Pt_pellet)/(1+
      ((Binf/SQRT(TK_pellet))*EXP(gamma*T0/TK_pellet))*(Yi_pellet("H2O")*Pt_pellet));//[mol/kg-Adsorbent]
      -> No change
106
107   //-------------------------------- Performance Indices ----------------------------------------//
108
109     qsw*Approach_to_ads_eq=q_c("H2O");
110
111   //-------------------------------- Initial Conditions
      ------------------------------------------------//
112
113     //q_c(CompAll)([Crystal.Interior]):0, Initial;
114     q_c(CompAll):0, Initial;
115     //q_c("H2O")(0):0, Initial;
116     //q_p(CompAll):0, Initial;
117     //q_ctimescsquared([0:Crystal.Endnode]):0, Initial;
118     //Rate_ads("H2O"):0, Initial;
```

```
Model Reaction

//Connection Variables with ADR2D Model
  CompAll               as StringSet(["H2S","O2S","S8","H2O","N2"]);
  Pt_cat                as pressure(description:"Reactor Pressure[bar]");
  T_cat                 as temperature_abs ;
  Yi_cat(CompAll)       as molefraction (description:"Component Mole Fractions");

// Reaction Specific Variables
  Ea1         as hidden Activation Energy (description:"Forward Reaction[J/mol]", value:49980, spec:
Fixed);
  Ea2         as hidden Activation_Energy (description:"Backward Reaction[J/mol]", value:86601, spec:
Fixed);
  rho_cat     as hidden dens_mass         (description:"Catalyst Density[kg/m³]", value:3600, spec:
Fixed);
  k1          as RealVariable;
  k2          as RealVariable;
  SwitchParameter as RealVariable(Fixed, 1);
  Approach_to_eq  as RealVariable;

// Reaction Rate Computation
  Rx(CompAll)     as  Reaction_Rate(Description:"Rate [kmol/m³/s]");
  Rate            as  Reaction_Rate(Description:"Rate [kmol/m³/s]");

// Claus Reaction Rate
k1=17.12457/*(5.292e-3)*/*EXP((-Ea1)/(8.314*T_cat));
k2=1168.434385/*(1.252)*/*EXP((-Ea2)/(8.314*T_cat));//(1e-3)*

Switch

  INITIAL STATE REACTION

    IF ((Yi_cat("H2S")>=0) and (Yi_cat("O2S")>=0) and (Yi_cat("H2O")>=0)) THEN//
      Rate=(k1*((Yi_cat("H2S")*Pt_cat)^0.95)*((Yi_cat("O2S")*Pt_cat)^0.22)-k2*(Yi_cat("H2O")*Pt_cat
))*rho_cat;//[mol/(kg-catalyst)/s]*[kg-catalyst*m³]*[m³/mol]=[kmol/m³/s]
    ELSE
      Rate=1E-10;
    ENDIF

    IF (SwitchParameter >= 0 and SwitchParameter < 1) STATE : NOREACTION;

  ENDSTATE

  STATE NOREACTION

    IF ((Yi_cat("H2S")>=0) and (Yi_cat("O2S")>=0) and (Yi_cat("H2O")>=0)) THEN//
      Rate=1E-10;//[mol/(kg-catalyst)/s]*[kg-catalyst*m³]*[m³/mol]=[kmol/m³/s]
    ELSE
      Rate=1E-10;
    ENDIF

    IF (SwitchParameter > 0 and SwitchParameter <= 1) STATE : REACTION;

  ENDSTATE

ENDSWITCH

   (k2*(Yi cat("H2O")*Pt cat)) * Approach to eq = (k1*((Yi cat("H2S")*Pt cat)^0.95)*((Yi cat("O2S")*
Pt cat)^0.22));

// Component Specific Reaction Rates: 2H2S + SO2 + N2 <==> (3/8)S8 + 2H2O + N2
  Rx("H2S") =   -2*Rate;// Consumption
  Rx("O2S") =     -Rate;// Consumption
  Rx("H2O") =    2*Rate;// Generation
  Rx("S8")  =(3/8)*Rate;// Generation
  Rx("N2")  =0;

End
```

```
Model TransportProperties


//Conection Parameters With The Model ADR2D
CompAll   as StringSet(["H2S","O2S","S8","H2O","N2"]);
u         as RealVariable(description:"True Velocity");
//D_mix as RealVariable;
D_mi(CompAll)  as RealVariable;
Rho_g     as dens_mass_vap;
Visc_Bar  as Viscosity_BarS;
Visc_Pas  as Realvariable(description:"Gas Phase Viscosity[kg/m/s]");
Cp_Vap_Mass       as RealVariable(description:"Mass Heat Capacity of Gas Phase[kJ/kg/K]");
Eps   as Voidage_Fraction;
//Phi as Voidage_Fraction;
//Cond_Cat  as RealVariable(description:"Catalyst Conductivity[kW/m/K]",0.00574,Fixed);
//Cond_Ads  as RealVariable(description:"Adsorbent Conductivity[kW/m/K]",0.0006,Fixed);
//Cond_s    as RealVariable;
Cond_g    as RealVariable(description:"Vapor Phase Conductivity[kW/m/K]");
//Declaring Reactor Scale Parameters

Dp        as length       (description:"Average Particle Diameter [m]", value:1.785E-3*2, spec:Fixed,
scale:1e-3);
D_Reactor as RealVariable (description:"Reactor Diameter [m]", value:0.06, spec:Fixed,scale:1e-2);

Phi_cat as RealVariable;
Phi_ads as RealVariable;
Phi_PCM as RealVariable;


N         as RealVariable;
K         as realVariable;
M         as RealVariable(Description:"Der Parameter");
a1        as Realvariable;
a2        as Realvariable;
B         as RealVariable;

Pe_mz     as RealVariable(Description:"Axial Mass Peclet Number");
Pe_hz     as RealVariable(Description:"Axial Heat Peclet Number");
Re_p      as RealVariable(Description:"Particle Reynolds Number");
Sc(CompAll)  as RealVariable(Description:"Schmidt Number");
Sc_total     as RealVariable(Description:"Schmidt Number");
Sh(CompAll)  as RealVariable(Description:"Sherwood Number");
Pr        as RealVariable(Description:"Prandtl Number");

D_ax(CompAll)      as Dispersion;
L_ax      as Dispersion;
L_ax1     as RealVariable;
L_ax2     as RealVariable;
D_ar      as Realvariable;
L_ar      as Realvariable;
Kwall     as Realvariable;
h_fs      as Realvariable;
k_fs(CompAll) as RealVariable;
//k_fstotal as RealVariable;
Nu        as Realvariable;
Visc_Pas=Visc_Bar*1e5;

//Calculating Solid Phase Conductivity - Rayleigh's Law
//Cond_s/Cond_Cat=1+(3*Phi)/(((Cond_Ads+2*Cond_Cat)/(Cond_Ads-Cond_Cat))-(Phi)+1.569*((Cond_Ads-Cond_
at)/(3*Cond_Ads-4*Cond_Cat))*(Phi^(10/3)));

//Calculating DimensionLess Numbers
Pr     = abs(Visc_Pas*Cp_Vap_Mass/(Cond_g));
Re_p   = rho_g*Dp*u/Visc_Pas;
Pe_hz  = abs(Re_p)*abs(Pr);
Nu     = (2+1.1*((abs(Pr))^(1/3))*((abs(Re_p))^(0.6)));
Sc(CompAll) = abs(Visc_Pas/(Rho_g*(D_mi(CompAll)+1E-10)));
//Sc_total=Visc_Pas/(Rho_g*D_mix);

// Axial Mass Dispersion Coefficient - D_ax
D_ax(CompAll)    =  (D_mi(CompAll))/eps* (20 + 0.5*(abs(Sc(CompAll)))*(abs(Re_p)));

// Axial Heat Dispersion Coefficient - L_ax
//L_ax *1000   =
(0.73/((abs(Re_p))*(abs(Pr))))+(0.5/(1+(9.7/(((abs(Re_p))*(abs(Pr)))))))*(u*Rho_g*Cp_Vap_Mass*1000*Dp
;// because this has to be kept in [Kw/m/K]
L_ax=  (7+0.5*abs(Pr)*abs(Re_p))*Cond_g;

// Fluid To Solid Heat Transfer Coefficient - H_fs
h_fs=(2+1.1*((abs(Pr))^(1/3))*((abs(Re_p))^(0.6)))*Cond_g/Dp;

// Fluid to Solid Mass Transfer Coefficient
k_fs(CompAll)=(2+1.1*(abs(Sc(CompAll)))^(1/3)*(abs(Re_p))^(0.6))*D_mi(CompAll)/Dp;
//k_fstotal=2+1.1*Sc_total^(1/3)*Re_p^(0.6)*D_mix/Dp;
Sh(CompAll)= k_fs(CompAll)*Dp/D_mi(CompAll);
```

APPENDIX C: MULTI-DIMENSIONAL FUNCTIONALITY DISTRIBUTION

The two-dimensional pseudo-homogeneous dispersed model equations written in ACM code is presented below.

```
Model ADR2D

//-------------------------------------------------------------------------------------------------------------
//================================BEGIN SIMULATION=================================================//
//-------------------------------------------------------------------------------------------------------------

//========================= Defining ComponentLists =========================//
  CompAll   as StringSet(["H2S","O2S","S8","H2O","N2"]);
  CompR     as StringSet(["H2S","O2S","S8","H2O"]);//Reacting Species List
  CompRnA   as StringSet(["H2S","O2S","S8"]);//Reacting Not Adsorbing Species List
  CompA     as StringSet(["H2O"]);// Adsorbing Species List
  SumStoiC  as RealParameter(Description:"Sum of Stoichiometric Coefficients",Value:-5/8);
  Fin as realvariable;
  //========================= Defining Geometric Parameters Reactor Specific ========================//

  Length    as length       (description:"Reactor Length - Axial Co-ordinate [m]", value:1, spec:Fixed);
  Diameter  as length       (description:"Reactor Diameter - Radial Co-ordinate [m]", value:0.06, spec:Fixed);
  pi        as RealParameter(value:355/113);
  Acs       as area         (description:"Cross Sectional Area of the reactor-Normal to the flow [m²]");
  Dp        as length       (description:"Average Particle Diameter [m]", value:3.6E-3, spec:Fixed, scale:1e-3);

  //========================= Numerical Handle For Geometry =========================//
  NoL       as IntegerParameter (Description:"Number Of Axial Nodes", 50);
  NoR       as IntegerParameter (Description:"Number Of Radial Nodes", 8);

//========================= Declaring Domain =========================//
  Axial as    LengthDomain(Length: Length, DiscretizationMethod:"BFD1", HighestOrderDerivative:2, SpacingPreference:(Length/NoL), Description:"Axis Of The Reactor");
  Radial as    LengthDomain(Length: Diameter/2, DiscretizationMethod:"BFD1", HighestOrderDerivative:2,SpacingPreference:(Diameter/NoR/2), Description:"Axis Of The Reactor");

//========================= Variable Decleration - Phase Ratios For Catalyst/Adsorbent/Gas ========================//
eps       as  Voidage_Fraction (description:"Reactor Porosity-Catalyst Specific[-]", value:0.36, spec:fixed);//
  Phi([0:Axial.EndNode],[0:Radial.EndNode])       as  Voidage_Fraction (description:"Adsorbent Fraction in the solid Phase[-]", spec:fixed);//,value:0.43, spec:free
  f1, f2 as  RealVariable;
  f3([0:Axial.EndNode],[0:Radial.EndNode]) , Fc_cb([0:Axial.EndNode],[0:Radial.EndNode]) , fc_eb([0:Axial.EndNode],[0:Radial.EndNode]) , Fa_cb([0:Axial.EndNode],[0:Radial.EndNode]) , Fa_eb([0:Axial.EndNode],[0:Radial.EndNode])  as  RealVariable;



For X in [6:45] do
 /* For R in [0:2] do
    Phi([X,R]) : 1;
  Endfor
  For R in [3:4] do
    Phi([X,R]) : 0;
  Endfor
  For R in [5:8] do
    Phi([X,R]) : 1;
  Endfor
*/
  phi(x,[0:1]):1;
  phi(x,[2:6]):1;
  phi(x,[7:8]):1;

Endfor

For X in [46:50] do
  /*For R in [0:2] do
```

```
    Phi([X,R]) : 0;
  Endfor
  For R in [3:4] do
    Phi([X,R]) : 1;
  Endfor
  For R in [5:8] do
    Phi([X,R]) : 0;
  Endfor
*/
  phi(x,[0:1]):0;
  phi(x,[2:6]):0;
  phi(x,[7:8]):0;

Endfor

/*
  Phi([0:15])=0;
  Phi([16:41])=1;
  Phi([42:50])=0;
*/
    f1=1-eps; f2=f1/eps;
  For X in [0:Axial.EndNode]  do
    For R in [0:Radial.Endnode] do
 f3(X,R)=1-phi(X,R);
 Fc_cb(X,R)=f3(X,R)*f2;
 Fc_eb(X,R)=f1*f3(X,R);
 Fa_cb(X,R)=phi(X,R)*f2;
 Fa_eb(X,R)=phi(X,R)*f1;
 endfor
  endFor
//========================= Declaring Time-Invarient Physical Properties =========================//
  Had       as RealVariable (description:"Heat of Adsorption[kJ/kmol]", value:/*-30500*/-57950, spec:
Fixed);//
  Hrx       as RealVariable (description:"Heat of Reaction[kJ/kmol]", value:-108600, spec:Fixed);//
  Rho_s     as RealVariable(description:"solid phase density[kg/m³]", value:/*750*/2400, spec:fixed);
  Cp_S      as RealVariable (description:"Catalyst Mass Heat Capacity [kJ/kg/K]", value:1.050, spec:
Fixed);
  //Cond s    as RealVariable(description:"Catalyst Mass Heat Capacity [kW/m/K]", value:0.004,
spec:Fixed);// to be fixed by Rayleigh's Approximation

//========================= Aspen-Plus Physical Properties =========================//
Viscosity([0:Axial.EndNode],[0:Radial.EndNode])                as visc_vap (description:"Gas Phase
Viscosity [cP]");
Visc_Bar([0:Axial.EndNode],[0:Radial.EndNode])                 as Viscosity_BarS (description:"Gas
Phase Viscosity [Bar-s]");
D_mol(CompAll,CompAll)([0:Axial.EndNode],[0:Radial.EndNode])   as Diffus_vap (description:"Vapor
Phase Binary Molecular Diffusivity[cm²/s]");
D_mix([0:Axial.EndNode],[0:Radial.EndNode])                    as Binary Diffusivity(description:
"Vapor Phase Total Molecular Diffusivity[m²/s]");
D_ax ([0:Axial.EndNode],[0:Radial.EndNode])                    as Dispersion (description:"Axial
Dispersion[m²/s]");
Rho_g([0:Axial.EndNode],[0:Radial.EndNode])                    as dens_mass_vap(description:"Mass
Density Of the gas phase [kg/m³]");
D_mi(CompAll)([0:Axial.EndNode],[0:Radial.EndNode]) as hidden Diffusivity;
MW_vap([0:Axial.EndNode],[0:Radial.EndNode]) as  molweight(description:"Gas Phase Molecualr Weight
[kg/kmol]");
Mol_den([0:Axial.EndNode],[0:Radial.EndNode]) as dens_mol_vap (description:"[kmol/m³]");
Cp Vap Molar([0:Axial.EndNode],[0:Radial.EndNode])             as cp mol vap(description:"Vapor Heat
Capacity[kJ/kmol/K]");// has to be divided by Mw to get Cp Vap Mass
Cp_Vap_Mass([0:Axial.EndNode],[0:Radial.EndNode])              as RealVariable(description:"Vapor
Heat Capacity[kJ/kg/K]");
Cond_g_W([0:Axial.EndNode],[0:Radial.EndNode])                 as cond_vap (description:"Vapor Phase
Conductivity[W/m/K]");
Cond_g([0:Axial.EndNode],[0:Radial.EndNode])                   as RealVariable (description:"Vapor
Phase Conductivity[kW/m/K]");

//========================= Declaring Distributed Variables =========================//
Ci(CompAll)                    as Distribution2D (XDomain is Axial,YDomain is Radial,Description:
"Gas Phase Conc [kmol/m³]") of conc mole;
CFlux(CompAll)                 as Distribution2D (XDomain is Axial,YDomain is Radial,Description:
"Gas Phase Conc [kmol/m²/s]") of RealVariable;
Pt                             as Distribution2D (XDomain is Axial,YDomain is Radial,Description:
"Gas Phase Pressure [Bar]") of Pressure;
Yi(CompAll)([0:Axial.EndNode],[0:Radial.EndNode])  as molefraction (description:"Component Mole
Fractions");
Ct([0:Axial.EndNode],[0:Radial.EndNode])           as conc_mole;
Tk                             as Distribution2D (XDomain is Axial,YDomain is Radial,Description:
"Gas Phase Temperature") of Temperature abs;
Ts                             as Distribution2D (XDomain is Axial,YDomain is Radial,Description:
"Solid Phase Temperature") of Temperature_abs;
Tc ([0:Axial.EndNode],[0:Radial.EndNode])as Temperature;
ug([0:Axial.EndNode]) as velocity;


//========================= Sub-Model Decleration =========================//
```

```
Reac([0:Axial.EndNode],[0:Radial.EndNode]) as Reaction;
Ads([0:Axial.EndNode],[0:Radial.EndNode])  as Adsorption;
TP([0:Axial.EndNode],[0:Radial.EndNode])   as TransPortProperties;

 //======================== Sub-Model Connections ========================//
 For X in [0:Axial.Endnode] do
   For R in [0:Radial.Endnode] do
  Reac(X,R).Tk=Ts(X,R);  Reac(X,R).Pt=Pt(X,R); Reac(X,R).Yi(CompAll)=Yi(CompAll)(X,R); Reac(X,R).Phi=Phi(X,R);
  Ads(X,R).Tk=Ts(X,R);   Ads(X,R).Pt=Pt(X,R);  Ads(X,R).Yi(CompAll)=Yi(CompAll)(X,R);  Ads(X,R).D mi=D_mi("H2O")(X,R); Ads(X,R).Phi=Phi(X,R);
  TP(X,R).u=ug(X);  TP(X,R).D_mix=D_mix(X,R); TP(X,R).Visc_Bar=Visc_Bar(X,R); TP(X,R).rho_g=rho_g(X,R); TP(X,R).Cp_Vap_Mass=Cp_Vap_Mass(X,R);  TP(X,R).Cond_g=Cond_g(X,R); TP(X,R).Phi=Phi(X,R);
/*TP(X,R).Cond_s=Cond_s;*/  TP(X,R).Epsilon=eps;
  ENDFOR
ENDFOR

For X in [0:Axial.Endnode] do
  For R in [0:Radial.Endnode] do
  Call (Viscosity(X,R)) = pVisc Vap(Tc(X,R), Pt(X,R), Yi(CompAll,X,R)) ;
  Visc_Bar(X,R)  =Viscosity(X,R)*(1e-8);
  Call (D_mol(CompAll)(CompAll)(X,R)) = pDiffus_bin_Vap(Tc(X,R), Pt(X,R), Yi(CompAll,X,R));//[cm²/s]
  D_mix(X,R)=SIGMA(Yi(CompAll,X,R)*(D_mol(CompAll)(CompAll)(X,R)))/100000;// [m²/s]. D_mol is in [cm²/s]---> 1[m²/s]=10000[cm²/s]
  Call (Rho_g(X,R)) = pDens_Mass_Vap(Tc(X,R), Pt(X,R), Yi(CompAll,X,R));//[kg/m³]
  Call (D_mi(CompAll)(X,R)) = pDiffus_Vap(Tc(X,R), Pt(X,R), Yi(CompAll)(X,R));
  Call (MW_vap(X,R)) = pMolWeight(Yi(CompAll)(X,R));
  Call (Mol den(X,R)) = pDens Mol Vap(Tc(X,R), Pt(X,R), Yi(CompAll)(X,R));
  Call (Cp Vap Molar(X,R)) = pCp Mol Vap(Tc(X,R),Pt(X,R),Yi(CompAll)(X,R));//[kJ/kmol/K]
  Cp_Vap_Mass(X,R)=Cp_Vap_Molar(X,R)/Mw_vap(X,R);//[kJ/kg/K]
  Call (Cond_g_W(X,R)) = pCond_Vap(Tc(X,R),Pt(X,R),Yi(CompAll)(X,R));//[W/m/K]
  Cond_g(X,R)=Cond_g_W(X,R)/1000;//[kW/m/K]
ENDFOR
ENDFOR

//======================== Formulating Distributed variables ========================//

For X in [0:Axial.EndNode] do
    For R in [0:Radial.Endnode] do
      Tc(X,R)=Tk(X,R)-273.15;
  Ct(X,R)=SIGMA(Ci(CompAll)(X,R));
  CFlux(CompAll)(X,R)= Ci(CompAll)(X,R)*ug(X);
  Yi(CompAll)(X,R)=Ci(CompAll)(X,R)/(Ct(X,R));
EndFor
ENDFOR

 // VAM FIX
Acs=pi*(Diameter^2)/4;
Fin=ug(0)*Acs*Ct(0,0)*MW vap(0,0);
for X in [Axial.Interior+Axial.EndNode] do
ug(X) = Fin/sigma(foreach (R in [1:Radial.EndNode]) pi*(Radial(R)^2-Radial(R-1)^2) * (mol_den(X,R)*Mw_vap(X,R)+mol_den(X,R-1)*MW_Vap(X,R-1))/2);
endfor
 //======================== PDE'S ========================//
 // Component Balances:
 For X in [Axial.Interior] do
  For R in [Radial.Interior] do
  $Ci(CompAll)(X,R)=  -CFlux(CompAll)(X,R).ddX  +   TP(X,R).D ax*Ci(CompAll)(X,R).d2dX2//
                                                +   TP(X,R).D ar*(Ci(CompAll)(X,R).d2dY2 + (1/Radial.Value(R)) * Ci(CompAll)(X,R).ddY)//
                                                +   Fc_cb(X,R)*Reac(X,R).Rx(CompAll) + Fa_cb(X,R)*Ads(X,R).Rad(CompAll);//
 EndFor
 EndFor

   //Energy Balance

For X in [Axial.Interior] do
  For R in [Radial.Interior] do
  ((1-eps)*rho_s*cp_s)*$Ts(X,R)            =  + (0.005)*(Ts(X,R).d2dX2 + (Ts(X,R).d2dY2 + (1/Radial.Value(R)) * Ts(X,R).ddY))//
                                              - (1-eps)*TP(X,R).h_fs*(6/Dp)*(Ts(X,R)-Tk(X,R))//
                                              + (-Hrx*Fc_eb(X,R)*Reac(X,R).Rate + Had*Fa_eb(X,R)*Ads(X,R).Rad("H2O"));//

EndFor
EndFor

For X in [Axial.Interior] do
  For R in [Radial.Interior] do
  (eps*Rho_g(X,R)*Cp_Vap_Mass(X,R))*$Tk(X,R)    =  - eps*Rho_g(X,R)*Cp_Vap_Mass(X,R)*ug(X)*Tk(X,R).ddX//
                                                  + TP(X,R).L_ax*Tk(X,R).d2dX2//
                                                  + TP(X,R).L_ar*(Tk(X,R).d2dY2 + (1/Radial.Value(R)) * Tk(X,R).ddY)//
```

```
                                                + (1-eps)*TP(X,R).h_fs*(6/Dp)*(Ts(X,R)-Tk(X,R));//
EndFor
EndFor

 // Overall Balance(Velocity Calculation) and Darcy's Law:
For X in [Axial.Interior+Axial.EndNode] do
  Pt(X).ddX=-(180*Visc_Bar(X,0))*((fl^2)/((eps^3)*(Dp^2)))*ug(X);
EndFor


//========================= Fixed Initial Conditions =========================//
Ci(CompR)(Axial.Interior,Radial.Interior):1e-10,Initial;
Ci("N2")(Axial.Interior,Radial.Interior):0.023295947,Initial;
Tk(Axial.Interior,Radial.Interior):523.15,Initial;
Ts(Axial.Interior,Radial.Interior):523.15,Initial;

//========================= Boundary Conditions =========================//

/*
Ci("H2S")(0,[0]+Radial.Interior+Radial.EndNode)=0.002329595;
Ci("O2S")(0,[0]+Radial.Interior+Radial.EndNode)=0.001164797;
Ci("S8")(0,[0]+Radial.Interior+Radial.EndNode)=0;
Ci("H2O")(0,[0]+Radial.Interior+Radial.EndNode)=0;
Ci("N2")(0,[0]+Radial.Interior+Radial.EndNode)=0.019801555;

Pt(0)=Ct(0,0)*(0.0814)*Tk(0,0);
ug(0)=0.2*1;

Tk(0,[0]+Radial.Interior+Radial.EndNode)=523.15;
Ts(0,[0]+Radial.Interior+Radial.EndNode)=523.15;

//@Z=L
Ci(CompAll)(Axial.EndNode,[0]+Radial.Interior+Radial.EndNode).ddX=0;
Tk(Axial.EndNode,[0]+Radial.Interior+Radial.EndNode).ddX=0;
Ts(Axial.EndNode,[0]+Radial.Interior+Radial.EndNode).ddX=0;

//@r=0
Ci(CompAll)(Axial.Interior,0).ddy=0;
Tk(Axial.Interior,0).ddy=0;
Ts(Axial.Interior,0).ddy=0;

//@r=R
Ci(CompAll)(Axial.Interior,Radial.EndNode).ddy=0;
Tk(Axial.Interior,Radial.EndNode).ddy=0;
Ts(Axial.Interior,Radial.EndNode).ddy=0;
*/

// @Z=0

Ci("H2S")(0,[0]+Radial.Interior+Radial.EndNode)=0.002329595;
Ci("O2S")(0,[0]+Radial.Interior+Radial.EndNode)=0.001164797;
Ci("S8")(0,[0]+Radial.Interior+Radial.EndNode)=0;
Ci("H2O")(0,[0]+Radial.Interior+Radial.EndNode)=0;
Ci("N2")(0,[0]+Radial.Interior+Radial.EndNode)=0.019801555;

Pt(0)=Ct(0,0)*(0.0814)*Tk(0,0);
ug(0)=0.2*1;

Tk(0,Radial.Interior+Radial.EndNode)=523.15;
Ts(0,Radial.Interior)=523.15;

//@Z=L
Ci(CompAll)(Axial.EndNode,Radial.Interior).ddX=0;
Tk(Axial.EndNode,Radial.Interior).ddX=0;
Ts(Axial.EndNode,Radial.Interior).ddX=0;

//@r=0
Ci(CompAll)(Axial.Interior+Axial.EndNode,0).ddy=0;
Tk([0]+Axial.Interior+Axial.EndNode,0).ddy=0;
Ts([0]+Axial.Interior+Axial.EndNode,0).ddy=0;

//@r=R
Ci(CompAll)(Axial.Interior+Axial.EndNode,Radial.EndNode).ddy=0;
//Tk(Axial.Interior+Axial.Endnode,Radial.EndNode).ddy=0;
//Ts([0]+Axial.Interior+Axial.EndNode,Radial.EndNode).ddy=0;

-TP([1:5]/*Axial.Interior+Axial.Endnode*/,Radial.EndNode).L_ar*Tk([1:5]
/*Axial.Interior+Axial.Endnode*/,Radial.EndNode).ddy=0.2394*(Tk([1:5]/*Axial.Interior+Axial.Endnode*/
,Radial.EndNode)-513.15);
-0.005*Ts([0:5]/*[0]+Axial.Interior+Axial.EndNode*/,Radial.EndNode).ddy=0.1779*(Ts([0:5]
/*[0]+Axial.Interior+Axial.EndNode*/,Radial.EndNode)-513.15);

-TP([6:45]/*Axial.Interior+Axial.Endnode*/,Radial.EndNode).L_ar*Tk([6:45]
/*Axial.Interior+Axial.Endnode*/,Radial.EndNode).ddy=0.2394*(Tk([6:45]
/*Axial.Interior+Axial.Endnode*/,Radial.EndNode)-473.15);
```

```
-0.005*Ts([6:45]/*[0]+Axial.Interior+Axial.EndNode*/,Radial.EndNode).ddy=0.1779*(Ts([6:45]
/*[0]+Axial.Interior+Axial.EndNode*/,Radial.EndNode)-473.15);

-TP([46:50]/*Axial.Interior+Axial.Endnode*/,Radial.EndNode).L_ar*Tk([46:50]
/*Axial.Interior+Axial.Endnode*/,Radial.EndNode).ddy=0.2394*(Tk([46:50]
/*Axial.Interior+Axial.Endnode*/,Radial.EndNode)-533.15);
-0.005*Ts([46:50]/*[0]+Axial.Interior+Axial.EndNode*/,Radial.EndNode).ddy=0.1779*(Ts([46:50]
/*[0]+Axial.Interior+Axial.EndNode*/,Radial.EndNode)-533.15);

//======================== Performance Indices ========================//
//Variables to plot
BreakH2o as RealVariable;
  BreakH2o=Yi("H2O")(Axial.EndNode,0);
  XX as realvariable;
  XD([0:Axial.EndNode],[0:Radial.EndNode]) as Realvariable;
    equilibrium([0:Axial.EndNode],[0:Radial.Endnode]) as Realvariable;
  XX=1-((sigma(Yi("H2S")(Axial.EndNode,[0]+Radial.Interior+Radial.Endnode)/(NoR+1))))/(sigma(Yi("H2S"
)(0,[0]+Radial.Interior+Radial.Endnode)/(NoR+1)));
For X in [0:Axial.Endnode] do
  For R in [0:Radial.EndNode] do
  XD(X,R)=1-((Yi("H2S")(X,R)))/(Yi("H2S")(0,0));
    equilibrium(X,R)=((Yi("H2O")(X,R))^2)*((Yi("S8")(X,R))^(3/8))/(((Yi("H2S")(X,R))^2)*(Yi("O2S")(X,
R)));
EndFor
EndFor
//-------------------------------------------------------------------------------------------------------
//==================================END SIMULATION=======================================================//
//-------------------------------------------------------------------------------------------------------

End
```

```
Model Reaction

//Connection Variables with ADR2D Model
  CompAll               as StringSet(["H2S","O2S","S8","H2O","N2"]);
  Pt                    as pressure(description:"Reactor Pressure[bar]");
  Tk                    as temperature_abs ;
  Yi(CompAll)           as molefraction (description:"Component Mole Fractions");
// Reaction Specific Variables
  Ea1         as hidden Activation Energy (description:"Forward Reaction[J/mol]", value:49980, spec:Fixed);
  Ea2         as hidden Activation_Energy (description:"Backward Reaction[J/mol]", value:86601, spec:Fixed);
  rho_cat     as hidden dens_mass         (description:"Catalyst Density[kg/m³]", value:/*734*/3600, spec:Fixed);
  k1          as RealVariable;
  k2          as RealVariable;
  Phi         as realvariable;
  //Approach to eq  as RealVariable;


// Reaction Rate Computation
  Rx(CompAll)     as  Reaction_Rate(Description:"Rate [kmol/m³/s]");
  Rate            as  Reaction_Rate(Description:"Rate [kmol/m³/s]");

// Claus Reaction Rate
k1=(/*5.292e-3*/17.12458)*EXP((-Ea1)/(8.314*Tk));
k2=(/*1.252*/1168.434385)*EXP((-Ea2)/(8.314*Tk));//(1e-3)*
  //IF ((Yi("H2S")>=0) and (Yi("O2S")>=0) and (Yi("H2O")>=0)) THEN//

//Rate=(k1*((Yi("H2S")*Pt*1000)^0.95)*((Yi("O2S")*Pt*1000)^0.22)-k2*(Yi("H2O")*Pt*1000))*rho_cat;//[mol/(kg-catalyst)/s]*[kg-catalyst*m³]*[m³/mol]=[kmol/m³/s]
Rate=(k1*((Yi("H2S")*Pt/**1000*/)^0.95)*((Yi("O2S")*Pt/**1000*/)^0.22)-k2*(Yi("H2O")*Pt/**1000*/))*rho_cat;//[mol/(kg-catalyst)/s]*[kg-catalyst*m³]*[m³/mol]=[kmol/m³/s]
  //ELSE
    //Rate=0;
  //ENDIF

     //(k2*(Yi("H2O")*Pt)) * Approach to eq = (k1*((Yi("H2S")*Pt)^0.95)*((Yi("O2S")*Pt)^0.22));


// Component Specific Reaction Rates: 2H2S + SO2 + N2 <==> (3/8)S8 + 2H2O + N2
  Rx("H2S") =   -2*Rate;// Consumption
  Rx("O2S") =     -Rate;// Consumption
  Rx("H2O") =    2*Rate;// Generation
  Rx("S8")  =(3/8)*Rate;// Generation
  Rx("N2")  =0;

End
```

```
Model Adsorption



//Connection Variables with ADR2D Model
  CompAll   as StringSet(["HCl","O2","Cl2","H2O","N2"]);
  Yi(CompAll)   as molefraction;
  Pt            as pressure       (description:"Reactor Pressure[bar]");
  Tk            as temperature abs(description:"Temperature[K]") ;
  Q             as conc mole      (description:"Adsorbate Concentartion in the bed[kmol/m³]");
  Pw            as Pressure       (description:"Partial Pressure Of Water[Bar]");
  Phi           as realvariable;


// Adsorption Specific Parameters
  rho_ad    as hidden dens_mass           (description:"Catalyst Density[kg/m³]", value:1199.3/*750*/
, spec:Fixed);
  Binf      as RealVariable               (description:"Maximum Adsorption Affinity [K^0.5/bar]",
value:5.3126E-5, spec:Fixed,scale:1e-8);//--> 1[Pa]=0.01[mbar]
  T0        as  hidden  temperature abs   (description:"Adsorption Reference Temperature [K]", value:
300, spec:Fixed);
  q0w       as conc_mole                  (description:"Loading Correction [kmol/m³-Adsorbent]",
value:/*8.05845*/12.886, spec:Fixed);
  gamma     as RealVariable               (description:"Adsorption Parameter [-]", value:23.235, spec
:Fixed);
  delta     as hidden  RealVariable       (description:"Adsorption Parameter [-]", value:0.68792,
spec:Fixed);
  Dp        as hidden  length             (description:"Average Particle Diameter [m]", value:
/*1.785E-3*/3.6E-3, spec:Fixed);
  Rg        as hidden  RealParameter      (description:"Regnault's Constant [J/K/mol]",Value:
8.3144621);

// Adsorption Specific Variables
  Kldf      as RealVariable   (description:"Linear Driving Force Coefficient[1/s]");// has to be
changed later on
  qsw       as conc_mole      (description:"Saturation Loading Capacity [kmol/m³-Adsorbent]");
  B         as RealVariable   (description:"Adsorption Affinity[1/bar]",scale:0.1);
  Qstar     as conc mole      (description:"Equillibrium Loading  [kmol/m³-Adsorbent]");
  qstr      as conc mole      (description:"Equillibrium Loading  [kmol/m³-Adsorbent]");
  r_pore as realvariable(3e-8,Fixed);//cm
  Mi as realvariable(18,fixed);
  D_k as realvariable;
  D_mi as RealVariable;
  Tau as RealVariable(1,Fixed);
  D_ad as realvariable;

// Adsorption Rate Computation ===> Equivalent To Reaction Rate
  Rad(CompAll)   as  Reaction Rate(Description:"Rate [kmol/m³/s]");
  RateAd         as  Reaction Rate(Description:"Rate [kmol/m³/s]");
  A              as RealVariable;
  F_dash         as RealVariable;

//Declaring Solvent Pore Diffusion Coefficient Variables for Zeolite 3A
D_k=9700*r_pore*sqrt(Tk/Mi);
D_ad*10000=(D_k*D_mi)/(Tau*(D_k+D_mi));

// Adsorption Rate
  Pw=Yi("H2O")*Pt;
  EqAD1:  qsw=q0w*EXP(delta*(1-(Tk/T0))); //[kmol/m³-Adsorbent] -> No change
  EqAD2:  B=(Binf/SQRT(Tk))*EXP(gamma*T0/Tk);// 1/bar  -> No change
  EqAD3:  qstr=B*qsw*Pw/(1+ B*Pw);//[mol/kg-Adsorbent]  -> No change

  EQAD4:  A=B*Pw;//Unitless
  EQAD5:  F_dash =qsw*B*Tk*(8.314e-5)/((1+A)^2);
  EqAD6:  Kldf=(60/(Dp^2)) * D_ad  * (1/(1+(1/0.5)*F_dash ));//[1/s]
 // EqAD7:  $Q =(Kldf*(qstr-Q));//[kmol/(kg-adsorbent)/s]*[kg-adsorbent/m³]=[kmol/m³/s]

 IF (0.00001/*0.9*/ < Phi ) then
  EqAD7:  $Q =(Kldf*(qstr-Q));//[kmol/(kg-adsorbent)/s]*[kg-adsorbent/m³]=[kmol/m³/s]
  Else
  $Q=0;
  ENDIF

  EqAD8:  RateAd=$Q;


  Rad("H2O")  =   -RateAd;// Consumption
  Rad("HCl")  =   0;// Consumption
  Rad("O2")  =   0;// Consumption
  Rad("Cl2")   =   0;// Generation
  Rad("N2")   =   0;

  Q:1e-10,Initial;
```

```
Model TransportProperties


//Conection Parameters With The Model ADR2D

u          as velocity(description:"True Velocity");
D_mix      as Binary_Diffusivity(description:"Gas Phase Diffusivity");
Rho g      as dens mass vap;
Visc Bar   as Viscosity BarS;
Visc Pas   as Realvariable(description:"Gas Phase Viscosity[kg/m/s]");
Cpg        as RealVariable(description:"Mass Heat Capacity of Gas Phase[J/kg/K]");
Cp_Vap_Mass       as RealVariable(description:"Mass Heat Capacity of Gas Phase[kJ/kg/K]");
Epsilon    as Voidage_Fraction;
Cond_Cat   as RealVariable(description:"Catalyst Conductivity[kW/m/K]",0.00574,Fixed);
Cond_Ads   as RealVariable(description:"Adsorbent Conductivity[kW/m/K]",0.00012,Fixed);
Cond_s     as RealVariable;
Cond_g     as RealVariable;
Phi as Voidage Fraction;

//Declaring Parameters
Dp         as length       (description:"Average Particle Diameter [m]", value:/*4.44E-3*/3.6E-3, spec
:Fixed,scale:1e-3);
D_Reactor as RealVariable (description:"Reactor Diameter [m]", value:0.06, spec:Fixed,scale:1e-2);

N          as RealVariable;
K          as realVariable;
M          as RealVariable(Description:"Der Parameter");
a1         as Realvariable;
a2         as Realvariable;
B          as RealVariable;

Pe_mz      as RealVariable(Description:"Axial Mass Peclet Number");
Pe_hz      as RealVariable(Description:"Axial Heat Peclet Number");
Re_p       as RealVariable(Description:"Particle Reynolds Number");
Sc         as RealVariable(Description:"Schmidt Number");
Pr         as RealVariable(Description:"Prandtl Number");
Nu         as Realvariable;

D_ax       as Dispersion;
L_ax       as Dispersion;
D_ar       as Realvariable;
L_ar       as Realvariable;
h_fs       as Realvariable;


//Calculating Solid Phase Conductivity - Rayleigh's Law
if(phi==0) then
  Cond_s=Cond_cat;
  elseif(phi==1) then
    Cond_s=Cond_Ads;
    else
Cond_s/Cond_Cat=1+(3*Phi)/(((Cond_Ads+2*Cond_Cat)/(Cond_Ads-Cond_Cat))-(Phi)+1.569*((Cond_Ads-
Cond_Cat)/(3*Cond_Ads-4*Cond_Cat))*(Phi^(10/3)));
endif


Visc_Pas=Visc_Bar*1e5;

//Calculating DimensionLess Numbers
Pr      = Visc_Pas*Cp_Vap_Mass/(Cond_g);
Re_p    = rho_g*Dp*u/Visc_Pas;
Pe_hz   = Re_p*Pr;
Nu      = (2+1.1*(Pr^(1/3))*(Re_p^(0.6)));

// Axial Mass Dispersion Coefficient - D_ax
D_ax    = 20*D_mix + 0.5*u*Dp;

// Radial Mass Dispersion Coefficient - D ar
M       = 57.85-35.36*Log10(Re p)+6.68*(Log10(Re p))^2;
D_ar    = (1/m + 0.38/Re_p)*u*Dp;

// Axial Heat Dispersion Coefficient - L_ax
L_ax *1000    = ((0.73/(Re_p*Pr))+(0.5/(1+(9.7/(Re_p*Pr)))))*(u*Rho_g*Cp_Vap_Mass*1000*Dp);// because
this has to be kept in [Kw/m/K]

// Radial Heat Dispersion Coefficient - Lar
N       = D_reactor/Dp;
a2      = 8*(2-(1-2/N)^2);
B       = 1.25*((1-epsilon)/epsilon)^(1.11);
K       = Cond_s/Cond_g;
a1      = (1-SQRT(1-epsilon))+ 2*(SQRT(1-epsilon))/(1-B*(K^-1))*((B*(1-K^-1)/(B*(1-K^-1))^2)*LOGE(K/B
) - (B-1)/(B*(1-K^-1))^2 - (B+1/2));
L_ar    = (a1*0+Re_p*Pr/a2)*Cond_g;

// Fluid To Solid Heat Transfer Coefficient - H_fs



h_fs=(2+1.1*(Pr^(1/3))*(Re_p^(0.6)))*Cond_g/Dp;
```