POLITECNICO DI MILANO

Facoltà di Ingegneria Industriale e dell'Informazione

Master's Degree in Chemical Engineering

AG2S™ TECHNOLOGY OPTIMIZATION: APPLICATION OF A GENERALIZED FRAMEWORK INCLUDING ROBUST OPTIMIZER INTO ASPEN HYSYS

Supervisor
Prof. Flavio Manenti

Tutor
Ing. Andrea Bassani

Candidate
Riccardo Fossati
Matr. 854474

Academic Year 2016-2017

Some things work out best when you don't try so hard.

RINGRAZIAMENTI

Al termine non solamente del lavoro di tesi ma di cinque anni di percorso universitario è doveroso e naturale dover esprimere alcuni sentiti ringraziamenti, rivolti a tutti quanti hanno reso possibile il raggiungimento del traguardo finale.

In primis grazie al prof. Manenti per avermi concesso l'opportunità di svolgere questo lavoro e di impegnarmi in un campo che mi ha sempre stimolato, in grado di coniugare alcuni ambiti tradizionali dell'ingegneria chimica con la progressiva spinta all'automazione e digitalizzazione che sta coinvolgendo non solo il mondo accademico ma anche quello industriale.

All'entusiasmo iniziale va sempre fatta seguire costanza e dedizione per non cedere di fronte alle piccole e grandi difficoltà: per questo grazie a tutti i tesisti e dottorandi del SuPER Team che hanno condiviso con me anche solo una parte di cammino e in particolare al mio tutor "adottivo" Andrea Bassani, che ha avuto risposte alle mie domande e fornito preziose indicazioni sulla base delle quali sono riuscito a impostare questo lavoro.

Grazie di cuore a tutte le persone che ho la fortuna di chiamare amici, sia quelli incontrati in università sia quelli conosciuti da una vita; nessuna frase di circostanza riuscirà a esprimere quanto siate importanti per me e quanto la vostra presenza (per fortuna tutt'altro che silenziosa) sia stata stimolo e motivazione per fare e crescere sempre in meglio.

Ad ultimo (ma, come al solito, sicuramente non in ordine di importanza) grazie alla mia famiglia: cugini, zii, nonni ma soprattutto mamma, papà e Dany. Se oggi festeggio un traguardo lo dedico ai vostri sacrifici e all'amore che siete stati capaci di trasmettermi, e non dimentico che domani rimane da scrivere e che gli insegnamenti che mi avete dato valgono per tutta la vita.

Spero che queste poche righe, per necessità troppo sintetiche, non facciano sentire nessuno escluso o dimenticato; vi porto tutti nel cuore e sono fermamente convinto che nessun incrocio, nessun incontro è stato per caso o poco significativo e ha anzi fatto la sua parte nel portarmi fino a qui.

Grazie,

Riccardo

INDEX

FIGURE INDEX

TABLE INDEX

ABSTRACT

H_2S and CO_2 are two of the most critical by-product generated from the global energy production via chemical route. First of all, despite of their high volume production, they do not represent a mayor feedstock or a commodity chemical useful for some successful industrial aim. Therefore, they are considered as a waste and when it is possible they are discharged in atmosphere and/or treated following the law limits dictated from different nations, causing therefore relevant environmental problems.

H_2S comes in large part from the deSulfurizing of hydrocarbons process and the CO_2 is produced in huge quantities from energetic industries, heavy industries, chemical, petrochemical, and combustion processes that release the CO_2 in the atmosphere. The capture and storage of the CO_2 is object of relevant discussion and technological improvements necessary to make the whole process economically and technologically sustainable.

The hydrogen sulphide is actually send to neutralization plants where with the help of air oxidation and, a consecutive reduction of H_2S remained with SO_2 produced in the previous step, it is stabilized and transformed as elemental Sulfur. This solid product is reused in the modern process industry, for example in the Sulfuric acid (H_2SO_4) production. Today, the Sulfuric acid production decreased due to the market saturation, and therefore these neutralization plants (Claus plants) highlighted a seriously decrease and a consequent crisis.

The CO_2 is totally discharged in atmosphere and it is the cause of the feared “greenhouse effect”, which would cause as many scientist said an increment of the temperature of the earth crust, with strong damages to the entire animal, human, plant lives present on the world. These facts has led many countries around the world to some internationals agreements necessary to the reduction of greenhouse effect gases reduction (cfr. Paris Agreements for the Climate Change). Actually, in the world 10 billion of ton per year of CO_2 are discharged in atmosphere.

Due to his thermodynamics stability and with is low chemical value, CO_2 has few industrial uses; one of the most important use is the industrial production of urea but, industrial applications as the dry reforming and catalytic hydrogenation were considered not reliable industrial applications.

In the next future, the environmental concern of the industrialized countries will be present and the society need to find a solution of such problems that regards the whole human population. To this question and with this objective, a new chemical and industrial application that gives a potential value to acid gases will be studied and optimized in this work.

This thesis work is therefore aimed towards the development of a methodology able to find optimal working conditions for the Regenerative Thermal Reactor, which is the core of the AG2S™ technology, the novel process configuration developed at Politecnico di Milano in order to shift selectivity of the Claus process from the production of elementary Sulphur to the more valuable syngas.

Literature search provided the numerical basis upon which the work has been conducted; the scarcity of available data in the field of acid gas reaction towards syngas reduced the investigation to three industrial cases. Those data were validated through the industrial software STRESS and the OpenSMOKE++ simulation environments, suitable for reacting system with complex kinetics.

A model representing the potential reactor configuration for the novel process in exam was developed by Fabio Cecchetto in his M.Sc. thesis. The regenerative thermal reactor (RTR) solution, in place of the classic Claus furnace, will be discussed and tested to evaluate the syngas yield.

In order to optimize the working conditions, a cross-platform architecture was built, with the aim to link Hysys, Matlab and OpenSMOKE++ to C++, the language in which the robust optimization algorithm available from the BzzMath library was written. In this way the potential of Aspen Hysys was extended to the complete study of processes based on complex kinetics thanks to a very flexible and easy-to-handle tool, a feature that could be applied to a wide range of different situations.

In the fourth and last part the real optimization of industrial units takes place, starting from the definition of suitable independent variables and objective functions. The results show the best operating conditions on a process and economic level.

Given the great availability of ASCII files, which represent the working interface of the C++ language, the tool herein described may represent a useful way to implement specific libraries (devoted, as the BzzMath one, to the coverage of many fields of the numerical analysis) into commercial packages like Aspen Hysys to solve problems which are typical of Chemical Engineering.

SINTESI

H_2S e CO_2 sono due dei più critici sottoprodotti generati dalla produzione di energia per via chimica. Innanzitutto, a dispetto della loro produzione in enormi quantità, essi non rappresentano una principale feedstock o una commodity chemical per successivi scopi industriali. Pertanto sono considerati scarti e quando possibile vengono scaricati in atmosfera e/o trattati a seconda dei limiti di legge dei vari paesi, causando indubbiamente rilevanti problemi ambientali.

H_2S deriva in gran parte dalla desolforazione dei combustibili fossili mentre la CO_2 è prodotta in grande quantità da industrie energetiche, manifatturiere, chimiche, petrolifere, petrolchimiche e dai processi di combustione usuali che la emettono in atmosfera. La cattura e il sequestro della CO_2 è ancora oggetto di rilevanti dibattiti e di tecnologie all'avanguardia per rendere il tutto economicamente e tecnologicamente sostenibile.

L'acido solfidrico (H_2S) attualmente viene convogliato a degli impianti di neutralizzazione che grazie alla ossidazione in aria e alla successiva riduzione dell'H_2S residuo con l'SO_2 prodotta nello step precedente viene stabilizzato a zolfo elementare. Tale prodotto solido è in parte riutilizzato nell'industria di processo, ad esempio nella produzione di acido solforico (H_2SO_4). Oggigiorno, la produzione di acido solforico, ha subito un forte calo data la saturazione di mercato e, pertanto tali impianti di neutralizzazione dei gas acidi hanno subito un forte ridimensionamento e conseguente crisi trovandosi a commercializzare un prodotto in perdita.

La CO_2 è quasi interamente scaricata in atmosfera, ed è causa ad oggi del tanto temuto "effetto serra" che causerebbe secondo molti scienziati un aumento della temperatura della crosta terrestre con conseguenti danni a livello di vita terrestre, tanto che ciò ha portato diversi paesi del mondo industrializzato ad accordi internazionali volti alla riduzione delle emissioni di gas serra (vedi ultimi accordi di Parigi). Nel complesso si stima che nel mondo vi siano emissioni tali da raggiungere i 10 miliardi di tonnellate annue di CO_2 emessa in atmosfera.

Data la sua enorme stabilità termodinamica e con il suo basso valore chimico, la CO_2 ha pochi utilizzi industriali; uno dei più importanti è quello legato alla produzione industriale di urea mentre, altre applicazioni come il dry reforming e la idrogenazione catalitica risultano ancora opzioni scartate a livello di impiego industriale.

Nel prossimo futuro, data come certa la coscienza ambientale dei paesi industrializzati è necessario trovare una soluzione a tali problemi che riguardano l'intera la popolazione mondiale. A questo interrogativo si pone come obiettivo il raggiungimento di una possibile via chimica, applicabile a livello industriale, che dia un potenziale valore ai gas acidi presenti nell'attuale e prossimo sviluppo industriale.

In questo studio ci si pone l'obiettivo di sviluppare una metodologia volta alla ricerca delle migliori condizioni operative possibili per il Reattore Termico Rigenerativo (RTR), il quale rappresenta un punto cardine della tecnologia AG2S™, una nuova configurazione di processo recentemente

sviluppata presso il Politecnico di Milano al fine di modificare la selettività del processo Claus verso la produzione di syngas tramite promozione una reazione redox tra i gas acidi.

Tre casi industriali sono dunque stati selezionati e presi in esame, ovvero sottoposti a validazione attraverso il software STRESS prima e l'ambiente di calcolo OpenSMOKE++ poi, particolarmente adatti per sistemi reagenti dalla cinetica complessa.

La configurazione del nuovo processo prevede dunque un reattore termico rigenerativo (RTR) in luogo della classica fornace Claus; un possibile modello è stato sviluppato da Fabio Cecchetto nel corso della tesi di Laurea Magistrale. Esso verrà studiato e testato con i dati disponibili per provare l'efficacia della soluzione in uso in termini di resa in syngas e conversione di gas acidi nella seconda parte di questo lavoro.

Nella terza parte verrà sviluppata l'architettura del sistema di ottimizzazione, pensata in maniera tale da coinvolgere tre differenti ambienti di calcolo: Aspen Hysys, Matlab e C++. Proprio in quest'ultimo trova applicazione l'algoritmo robusto di ottimizzazione disponibile presso la libreria BzzMath che rappresenta il cuore dell'intera struttura. L'algoritmo è stato dunque sviluppato utilizzando Matlab Compiler SDK, in maniera tale da trasformare la funzione in una shared library interpretabile da Visual Studio come avente al suo interno detta funzione in esame. Il metodo è del tutto generale e la sua grande flessibilità consente di estendere le potenzialità di Aspen Hysys, in maniera da giungere fino allo studio completo di processi basati su reazioni dalla cinetica complessa, prima difficilmente coniugabili con la struttura ben definita del pacchetto commerciale.

Nella quarta e ultima parte trova luogo l'ottimizzazione vera e propria del processo, passante per la definizione di opportune variabili indipendenti e funzioni obiettivo. Sono dunque riportati i risultati derivanti da questo tipo di analisi a livello di processo ed economico.

Un completo riciclo di H_2S nell'impianto di riconversione e una rilevante riduzione delle emissioni di CO_2, oltre alla produzione di syngas da questi agenti inquinanti, rappresenta una strada praticabile dal punto di vista ambientale, commerciale e delle sostenibilità energetica, focalizzando sempre più l'attenzione sulla possibile risorsa che i gas acidi possono offrire.

Data dunque la grande disponibilità dei file ASCII, che rappresentano l'interfaccia operativa del linguaggio C++, lo strumento qui descritto consente di implementare librerie già disponibili (che, come BzzMath, coprano campi dell'analisi matematica) in pacchetti commerciali molto diffusi anche possedendo conoscenze solo basilari in ambito di programmazione, agevolando la risoluzione di problemi tipici dell'ingegneria chimica.

INTRODUCTION

1. State of the Art

The removal of Sulfur components from liquid and gas streams is required in many sectors of the hydrocarbon processing industry. With more stringent fuel regulations and increasing environmental concerns, together with the need to process sourer crude oils and natural gases, Sulfur recovery has become one of the leading issues in emission reduction. The term Sulfur cycle (Fig. 1) designates a large number of processes widely used in the refining industry, for purposes ranging from the transformation and/or capture of Sulfur compounds contained in the petroleum fractions to their removal, generally as elemental Sulfur. The general idea of the Sulfur cycle is to eliminate Sulfur compounds from various petroleum fractions (Heinrich and Kasztelan, 2001).

This is realized via: a) isolation and concentration of the undesired Sulfur compounds; b) Sulfur species transformation mainly into hydrogen sulphide (H_2S) in HydroDeSulphurization (HDS) units) and in hydrocracking or catalytic cracking units; c) capture and enrichment of H_2S via solvent washing (e.g. amines units); d) conversion of H_2S into elemental Sulfur in the Sulphur Recovery Unit (SRU): Claus and/or other processes. Elemental Sulfur is the ultimate state of recovery of the Sulfur species.

In the past, recovered elemental Sulfur had considerable value and was sold in the commercial marketplace. However, as the hydrocarbon extraction industry continually recovers more Sulfur, the supply far exceeds demand and prices are driven down to levels where transportation is no longer economically possible. This market is now expected to exhibit a chronic oversupply. On a worldwide basis, approximately 60 million metric tons of Sulfur were produced in the year 2000. It is generally assumed that the Claus process had produced 85% of this Sulfur, 90% of which being used for Sulfuric acid (H_2SO_4) production (60% of H_2SO_4 is used for fertilizer production). The discussions below intend to give a general idea of the capabilities of various Sulfur recovery processes, while taking into account the nature of the stream. Compared to gas processing, petroleum refining is a source of gas with low carbon dioxide (CO_2) content; nevertheless, if there is a catalytic cracking unit in the refinery scheme, the gas may contain some other contaminants such as carbonyl sulphide (COS), organic Sulfur, cyanides, ammonia and organic acids.

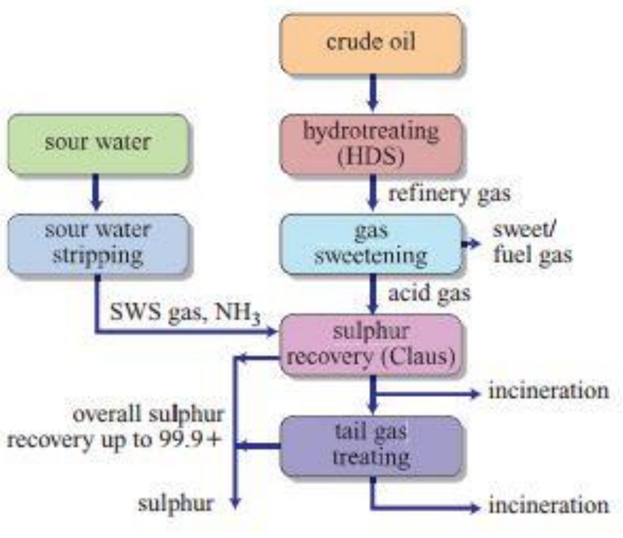


Figure 1: Typical Sulfur cycle in the refining industry

1.1. The Claus Process

The objective of the Claus process is to recover elemental Sulfur (S_x , with x between 2-8 depending on the temperature) from gas streams containing hydrogen sulphide (H_2S) stripped from gas sweetening solvents. The Claus process produces elemental Sulfur by the partial oxidation of H_2S:
$H_2S + \frac{1}{2} O_2 + \frac{1}{8} S_8 + H_2O$ ΔH= -209 kJ (1)

The Claus plant effluent gases are either sent to an incinerator or to a TGT unit, depending on the local air pollution control regulations. The final effluent gas, which cannot be valorized, is incinerated to convert all of the Sulfur compounds to Sulfur dioxide (SO_2) in a thermal or catalytic incinerator. Being of very good quality, the elemental Sulfur produced in the SRU, Claus or Claus plus TGT, is used as a basic chemical in the industry. The properties of elemental Sulfur are well described in various literature (Meyer, 1976; Shuai and Meisen, 1995). In the original Claus process, reaction (1) was carried out in a single step over a catalyst. Since the heat of the reaction was dissipated only by radiation, high Sulfur recovery was very difficult to obtain.

A very important modification to the Claus process was made in 1940, which allowed energy recovery, increased process capacity and eliminated the issue of maintaining, in the catalytic reactor, the low temperature favouring high Sulfur recoveries. In this modified-Claus process (Fig. 2), reaction (1) is carried out in two stages. In the first stage, the thermal section, one third of the H_2S is oxidized to SO_2 with air or oxygen enriched air at high temperature (generally 925-1,200°C):
$H_2S + \frac{3}{2} O_2 \rightarrow SO_2 + H_2O$ ΔH= -518 kJ (2)

This reaction is highly exothermic and is not limited by equilibrium. The unburned H_2S in the acid gas reacts with the SO_2 (obtained through reaction (2), to yield the stoichiometric H_2S/SO_2 ratio of 2:1) to form elemental Sulfur vapour: 2
$H_2S + SO_2 \rightarrow \frac{3}{2} S_2 + 2 H_2O$ (3)

This reaction is endothermic and is limited by equilibrium. About 60-70% of the conversion of H_2S to elemental Sulfur occurs in the thermal stage. An important function of the thermal section is also to destroy the impurities that may be present in the feed acid gas stream, such as ammonia (NH_3), hydrocarbons, etc. During the thermal stage, side reactions also occur in the presence of CO_2 or hydrocarbons, which produce COS and CS_2.

In the second stage (i.e. catalytic section), the overall conversion of H_2S to elemental Sulfur is increased in a series of catalytic reactors (1 to 3) by reaction of the generated SO_2 and the unreacted H_2S H_2S over fixed beds of Claus catalysts at much lower temperatures (190-360°C): $2 H_2S + SO_2 \rightarrow \frac{3}{8} S_8 + 2 H_2O$ ΔH= -108kJ (4)

Reaction (4) is called the Claus reaction. The use of appropriate catalysts at selected temperatures optimizes the Claus reaction yield and also allows COS and CS_2 produced in the thermal stage to be eliminated. High-Pressure (HP) steam is generated in the Waste Heat Boiler (WHB), in which the gases are cooled from the high flame temperature to the lower catalytic reactor (converter)

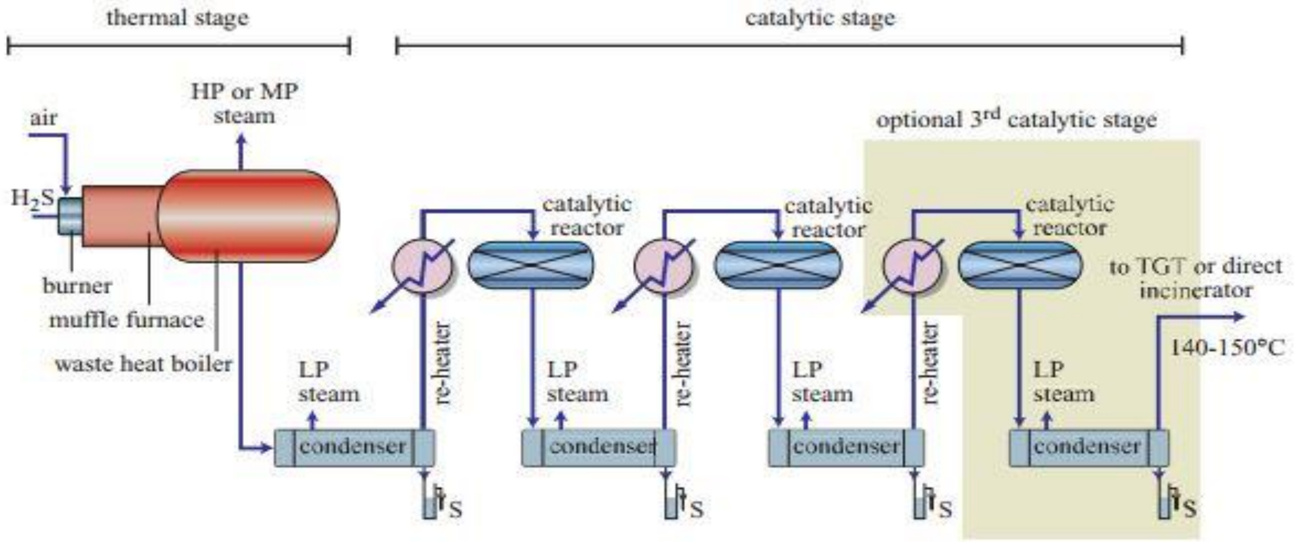


Figure 2: Typical Claus process scheme (straight-through design)

temperature (see again Fig. 2). Sulphur condensers are provided to condense and separate the elemental Sulfur formed after the thermal stage and after each catalytic reactor. The heat released by the Claus reaction is recovered as Low-Pressure (LP) steam in each Sulfur condenser. Product removal, lower catalytic converter temperatures and an increased number of catalytic converters enhance Sulfur recovery. Table 1 presents the typical, total Sulfur recovery of the modified-Claus process depending on the number of catalytic reactors used.

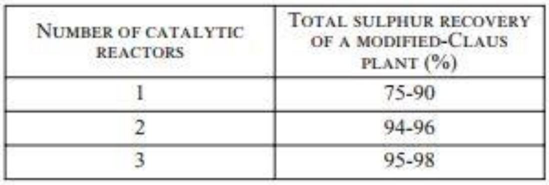
Nowadays, Sulfur recovery plants are based on the modified-Claus process, although the original Claus process is still implemented to treat very low H_2S concentrations gases, though in this case, it is referred to as direct oxidation process. There are a number of different process configurations for the modified-Claus process, depending mainly on the H_2S concentration in the Claus feed gas.

1.1.1. Chemistry and thermodynamics of the Claus process

In principle, ideal performance is achieved when all stoichiometric requirements for the basic process reactions are satisfied under the most favourable thermodynamic conditions, and equilibrium is reached at all points in the process. While thermodynamic favourability determines ideal performance, the practicable capability is dictated by the kinetic limitations imposed by the operating conditions and plant equipment.

Thermodynamics: The basic design of a typical modified-Claus plant can be best understood by looking at the thermodynamic equilibrium curves calculated for the reaction of pure hydrogen sulphide (H_2S) with air (Fig. 3):

$$H_2S + \frac{1}{2} O_2 \leftrightarrow \frac{1}{x} Sx + H_2O \qquad (5)$$



Number of catalytic reactors	Total sulphur recovery of a modified-Claus plant (%)
1	75-90
2	94-96
3	95-98

Table 1: Total Sulfur recovery of the modified-Claus process over the number of catalytic reactors used

Calculations are based on the principle of minimization of the Gibbs free energy. All three curves were calculated without Sulfur removal from the system. The difference between the upper and lower curves (see again Fig. 3) results from the Sulfur vapour species under consideration and the differences in thermodynamic data. The shape of the curves in Fig. 3 is a direct result of the temperature dependency of the Sulfur vapour composition shown in Fig. 4. High molecular weight species dominate at lower temperatures, and vice versa. Thus, for a fixed number of Sulfur atoms, fewer moles of Sulfur vapour are formed at lower temperatures. This decreases the Sulfur vapour partial pressure and tends to shift the equilibrium of reaction (5) to the right as well as increase the conversion. The opposite is true at higher temperatures. The same phenomenon causes the conversion to increase at low temperatures and decrease at high temperatures, as the total system pressure is increased. The theoretical degree of conversion is high at low temperature, falls off rapidly and passes through a minimum at 560°C, and then increases more slowly at higher temperatures. In the thermal stage region, it is not possible to reach Sulfur recoveries of over 70%.

Moreover, care must be taken to quench the reaction mixture rapidly in order to avoid reverse reaction. To convert more gases to Sulfur, thermodynamics suggests lower temperatures in the catalytic region. Before entering catalytic converters, elemental Sulfur must be condensed from the gas stream to prevent Sulfur condensation on the catalytic bed and improve thermodynamic equilibrium yields. For thermodynamic reasons, the catalytic unit should be operated at as low a temperature as possible above the Sulfur dewpoint, provided that the rate of the reaction is fast enough. In practice, Sulfur recovery is maximized by using two or more catalytic converters with Sulfur removal between each converter, and by decreasing temperature in successive converters.

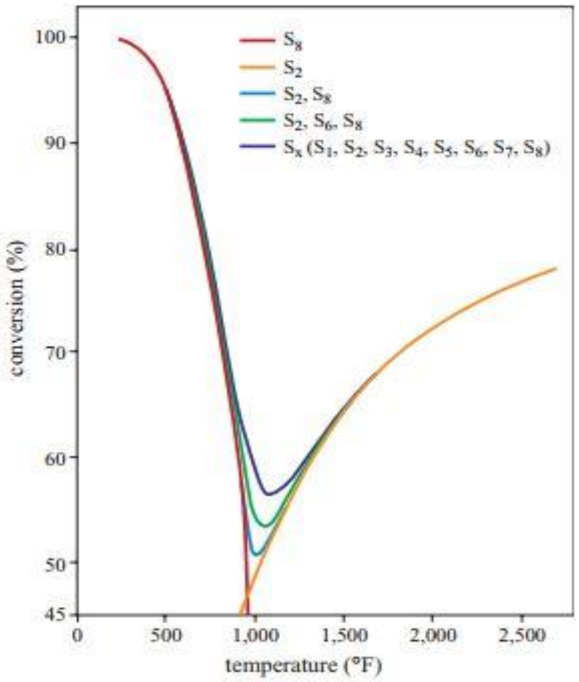


Figure 3: Equilibrium conversion of H_2S to elemental Sulphur (Paskall, 1979)

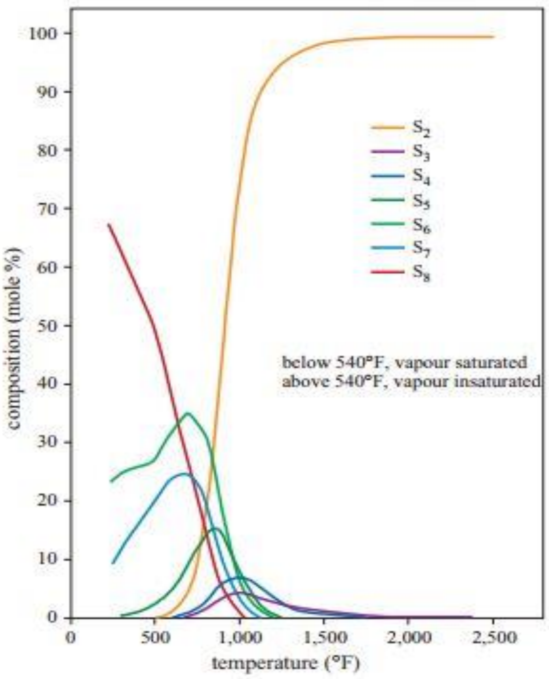


Figure 4: Equilibrium composition of Sulphur vapor from reaction of H_2S with stoichiometric air

1.1.2. Chemistry in the thermal stage

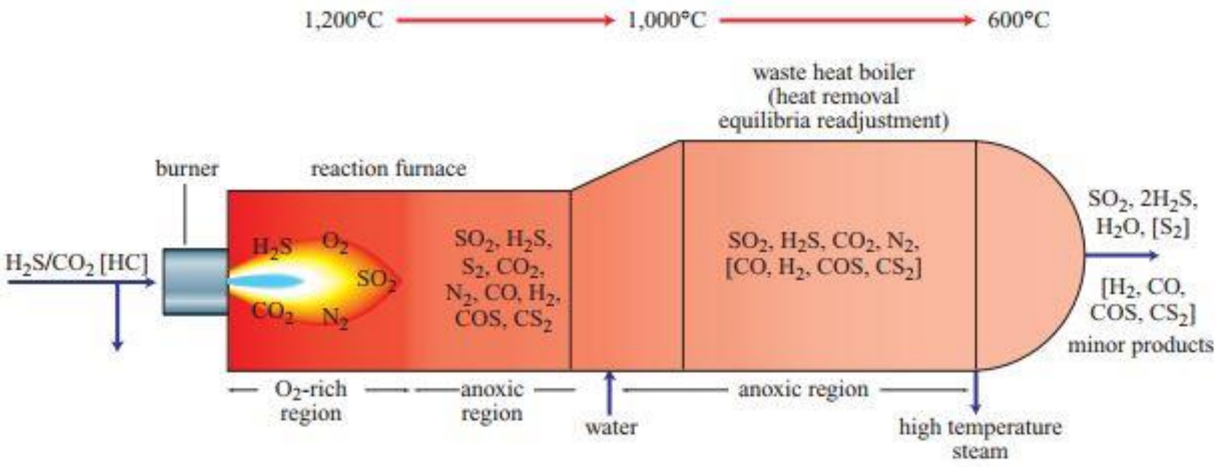


Figure 5: Expanded overview of Claus reaction furnace and waste heat boiler (Connock, 1999b).

Combustion processes occurring in the Claus reaction furnace are complicated (Connock, 1999 b). The presence of CO_2 and small quantities of hydrocarbons in the feed acid gas must be taken into account to ensure that gases leaving the WHB have the desired 2:1 H_2S/SO_2 ratio. CO_2 is particularly important as it becomes involved in a multitude of processes leading to CO and COS, as well as affecting the amount of hydrogen finally appearing in the product gas. Fig. 5 provides a simplistic overview of Claus furnace chemistry showing what are considered to be the most important reactions that occur in the furnace in the absence of hydrocarbon contaminants, but in the presence of CO_2. Chemistry occurring in the Claus reaction furnace may be split into two types: combustion reactions occurring in the oxygen-rich region; reactions proceeding in the oxygen-free (anoxic) region driven by the high temperature resulting from combustion reactions.

The carryover of hydrocarbons with the acid gas into the Claus reaction furnace further complicates the reactions. The hydrocarbon impurities may range from complex alkanes to BTX. Although thermodynamic considerations suggest that these hydrocarbons should fully combust to CO_2 and H_2O, the rate at which they do so is questionable since the CH bond is generally stronger than the SH bond. Thus, kinetic factors will affect the fate of the hydrocarbons in the reaction furnace, since they will be in competition with H_2S for a restricted oxygen supply. In the event that hydrocarbons are not fully combusted, it is expected that they will produce CO, C, COS and CS_2 through reactions with the Sulfur rich environment:

$CH_4 \rightarrow C + 2 H_2$ (6)

$C + H_2O \rightarrow CO + H_2$ (7)

$CO + H_2O \rightarrow CO_2 + H_2$ (8)

$CH_4 + 2 S_2 \rightarrow CS_2 + 2 H_2S$ (9)

$C + S_2 \rightarrow CS_2$ (10)

COS and CS_2 lower the Claus Sulfur recovery, unless their conversion to H_2S is achieved by hydrolysis at the relatively high temperatures found in the first catalytic converter. It has been well established that the design and mode of operation of the furnace can significantly influence the degree to which hydrocarbons are converted to COS and CS_2.

Ammonia (NH_3) destruction is also a problem when treating refinery Sour Water Stripper (SWS) off-gas. A particulare split-flow process is often applied in refinery SRU's that must process sour water stripper (SWS) off-gas and destroy the ammonia it contains.

1.1.3. The industrial Claus processes and their evolution

The Claus is the technology of choice for gases containing high concentrations of H_2S and/or large quantities of Sulfur. Nevertheless, when H_2S content is lower than 30% and/or the amount of Sulfur is less than 10-20 tons of Sulfur per day (t S/d), some other processes are often more economical.

Rich acid gases (H_2S>50%): The H_2S content of the Claus acid gas feed encountered in most refineries is around 80%. It is treated in the simplest, straight-through Claus process where all of the acid gas is processed in the reaction furnace. According to the Claus reaction, the air needed for combustion is one-third of that required for the complete combustion of H_2S. Consequently, the Claus furnace operates far away from complete combustion in the straight-through design. A minimum temperature of 925°C is generally considered to sustain a stable flame. A higher flame temperature is often required to destroy contaminants when present.

Medium acid gases (10%<H_2S<50%): When H_2S concentration is low, the stability of the flame cannot be reached and the straight-through design is no longer a good choice. The following possibilities must therefore be chosen.

Acid gas bypass: When part of the feed gas is bypassed, the furnace operates nearer to complete combustion, if one considers the same quantity of air entering the Claus unit as in the straight-through design. This results in an increased flame temperature. The design is referred to as the split-flow design. The upper limit for acid gas bypass is two-thirds of feed gas, as the furnace must be operated under reducing conditions. However, if contaminants are present in the feed, they will remain in part and enter directly into the catalytic section of the Claus. Troubleshooting and/or instability of the Sulfur plant may then occur, since these contaminants highly contribute to deactivation and plugging of the catalytic converters. Nevertheless, when this solution is applicable, it is the simplest and most economical way to treat medium acid gas composition.

Feed preheat: In order to maintain or raise the flame temperature, combustion air preheating and acid gas preheating must be examined. Acid gas preheating is more difficult to apply when the acid gas is recovered from the amine regenerator at low pressure. Moreover, corrosion must be properly checked as thermal cracking of the acid gas constituents may occur. When considering a revamp of an existing straight-through unit, specific attention must be paid to the design of the burner in order to avoid corrosion and to enable the use of high temperature gas.

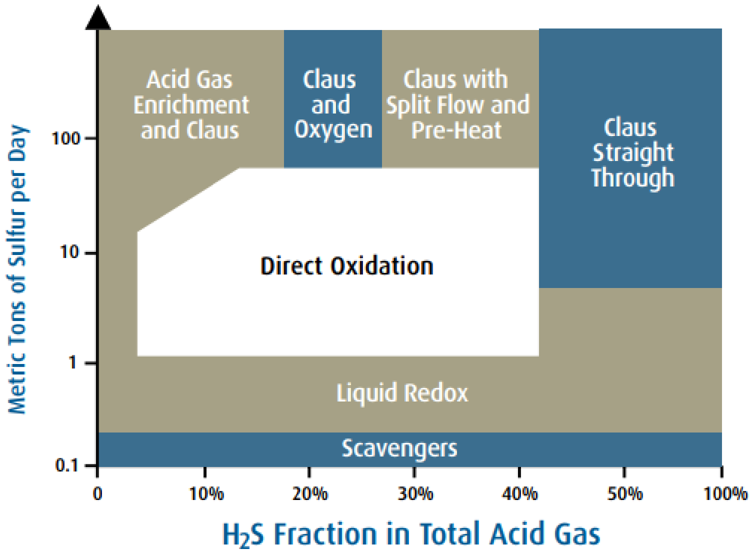


Figure 6: Sulphur Recovery Process Applicability Range

Oxygen enrichment: Oxygen enrichment raises the flame temperature by limiting the nitrogen effect of air. Moreover, this results in reductions of the equipment size and investment cost, since the global flow processed in the Sulfur plant is lower (Lee and Moore, 1997). Nevertheless, in order to be applied, this technology requires an available and economical source of oxygen. For the time being, only low-level enrichments (i.e. up to 28%) are addressed, since they lead to minor changes at the Claus unit and provide up to a 25% increase in capacity.

Fuel gas addition: Fuel gas can be added to ensure a high flame temperature, though this can bring about some disadvantages. Firstly, this will enlarge the size of the Sulfur plant and lower the total Sulfur recovery efficiency. Moreover, it could provoke catalyst deactivation or converter plugging, thus the consequences of fuel gas addition on the Sulfur recovery unit must be carefully examined.

Lean acid gas (5%<H_2S<30%): For this kind of gas, the Claus process is less competitive because it is difficult to ensure a stable flame at a low operating cost; consequently, some other technologies are of interest. One method is to replace the thermal section of the modified-Claus process by a catalytic section. Over the catalyst, air oxidizes H_2S to SO_2, which reacts with additional H_2S to produce elemental Sulfur.

Extra-lean acid gases (H_2S**<5%):** Several technologies are proposed depending on the H_2S content and the Sulfur tonnage to be recovered. A way to treat extra-lean acid gas is through direct oxidation. In this case, the higher the H_2S content, the lower the selectivity of the reaction, though the upper limit is around 1.5% H_2S. Other processes, including redox processes (wet oxidation) or non-regenerative processes, achieve very high recovery efficiency of nearly 100%. Among the redox, some utilize vanadium complex, as does the Stretford process. Others are based on iron chelates: Lo-Cat, Sulferox or Sulfint HP processes. The chemical consumption cost limits their application to low Sulfur tonnage (≈ 10 tS/d). When the quantity of Sulfur to be removed is very low (≈ 0.1 tS/d) scavengers may be used. These liquid or solid chemicals selectively react with H_2S. Iron-sponge and Sulfatreat caustic scrubbing are part of these processes. They are based on non-regenerative chemicals such as activated carbon, iron oxide, caustic solution or regenerative chemicals (e.g. triazine). The disposal cost for the used chemicals should be accurately considered in the global evaluation of the process.

1.1.4. Claus operating variables

The overall Sulfur recovery efficiency of a modified-Claus unit is greatly dependent on the design, maintenance and operation of the unit. The most important control variable in the operation of Claus plants is the ratio of H_2S to SO_2 in the gases entering the catalytic converters. Maximum conversion requires that this ratio is maintained constant at the stoichiometric proportion of 2 moles of H_2S to 1 mole of SO_2. Appreciable deviation from the stoichiometric ratio leads to a drastic reduction in conversion efficiency. Several methods, based on controlling the air flow by continuous analysis of the ratio of H_2S to SO_2 in the plant tail gas, have been developed and are used ever more frequently. Several analytical instruments based on vapour chromatography and ultraviolet absorption are available commercially. In addition, it is very important to operate the different catalytic converters at the right temperature (Bohme and Sames, 1999). For example, the condensation of elemental Sulfur on the catalytic converters can be avoided by maintaining their temperature above the Sulfur dew point of the gas mixture.

When the feed gas contains contaminants, a higher flame temperature is often required to destroy them. Ammonia, heavy hydrocarbons, BTX, mercaptans and cyanides are among the contaminants most often encountered. As regards ammonia, it is typically provided by the off-gas of the SWS of the refinery. The Claus unit is considered the best place to destroy this off-gas, which also contains H_2S, thanks to its high furnace temperature. NH_3 must be destroyed in the reaction furnace, otherwise Sulfur trioxide (SO_3) can form due to the following reactions:

$$2\ NH_3 + \frac{5}{2}\ O_2 \rightarrow 2\ NO_3 + H_2O \qquad (11)$$

$$2\ NO + O_2 \rightarrow 2\ NO_2 \qquad (12)$$

$$SO_2 \xrightarrow{[NO2]} SO_3 \qquad (13)$$

SO_3 causes severe downstream problems like corrosion, catalyst deactivation and salt formation. Accumulation of such ammonium salts would lead to unreliable operation and unacceptable maintenance costs. Two methods are available to successfully destroy NH3 using the following reaction:

$$2\ NH_3 + \frac{3}{2}\ O_2 \rightarrow N_2 + 3\ H_2O \qquad (14)$$

The first method involves a split-flow reaction furnace design; the second requires a high intensity reaction furnace burner. It is essential for NH3 to be almost completely destroyed, being that ammonia concentrations even as low as 500-1,000 ppm by volume (ppmv) can cause problems. As for BTX and heavy hydrocarbons, acid gas may contain these compounds since solvent units are able to co-absorb them together with H_2S. Feed stream must be analysed and the necessary measures must be taken to keep them from entering the Claus catalytic section, as they may cause catalyst deactivation, plugging and off-specification Sulfur for market purposes (e.g. green Sulfur compared to bright yellow Sulfur). When acid gases of high H_2S content are processed, the temperature in the reaction furnace is usually high enough to result in the complete combustion of all hydrocarbons to CO_2 and water, and no carbonaceous material deposition is experienced. However, at the low combustion temperature occurring in straight-through plants, processing gases with less than approximately 40-50% H_2S, cracking and partial combustion of hydrocarbons produce complex carbonaceous materials that are carried into the catalytic reactors, gradually deteriorating catalyst performance. In addition, hydrocarbons can be fed directly to the first catalytic converter without being burned when a split-flow design is utilized. These hydrocarbons can also cause catalyst deterioration. Regarding mercaptans and cyanides, the same general considerations must be applied for the Claus unit. For other processes, such as redox or scavenger processes, problems connected to chemical odours and disposal must be carefully considered.

2. Safety and environmental issues

Liquid Sulfur produced in the modified-Claus process contains 150-400 ppm wt of residual hydrogen sulphide in the form of both dissolved hydrogen sulphide (H_2S) and hydrogen polysulphides (H_2S_x) in equilibrium with Sulfur. H_2S/H_2S_x equilibrium is temperature dependent: at 130°C, $H_2S/H_2S_x \approx 10$, and at 150°C, $H_2S/H_2Sx \approx 1$. The decomposition of H_2S_x to elemental Sulfur is a very slow reaction:

$$H_2S_x \text{ (liquid)} \leftrightarrow H_2S \text{ (dissolved)} + (x\text{-}1)\ S \text{ (liquid)} \qquad (15)$$

H_2S dissolved in the liquid phase passes into the gaseous phase by physical desorption:

$$H_2S \text{ (dissolved)} \leftrightarrow H_2S \text{ (gas)} \qquad (16)$$

The H_2S content in the atmosphere becomes progressively more dangerous above 50 ppmv and is even lethal at 600 ppmv. The lower explosion and flammability limit of H_2S is approximately 3.5% by volume at 150°C in air. During handling, storage and transportation, when Sulfur is agitated and cooled, H_2S is released from the Sulfur. The H_2S concentration in the surrounding atmosphere can reach toxic level, fire and explosion limits. Therefore, the industry must produce liquid or solid elemental Sulfur with a maximum value of 10 ppm wt of hydrogen sulphide ($H_2S + H_2S_x$). Liquid Sulfur should be degassed for the following main reasons: a) safe storage and transportation of liquid Sulfur; b) safer working conditions for personnel handling liquid Sulfur; c) less corrosion in Sulfur storage tanks, transport piping and road tankers/ships; d) lower hydrogen sulphide emissions into the atmosphere; e) higher Sulfur strength thanks to the formation of polymeric form of Sulfur S_x; the solid Sulfur produced from undegassed liquid Sulfur is more friable.

The principle of the degasification processes of liquid Sulfur is to release the dissolved H_2S gas and to accelerate the decomposition of the H_2S_x to H_2S. The release of the dissolved H_2S is obtained by agitation of the liquid Sulfur. The decomposition of the H_2S_x to H_2S can be accelerated by means of a catalyst, reducing the energy consumption and the equipment size. The released H_2S gas must be removed from the gas space above the Sulfur by flushing with a sweep gas (air, Claus tail gas or inert gas). Air seems to be the best sweep gas. Indeed, tail gas still contains residual H_2S, therefore the degassing rate is reduced because the H_2S available in the liquid Sulfur tends to be in equilibrium with H_2S in the gaseous phase. Air contributes to degassing by promoting the direct oxidation of H_2S to elemental Sulfur through oxygen. Using air as sweep gas prevents accumulation of pyrophoric iron sulphide (FeS) formed on carbon steel surface by reaction with H_2S. This accumulation generally occurs when using inert gas sweeping. The physical desorption of dissolved H_2S shifts to the right the equilibrium of reactions (15) and (16). Sweep gas quantity should be designed in such a way that the H_2S concentration is 1.5% by volume, maximum.

3. The AG2S™ Technology

Practically every refinery is equipped with a well-known Claus Sulfur recovery unit (SRU). Flue gases rich in CO_2 are currently vented, contributing to climate changes; sequestration of CO_2 is technically viable, but remote disposal still remains a questionable end. As a consequence, H_2S and CO_2 lead to significant capital, operating and financial costs, burdening on plants yearly budget. On the other hand, hydrocarbon industry is hungry of hydrogen or mixtures of hydrogen and carbon monoxide (syngas); consequently, refineries and petrochemical plants are equipped, for instance, with steam reforming units where steam and light hydrocarbons react together to produce syngas (e.g., $H_2O + CH_4 = 3\ H_2 + CO$).

It is remarkable that H_2S is both a potential source of hydrogen and a potential reducing agent for CO_2. Also, the reacting chemical system constituted of $H_2S + CO_2$ looks like traditional and proven Claus chemical systems $H_2S + O_2$ (thermal section) and $H_2S + SO_2$ (catalytic section). Accordingly, it is reasonable to envisage that syngas production from H_2S and CO_2 can be achieved under proper conditions.

Actually, from detailed thermodynamic-kinetics studies, oxy-reduction reaction between H_2S and CO_2 results feasible. Potential industrial benefit is threefold: acid gases from deSulfurization are neutralized, CO_2 is promoted as chemical feed and syngas mixture is produced. A description about this process and technology, follows along with a comparison with traditional Claus SRU.

3.1. Description of the synthesis route

Reaction between H_2S and CO_2 can take place at temperature >1000°C in a gas-phase reactor (hereinafter called "RTR" – Regenerative Thermal Reactor); high temperatures are necessary to (i) activate the reactive system from chemical-thermodynamics standpoint, (ii) quicken kinetics and (iii) reduce by-products. Practically, the reactants are injected into a common chamber in pre-mixed or un-mixed mode, and then reacted. As a result, these contaminants are "regenerated" into valued products. In Fig. 7 a concept process flow diagram is given.

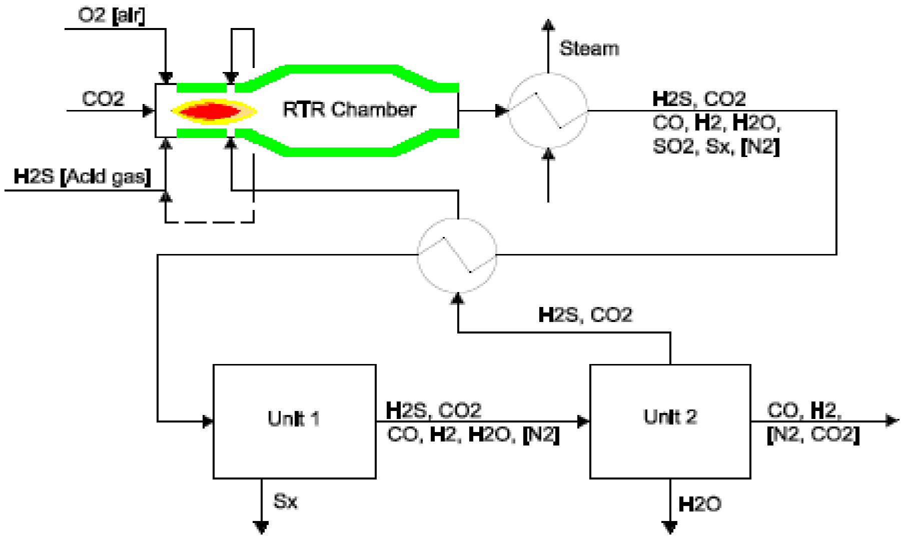


Figure 7: AG2S™ technology process flow diagram [12]

The regenerative thermal oxy-reduction between H_2S and CO_2 occurs in the RTR refractory lined chamber, assisted by a minor injection of either air or oxygen; RTR effluent, constituted of products and unreacted feed species, is quenched and cooled in heat exchangers. The effluent is subsequently processed in a catalytic unit (Unit 1), analogous to Claus SRU section, where it is cooled down by means of a train of heat exchangers (Sulfur condensers) and treated in catalytic reactors:

- The cooling allows separating condensed species (S_x) from main gaseous stream;

- A catalytic treatment is necessary to convert SO_2 into elemental Sulfur. Since SO_2 amount at outlet of RTR is minor, only one Claus catalytic reactor is enough in Unit 1;

- A second catalytic treatment is necessary to convert by-products, as COS and CS, residual SO_2 and Sulfur vapours into H_2S and CO_2. This improves overall process selectivity on Sulfurous species, but also reduces the process yield as a portion of CO is consumed and CO_2 is produced.

Finally, effluent undergoes a separation treatment (Unit 2) where water is knocked-down, unreacted H_2S and CO_2 are recycled to RTR and an H_2/CO rich mixture is exported.

The RTR looks like the Claus SRU thermal reactor from reaction kinetics and reactor engineering standpoint. As a consequence, investigating approach and tools used for SRU can be also adopted for the RTR. Reaction mechanism between hydrogen sulphide (or acid gases) and carbon dioxide is complex and involves thousands of chemical reactions and more than 100 molecular and radical species. Specific thermodynamic-kinetics models must be used for a detailed study of RTR. Following reaction macro-steps are expected to occur in the RTR:

1) thermal decomposition of H_2S into hydrogenated free radicals (SH, H, …) (thermal activation);

2) CO_2 reduction to CO by free radicals (H, OH, …);

3) propagation of oxygen-based free radicals;

4) formation of Sulfurous oxides (for example, SO) and inhibition of backward reaction from CO to CO_2 (according to inhibition effect observed by Mueller);

5) formation of stable SO_2 from SO;

6) formation of H_2.

Resulting overall RTR reaction is $CO_2 + 2\ H_2S \rightarrow CO + H_2 + S_2 + H_2O$, plus by-products and unreacted feed species. It is to be highlighted that such reaction is chemically more noble than traditional Claus reaction, as the hydrogen in H_2S is not nailed in water (and then lost forever), but it is freed to form H_2. More detailed description on the synthesis route can be found in literature [3].

The above synthesis route realizes to be also superior from thermodynamic standpoint than traditional Claus thermal unit. The overall reaction of Claus thermal oxidation ($H_2S + 0.5\ O_2 = H_2O + 0.5\ S_2$) is strongly exothermic ($\Delta H^0 = -156.98$ kJ/mol): as a common practice, released heat is recovered generating high and low pressure steam, which is a practical but poor method from thermodynamic standpoint. On the contrary, since RTR reaction is endothermic ($\Delta H^0 = +210.97$ kJ/mol), the heat balance can be adjusted by burning a proper amount of H_2S in order to minimize generation of steam, and therefore improving the thermodynamic efficiency of the process.

3.2. The technology

Synthesis of syngas starting from H_2S and CO_2 relies on the RTR feasibility and overall plant efficiency. Similitude between RTR and Claus thermal reactor and, in general, between the proposed synthesis route and traditional Claus Sulfur recovery unit, is encouraging for process and technological design.

As per Fig.7, fresh reactants (CO_2 and H_2S coming from upstream plants, assisted by air or oxygen) are injected into the RTR along with recycled reactants; recycle is not necessary for the thermal oxy-reduction, but it is essential for improving self-sustainability of the plant from process standpoint, that is, for reduction or elimination of exhausts. Since RTR reaction is activated at high temperature only, reactants can be fed in either pre-mixed or un-mixed mode.

The recycle stream pressure corresponds to the Unit 2 operating pressure; since this is higher than the RTR one, recycled H_2S and CO_2 can be injected into RTR without need of a blower. On the other

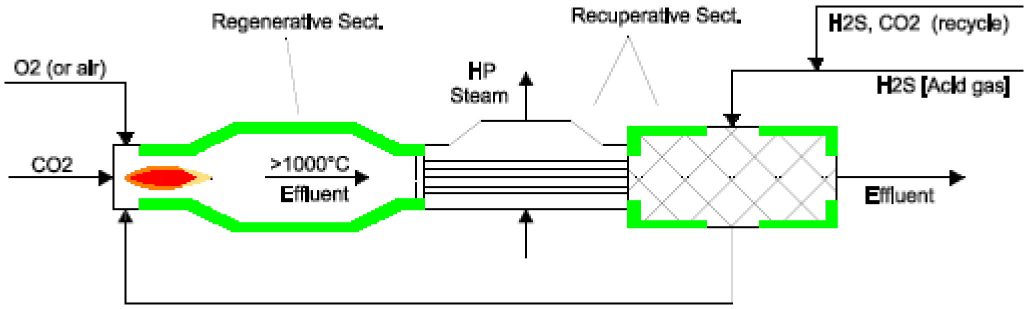


Figure 8: an example of a possible RTR arrangement [12]

hand, the recycle must be pre-heated in some way because its injection at low temperature into RTR quenches the oxy-reduction. Different methods are possible for pre-heating. One of the most conventional is to install a heat exchanger where cold recycle and an external hot fluid cross. Alternatively, recycle stream temperature can be raised by an in-line burner, where natural gas or H_2S are combusted by means of an injection of oxygen or air. Finally, a third and energetically efficient solution can be envisaged: to install a feed/effluent type heat exchanger downstream of the RTR (as shown in Fig. 7). In practice, recycle pre-heating can be done by a combination of the mentioned methods. The recycle can be also mixed with fresh reactants before pre-heating and injection into the reactor.

To assure a proper production yield, it is essential to minimize regressive reactions during cooling of RTR effluent; for this purpose, a waste heat boiler (WHB) installed just at outlet of RTR chamber is the best as it can quench the reactions. The WHB and recycle pre-heating equipment play a key role in the regenerative process, therefore they could be considered a portion of the RTR. Fig.8 shows one of the possible configurations of the RTR and heat exchangers: (i) the recycle is mixed with fresh reactants, pre-heated in the feed/effluent exchanger and then injected into the RTR, (ii) the oxy-reduction takes places at high temperature in the refractory lined chamber, (iii) effluents are quenched in the boiler and used for preeating. On the whole, the RTR could be also seen as a system constituted of a “regenerative” and a“recuperative” section, and not just the reaction chamber itself.

Fig.9 and Tab.2 report some thermodynamics-kinetics simulation results, based on *Politecnico di Milano*’s OpenSMOKE++ software, for a simplified RTR reaction and reactor: it is assumed that (i) fresh reactants (CO_2, H_2S and O_2) are pure and with exact stoichiometric ratio, (ii) there is no recycle, and (iii) the reactor is ideal plug-flow and adiabatic. Reaction temperature is set to 1300°C approx., at an operating pressure of 150kPa(a). As shown in Fig. 9, CO_2 and H_2S instantaneously react; with a residence time around 0.4s the system can be considered at thermodynamic equilibrium. RTR effluent is rich in syngas species and elemental Sulfur whereas H_2S and CO_2 concentrations are significantly reduced; conversely to traditional Claus thermal reactor, SO_2 concentration is minor.

It is also to be underlined that (i) H_2/CO ratio strongly depends on the residence time and (ii) reactants conversion is strictly related to reaction temperature. This is shown in Tab.2, where two effluent compositions are given for two different oxy-reduction temperatures: higher the reaction temperature, higher the syngas species yield, nevertheless, higher the SO_2 as well. As a consequence, as syngas and SO_2 represent a profit and a cost respectively, the process is characterized by an optimum. The reaction temperature can be easily adjusted by O_2 flowrate.

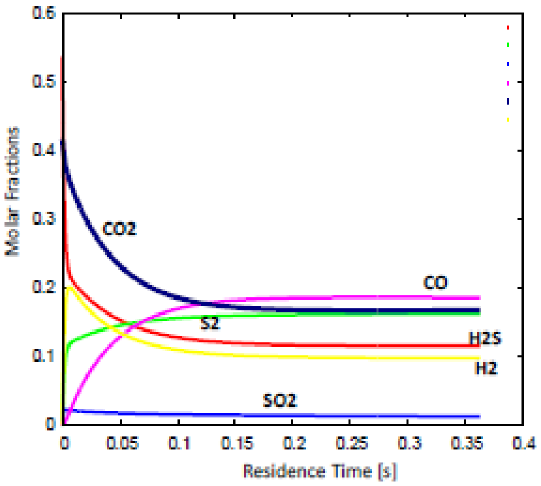
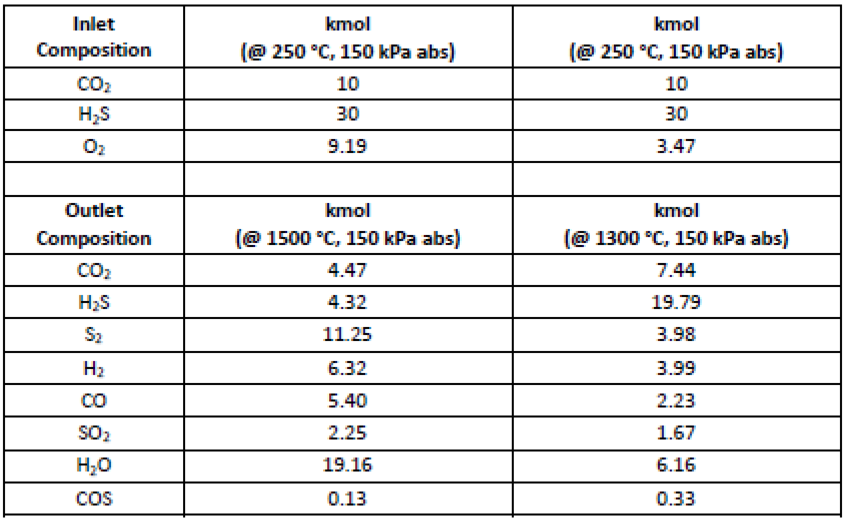


Figure 9: concentration profiles in the RTR [12]



Inlet Composition	kmol (@ 250 °C, 150 kPa abs)	kmol (@ 250 °C, 150 kPa abs)
CO_2	10	10
H_2S	30	30
O_2	9.19	3.47
Outlet Composition	kmol (@ 1500 °C, 150 kPa abs)	kmol (@ 1300 °C, 150 kPa abs)
CO_2	4.47	7.44
H_2S	4.32	19.79
S_2	11.25	3.98
H_2	6.32	3.99
CO	5.40	2.23
SO_2	2.25	1.67
H_2O	19.16	6.16
COS	0.13	0.33

Table 2: Input and output RTR compositions for two different reactions temperatures [12]

As mentioned, Unit 1 corresponds to a traditional catalytic Claus SRU section; although stoichiometry of RTR effluent is different, its qualitative chemical composition is the same of catalytic Claus gas, therefore the required equipment is also the same. Important to underline, since SO_2 amount coming from RTR is minor with respect to Claus thermal section, SO_2 can be removed by H_2S in one catalytic passage (Claus reaction: $2\ H_2S + SO_2 = 1.5\ S_2 + 2\ H_2O$). A second and final catalytic reactor provides for hydrogenation of residual traces of SO_2 and Sulfur vapours ($xH_2 + S_x = xH_2S$); on the hydrogenation catalyst, COS/CS_2 hydrolysis and shift conversion take place as well. As a result, hydrogenation step has the intrinsic advantage to produce H_2, but also the intrinsic disadvantage to consume CO and produce CO_2.

Separation of syngas species from H_2S and, partially, from CO_2 is accomplished in Unit 2. In order to completely remove water, a contact condenser may be necessary before the washing. Due to acid behaviour of H_2S and CO_2, Unit 2 is conceived as a standard chemical washing unit. An amine-based solution can be used for knocking-down H_2S and CO_2 in an absorbing column and then in a stripping column, H_2S and CO_2 are recovered and recycled to RTR. CO_2 in excess leaves along with the syngas: thus, a subsequent separation, like washing or selective membrane, may be necessary according to the use of gas.

3.3. RTR vs Claus SRU

Above synthesis route can potentially replace a traditional Claus Sulfur recovery unit; in this case, acid gas coming from upstream amine unit represents the fresh reactant; again, a minor injection of air or oxygen is necessary. In Fig.10 a possible process scheme is reported (stream number reported in brackets). Simulations have been carried-out by a commercial process package coupled with aforementioned SRU thermodynamics-kinetics data base. Fig.10 scheme is based on pre-heating by an external hot fluid (HP steam) and an injection of fuel gas in the RTR (cofiring) for assuring proper reaction temperature.

For comparative reasons, reaction temperature in the RTR is kept at 1180°C approx. Hot effluent is completely quenched and cooled in the WHB till to 390°C. Effluent from RTR is rich in hydrogen and carbon monoxide; these syngas species passively flow till to the hydrogenation reactor, where significant portions of CO and CO_2 are consumed and produced, respectively. Finally, gas exported from Unit 2 is rich in hydrogen. Such a gas, after proper purification, can be used in hydrogenation processes or, after partial CO_2 removal, as a syngas. For instance, required stoichiometric ratio for methanol synthesis is reached by removing two thirds of carbon dioxide. Therefore, a subsequent purification/washing operation could be necessary depending on the use.

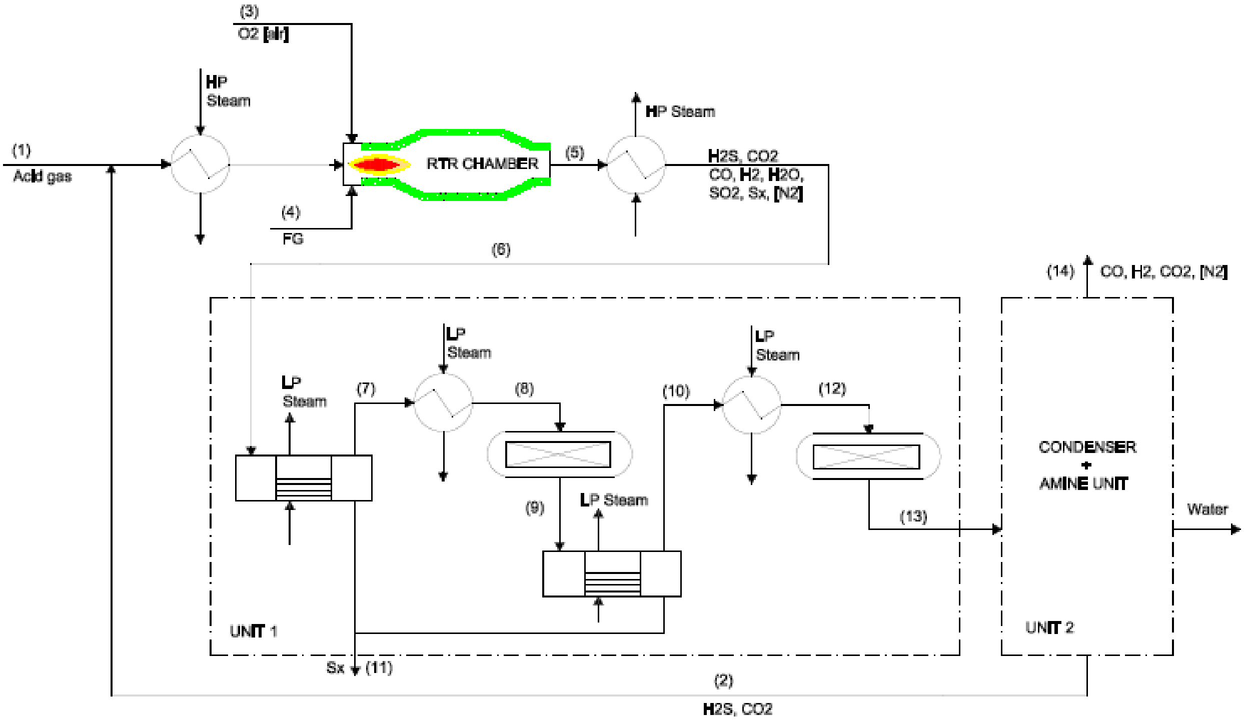


Figure 10: Process plant scheme with pre-heating by external source [12]

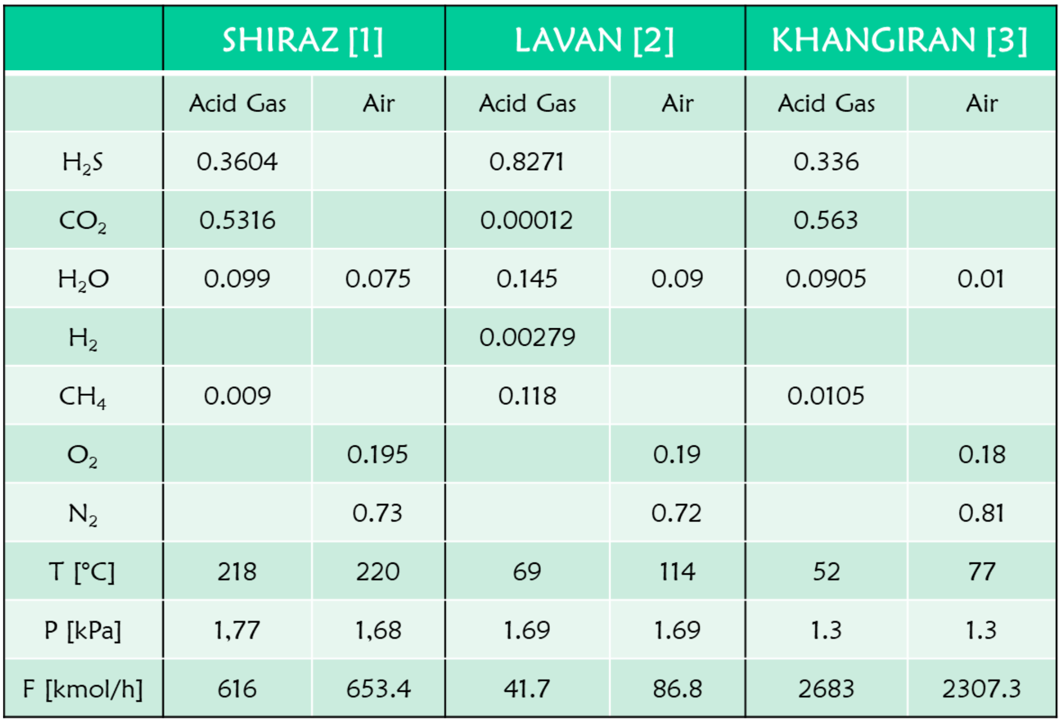
PART 1 – DATA VALIDATION

The main objective of the first part of this thesis work was to validate a recent kinetic scheme, developed by Ayoub El Ziani during his M.Sc. Thesis at Politecnico di Milano, relative to the oxy-reduction conversion of H_2S and CO_2, complete with detailed mechanisms about the pyrolysis and the oxidation of typical tail streams compounds, rich in C H O N and S elements.

1. Literature data

As usual when the research is directed towards data of industrial origin, their availability in literature is quite scarce. Only three papers reported updated, even if only partially complete, values related to real Claus furnaces currently operated, as their aim was to model a suitable set of reactions able to kinetically reproduce, with good agreement, what goes on in the thermal section of a SRU unit.

Table 3 shows the collected data, followed by a recap of the simulation techniques used by each one of the three autors. This is useful in order to address once more the state of the art, this time regarding the modelling and the simulation of a Claus furnace environment; as this is exactly one of the main focal points of this work.

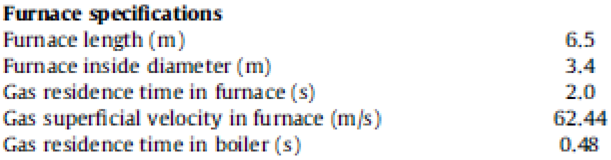


	SHIRAZ [1]		LAVAN [2]		KHANGIRAN [3]	
	Acid Gas	Air	Acid Gas	Air	Acid Gas	Air
H_2S	0.3604		0.8271		0.336	
CO_2	0.5316		0.00012		0.563	
H_2O	0.099	0.075	0.145	0.09	0.0905	0.01
H_2			0.00279			
CH_4	0.009		0.118		0.0105	
O_2		0.195		0.19		0.18
N_2		0.73		0.72		0.81
T [°C]	218	220	69	114	52	77
P [kPa]	1,77	1,68	1.69	1.69	1.3	1.3
F [kmol/h]	616	653.4	41.7	86.8	2683	2307.3

Table 3: Input compositions and conditions of the 3 industrial Claus furnaces in exam

1.1. Shiraz

This paper was aimed to develop an accurate model of a SRU unit, based on a rigorous kinetic study such that all of the major reactions occurring in the thermal and catalytic stage are considered. Simulation results have then been compared to actual industrial data referred to a plant operating in Shiraz, Iran.



Furnace specifications	
Furnace length (m)	6.5
Furnace inside diameter (m)	3.4
Gas residence time in furnace (s)	2.0
Gas superficial velocity in furnace (m/s)	62.44
Gas residence time in boiler (s)	0.48

Figure 11: Shiraz industrial furnace specifications

In the presence of high temperature and due to the nature of the reactants, that makes a lot of reactions occur in the furnace and catalytic reactors, the authors chose to consider only 16 major reactions in thermal reactor and 6 reactions in catalytic reactors: the one considered to be the most probable ones to admit accurate results with shorter calculation time and good agreement with industrial data.

Key reactions, such as H_2S pyrolysis and re-associations, COS and CS_2 production and main hydrocarbons combustion reactions are considered in this study. As well, some less affecting reactions like reaction between CO and H_2, and CO/CO_2 equilibrium are all considered. Moreover production and distribution of the Sulfur allotropes are considered and determined. Complete list of considered reactions are presented for both furnace and reactors in Tables 4 and 5, respectively.

Following assumptions are considered in kinetic modelling of the furnace:

- Ideal gas condition because of low pressure conditions.
- Steady state conditions.
- Adiabatic condition, because furnace is completely insulated from inside refractory layer and outside.
- Combustion chamber operates as like as plug flow reactors in mass and energy balances, because of the high inlet flow rate and therefore high mass and heat Peclet number (> 500).
- Radial dispersion is ignored, because of the turbulent fluid regime.
- Hagen–Poiseuille equation is considered for pressure drop along the furnace as the following:

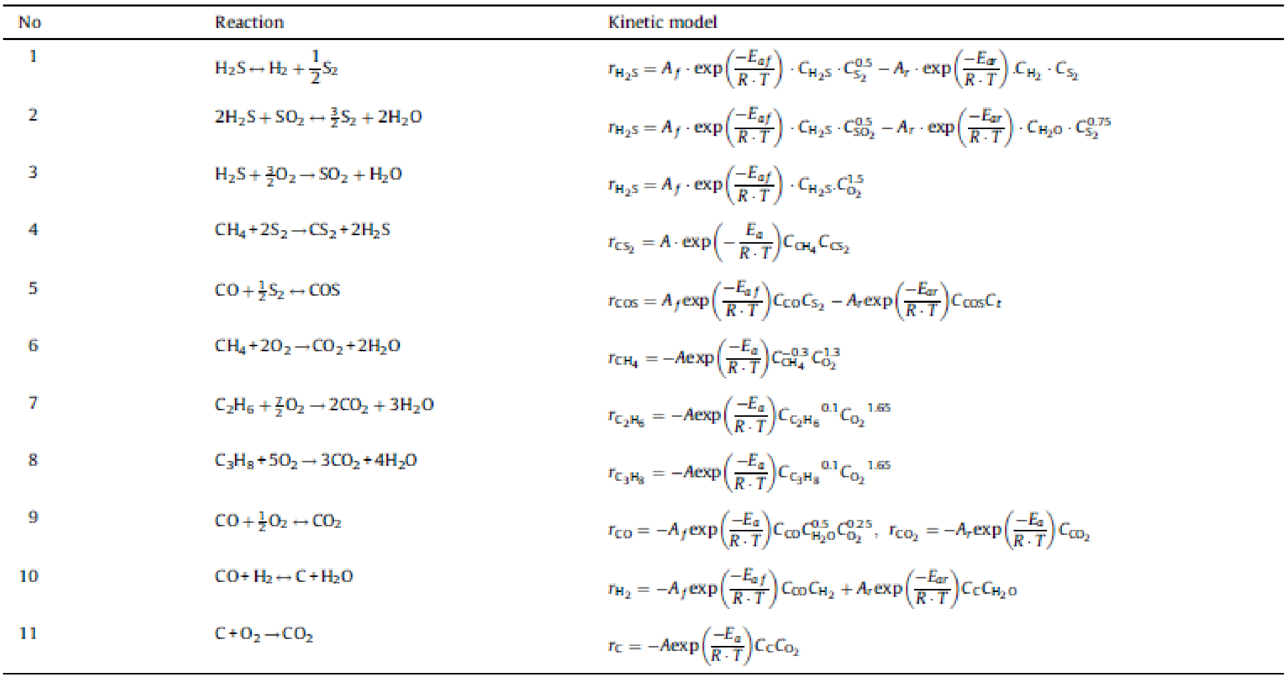
$$\frac{dp}{dv} = \frac{32\mu v}{D^2 A}$$

This equation is valid for the present case study with the Poiseuille flow number less than 0.5. It can be noted that total furnace (including burner, combustion chamber and WHB) pressure drop is 10–20 kPa (less than 7% of inlet pressure) and combustion chamber pressure drop is only less than 3% of inlet pressure.

- The furnace is assumed as a plug flow reactor.
- The reaction kinetics are described on the base of species concentrations which may be substituted with component flow rates (F_i) using the ideal gas equation.

The ordinary differential equations derived from mass and energy balances were solved by Runge–Kutta order V and the numerical algorithm was written in MATLAB programming software, version 2012b.

In this article, the equilibrium model is solved using the commercial simulator with the same operational conditions and the feed condition, whereas kinetic models are solved by MATLAB programming software. The results of the models are compared with each other and with the real industrial case unit data.



No	Reaction	Kinetic model
1	$H_2S \leftrightarrow H_2 + \frac{1}{2}S_2$	$r_{H_2S} = A_f \cdot \exp\left(\frac{-E_{af}}{R \cdot T}\right) \cdot C_{H_2S} \cdot C_{S_2}^{0.5} - A_r \cdot \exp\left(\frac{-E_{ar}}{R \cdot T}\right) . C_{H_2} \cdot C_{S_2}$
2	$2H_2S + SO_2 \leftrightarrow \frac{3}{2}S_2 + 2H_2O$	$r_{H_2S} = A_f \cdot \exp\left(\frac{-E_{af}}{R \cdot T}\right) \cdot C_{H_2S} \cdot C_{SO_2}^{0.5} - A_r \cdot \exp\left(\frac{-E_{ar}}{R \cdot T}\right) \cdot C_{H_2O} \cdot C_{S_2}^{0.75}$
3	$H_2S + \frac{3}{2}O_2 \rightarrow SO_2 + H_2O$	$r_{H_2S} = A_f \cdot \exp\left(\frac{-E_{af}}{R \cdot T}\right) \cdot C_{H_2S} . C_{O_2}^{1.5}$
4	$CH_4 + 2S_2 \rightarrow CS_2 + 2H_2S$	$r_{CS_2} = A \cdot \exp\left(-\frac{E_a}{R \cdot T}\right) C_{CH_4} C_{CS_2}$
5	$CO + \frac{1}{2}S_2 \leftrightarrow COS$	$r_{COS} = A_f \exp\left(\frac{-E_{af}}{R \cdot T}\right) C_{CO} C_{S_2} - A_r \exp\left(\frac{-E_{ar}}{R \cdot T}\right) C_{COS} C_t$
6	$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$	$r_{CH_4} = -A\exp\left(\frac{-E_a}{R \cdot T}\right) C_{CH_4}^{-0.3} C_{O_2}^{1.3}$
7	$C_2H_6 + \frac{7}{2}O_2 \rightarrow 2CO_2 + 3H_2O$	$r_{C_2H_6} = -A\exp\left(\frac{-E_a}{R \cdot T}\right) C_{C_2H_6}{}^{0.1} C_{O_2}{}^{1.65}$
8	$C_3H_8 + 5O_2 \rightarrow 3CO_2 + 4H_2O$	$r_{C_3H_8} = -A\exp\left(\frac{-E_a}{R \cdot T}\right) C_{C_3H_8}{}^{0.1} C_{O_2}{}^{1.65}$
9	$CO + \frac{1}{2}O_2 \leftrightarrow CO_2$	$r_{CO} = -A_f \exp\left(\frac{-E_a}{R \cdot T}\right) C_{CO} C_{H_2O}^{0.5} C_{O_2}^{0.25}, \ r_{CO_2} = -A_r \exp\left(\frac{-E_a}{R \cdot T}\right) C_{CO_2}$
10	$CO + H_2 \leftrightarrow C + H_2O$	$r_{H_2} = -A_f \exp\left(\frac{-E_{af}}{R \cdot T}\right) C_{CO} C_{H_2} + A_r \exp\left(\frac{-E_{ar}}{R \cdot T}\right) C_C C_{H_2O}$
11	$C + O_2 \rightarrow CO_2$	$r_C = -A\exp\left(\frac{-E_a}{R \cdot T}\right) C_C C_{O_2}$

Table 4: List of considered reactions in Claus thermal reactor]

The equilibrium model has shown the most agreement to the industrial data and kinetic model has shown the better results in some cases. COS and CS_2 are the most challenging components: in spite of more accurate results of outlet components in equilibrium model, deviations of COS and CS_2 are much greater than those in the kinetic model. It is revealed that COS and CS_2 production are not consistent with equilibrium assumptions and they follow kinetically limited mechanism. In addition, deviation of kinetic model is distributed more homogeneously than equilibrium model. In kinetic model all components have shown deviation smaller than 13% (except than CS_2), whereas in equilibrium model, some of the components deviate around 88% and some other deviates around 0.0%. Standard deviation of kinetic model is 26.24 while equilibrium model has a standard deviation of 135.36 from industrial case.

Fig. 12 shows temperature profile in combustion chamber, calculated by kinetic model. According to Fig. 12 temperature soars to as high as 1070 °C in the inlet section of combustion chamber and then because of endothermic reactions, begins to decrease gradually and finally approaches 972 °C. As an average along the bed, the temperature is around 1000 °C in the combustion chamber.

Fig. 13 shows H_2S, CO_2CO_2 and H_2O molar flow profiles along the combustion chamber. According to the profiles, H_2S composition decreases with a sharp rate at the beginning of the gas inlet to the furnace. CO_2 flow increases sharply at the inlet because of fast hydrocarbons combustion and then decreases because a part of CO_2 converts to CO.

Fig. 14a shows CO, COS, SO_2, S_2 and H_2 composition profiles along combustion chamber length. Compositions of H2, S_2, COS and CO products increase sharply at initial length of the furnace then because of consumption of H_2 in the reaction with syngas products its composition decreases. Similar to H_2, SO_2, SO_2 molar flow decreases after a while and this is because of SO_2 consumption in Claus reaction. After 0.5 m from the furnace inlet, CO composition remains constant until the end of thermal reactor. Because of limited resources of oxygen, a part of hydrocarbons burns incompletely and produces more CO. In addition, the burning reaction of H_2 cannot progress completely because of low availability of oxygen.

Fig. 14b shows conversion of key components in combustion chamber. In this figure negative amount of conversion means production of a component and black point at the end of each profile shows conversion in real industrial data. It is obvious that all available oxygen consumes at the inlet of reactor and therefore its conversion soars to 100%, instantly. Because of high tendency of H_2S to react with oxygen, H_2S conversion has a high slope at the reactor inlet and then because of oxygen lack, the rate of SO_2 formation decreases. The same results can be seen for CO_2. At the inlet of combustion chamber, all hydrocarbons react with oxygen which produces large amounts of CO_2 that its production is presented by negative conversion. As the reaction progresses, conversion trend of CO_2 revolves from negative to positive amount that is because of consumption of CO_2 in other reactions. H_2O is the product of all combustion reactions, therefore its composition usually increases and its conversion plot is negative.

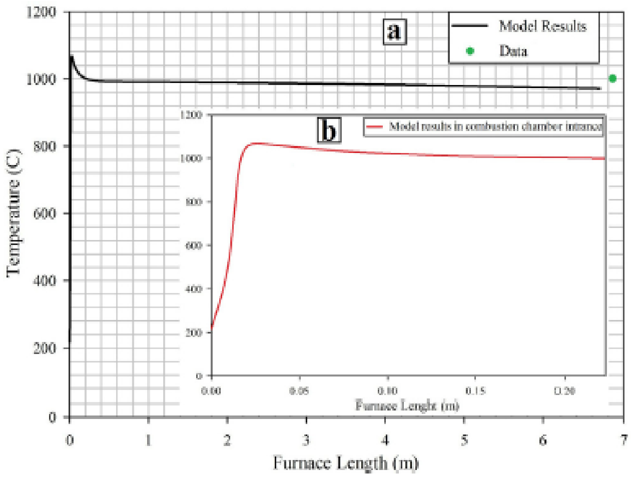


Figure 12: Temperature profile along combustion chamber length (a) total length (b)

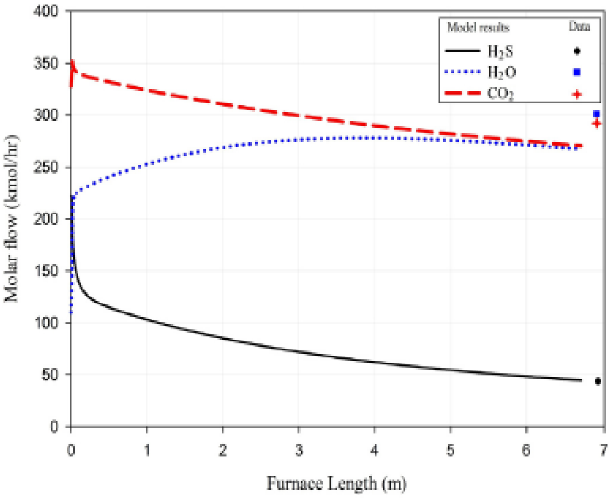


Figure 13: H_2S, CO_2 and H_2O composition profiles along combustion chamber (end points are industrial data).

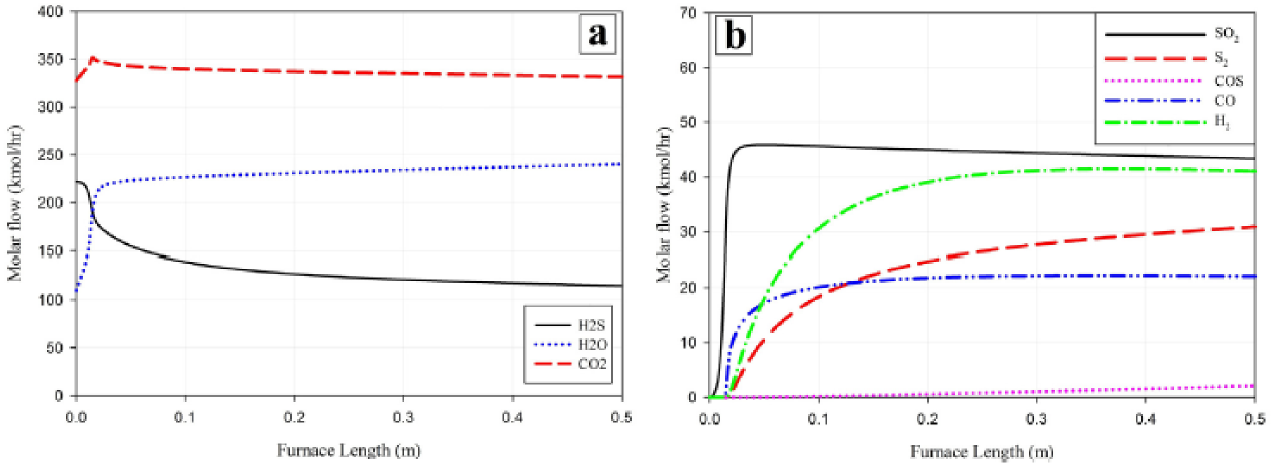


Figure 14: (a) H_2S, H2O and CO2 and (b) SO2, S_2, COS, CO and H_2 molar flow changes along first 0.5 m of the furnace.

1.2. Lavan

The main goal of this research is to develop a mathematical model for an industrial Claus process and optimization the process performance to enhance Sulfur recovery. The conventional Claus process consists of thermal conversion in furnace and catalytic conversion in two series fixed bed rectors. The furnace and catalytic reactors are modeled based on the mass and energy conservation laws at steady state condition. To prove the accuracy of the developed mathematical model, the simulation results of the process are compared with the available plant data. Then, the optimal condition of Claus process is calculated considering Sulfur recovery as the objective function using Genetic algorithm as a useful method in global optimization. The attainable decision variables are inlet temperature of furnace and fixed bed reactors, feed distribution along the furnace and flow rate of air in the furnace.

Per the authors, the following main and side reactions take place in the thermal section:

$$H_2S \leftrightarrow H_2 + S_2 \qquad (1)$$

$$H_2S + 1.5O_2 \rightarrow SO_2 + H_2O \qquad (2)$$

$$H_2S + 0.5SO_2 \leftrightarrow H_2O + 0.75S_2 \qquad (3)$$

$$CH_4 + 1.5O_2 \rightarrow CO + 2H_2O \qquad (4)$$

$$C_2H_6 + 2.5O_2 \rightarrow 2CO + 3H_2O \qquad (5)$$

$$C_3H_8 + 3.5O_2 \leftrightarrow 3CO + 4H_2O \qquad (6)$$

$$C_4H_{10} + 4.5O_2 \rightarrow 4CO + 5H_2O \qquad (7)$$

$$CO + 0.5S_2 \leftrightarrow COS \qquad (8)$$

$$CO_2 + H_2 \leftrightarrow CO + H_2O \qquad (9)$$

$$CO + H_2S \leftrightarrow COS + H_2 \qquad (10)$$

$$NH_3 \rightarrow 0.5N_2 + 1.5H_2 \qquad (11)$$

$$CH_4 + 2S_2 \rightarrow CS_2 + 2H_2S \qquad (12)$$

In this study, a one-dimensional model is developed to simulate the thermal and catalytic sections of Claus process. In the considered mathematical model, the following assumptions are adopted:

- Plug flow pattern is considered.
- Radial diffusion of mass and energy is negligible.
- The system operates at steady-state condition.
- The gas mixture is an ideal gas.
- The catalytic reactions take place on the catalyst surface.

To develop the mass and energy balance equations, a differential element along the axial direction of the reactor is considered and governing equations are written over the element. The mass and energy balance equations in the furnace are as usual:

$$-\frac{F}{A}\frac{dy_i}{dz}+\sum_{i=1}^{N} r_i = 0$$

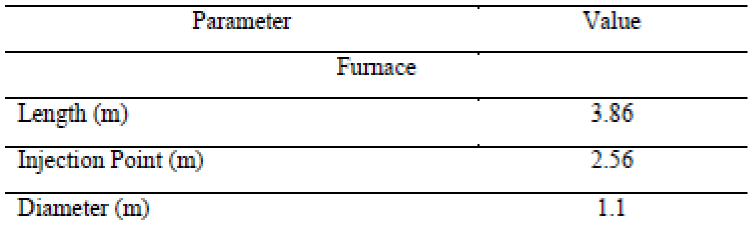
$$-\frac{F}{A}C_P\frac{dT}{dz}+\sum_{i=1}^{N} r_i(-\Delta h_i^{0,f}) = 0$$

This set of equations is solved with 4th order Runge-Kutta method.

The optimal condition of Claus process will be computed considering Sulfur recovery as the objective function while using Genetic algorithm (evolution of the population of candidate solutions to the problem). The attainable decision variables are inlet temperature of furnace and reactors, feed distribution and flow rate of air in the furnace. To ensure that the feed temperature at the reactor inlet is not too low for the hydrolysis and Claus reaction to occur, the lower bound on temperature of feed is set at 150°C.

The reliability and accuracy of the developed mathematical model of Claus process is then validated by comparing the simulation results with the available plant data in Lavan Refinery Complex in Iran. As represented in Table 5, the simulation results have a good agreement with the available plant data at the same process condition.

Conventionally, hydrogen sulfide conversion occurs in the adiabatic catalytic reactors in Claus process. In the thermal section of Claus process, hydrogen sulfide could be decomposed to elemental Sulfur and hydrogen. On the other hand, in presence of oxygen, hydrogen sulfide is converted to water vapor and Sulfur dioxide. In addition, hydrogen sulfide could react with carbon dioxide and hydrocarbons to form CS_2 and COS, respectively. In the catalytic section, the hydrogen sulfide and Sulfur dioxide reacts and elemental Sulfur is produce. Table 5 presents the value of considered decision variables at optimized and conventional condition. It is shown that, whereas a part of the feed stream is injected intto the middle part of furnace in the conventional configuration; the total feed stream is injected into the furnace inlet in the optimized system. The presented data show that injection a part of feed stream at middle part of furnace has a negative effect on the Sulfur recovery. Indeed, the simulation results prove the performance of Straight-Through Process over Split Flow Process in the Lavan Refinery



Parameter	Value
Furnace	
Length (m)	3.86
Injection Point (m)	2.56
Diameter (m)	1.1

Table 5: Lavan industrial furnace specifications

Reaction No.	Rate of reaction
1	$r_{H_2S}\left(\frac{mol}{m^3\,sec}\right) = 4.3\times10^6\exp\left(-\frac{26}{RT}\left(\frac{kcal}{mol}\right)\right)C_{H_2}C_{S_2} - 3.6 \times 10^8\exp\left(-\frac{48}{RT}\left(\frac{kcal}{mol}\right)\right)C_{H_2S}$
2	$r_{H_2S}\left(\frac{mol}{cm^3\,sec}\right) = 14\exp\left(-\frac{11}{RT}\left(\frac{kcal}{mol}\right)\right)P_{H_2S}P_{O_2}^{1.5}$
3	$r_{H_2S}\left(\frac{mol}{cm^3\,sec}\right) = 15900\exp\left(-\frac{49.9}{RT}\left(\frac{kcal}{mol}\right)\right)P_{H_2S}P_{SO_2}^{0.5} - 500\exp\left(-\frac{49.9}{RT}\left(\frac{kcal}{mol}\right)\right)P_{H_2O}P_{S_2}^{0.75}$
4	$r_{CH_4}\left(\frac{mol}{cm^3\,sec}\right) = 10^{13.2}\exp\left(-\frac{48.4}{RT}\left(\frac{kcal}{mol}\right)\right)C_{CH_4}^{0.7}C_{O_2}^{0.8}$
5	$r\left(\frac{mol}{\ }\right) = 1.3\times10^{12}\exp\left(-\frac{30}{\ }\left(\frac{kcal}{\ }\right)\right)C^{0.1}\ C^{1.65}$
8	$r_{CO}\left(\frac{kmol}{m^3\,sec}\right) = 3.78\times10^5\exp\left(-\frac{6700}{T}\right)C_{CO}C_{S_2} - 2.05 \times 10^9\exp\left(-\frac{21630}{T}\right)C_{COS}C_t$
9	$r_{CO_2}\left(\frac{kmol}{m^3\,sec}\right) = 3.95\times10^{10}\exp\left(-\frac{31220}{T}\right)C_{CO_2}C_{H_2}^{0.5}$
10	$r_{CO}\left(\frac{kmol}{m^3\,sec}\right) = 1.59\times10^5\exp\left(-\frac{13340}{T}\right)C_{CO}C_{H_2S}^{0.5}$
11	$r_{NH_3}\left(\frac{mol}{cm^3\,sec}\right) = 0.0042\exp\left(-\frac{16.5}{RT}\left(\frac{kcal}{mol}\right)\right)P_{NH_3}^{1.25}$
12	$r_{CH_4}\left(\frac{kmol}{m^3\,sec}\right) = 5.53\times10^{10}\exp\left(-\frac{19320}{T}\right)C_{CH_4}C_{S_2}$

Table 6: Rate of thermal reactions

unit. Generally, applying the obtained optimal configuration in place of the conventional Claus process increases the Sulfur recovery as much as 1.07%.

The conversion of H_2S in the optimized system increases about 1.62% compared to the conventional Claus process. It appears that about 70% hydrogen sulfide is converted to elemental Sulfur in the thermal section. Due to high temperature in the furnace, the reactions have a very fast kinetics and take place upon a short portion of the total volume..

In the furnace, hydrogen sulfide and oxygen reacts and Sulfur dioxide is produced. Thus, hydrogen sulfide concentration decreases (fig. 15), while Sulfur dioxide increases along the furnace length in the optimized system. Since a part of hydrogen sulfide is injected to the middle part of furnace, hydrogen sulfide is consumed in the first part of furnace, then increases at injection point and then decreases again, smoothly. The stoichiometry coefficient of reactants in the main catalytic reaction shows that the optimum value of H_2S to SO_2 is 2, based on the stoichiometric coefficient of reactants, which correspond to the obtained value for this ratio at the outlet of furnace reactor at the optimized condition.

Since the net reaction is exothermic, temperature increases along the furnace and the catalytic reactors. Decreasing temperature in the catalytic zone can shift equilibrium conversion in the main reaction toward higher H_2S and SO_2 conversion. On the other hand, it decreases rate of reaction and applying larger reactor to convert reactant is necessary. In addition, since hydrolysis reactions are non-reversible, decreasing temperature decreases rate of CS_2 and COS conversion. Thus, there is an optimal temperature profile to improve Sulfur recovery in the Claus process. Applying the obtained optimal temperature profile on the system increases Sulfur recovery significantly.

In a Claus Sulfur Recovery Unit, COS and CS_2 are formed in the acid gas burner and are partially destroyed in the first catalytic bed. It appears that COS concentration increases along the furnace length. In the catalytic section, the produced COS and CS_2 in the thermal section are hydrolyzed to the carbon dioxide and hydrogen sulfide. Applying the obtained optimal operating condition on the system, decreases COS production from 0.004 to 0.002: ne of the main problems in the conventional system is in fact COS production at the second half of furnace after injection point.

Figure 16 shows hydrogen profile along the conventional and optimized systems. Higher temperature in the furnace and lower oxygen injection at the optimal conditions yield a higher rate of hydrogen sulfide decomposition, so more hydrogen is produced. Thus, operating at higher temperature increases elemental Sulfur and hydrogen production from hydrogen sulfide decomposition.

	Simulation Results	Plant Data	Relative Error
	Thermal section		
Temperature (°C)	1266	1227	3.17
H_2S (kmol hr^{-1})	7.64	7.664	0.31
SO_2 (kmol hr^{-1})	3.32	3.308	0.36
S_2 (kmol hr^{-1})	11.7	11.697	0.02
H_2 (kmol hr^{-1})	2.033	2.06	1.31
CO (kmol hr^{-1})	0.0731	0.075	2.53
COS (kmol hr^{-1})	0.0228	0.023	0.87

Table 7: Comparison between simulation results and plant data

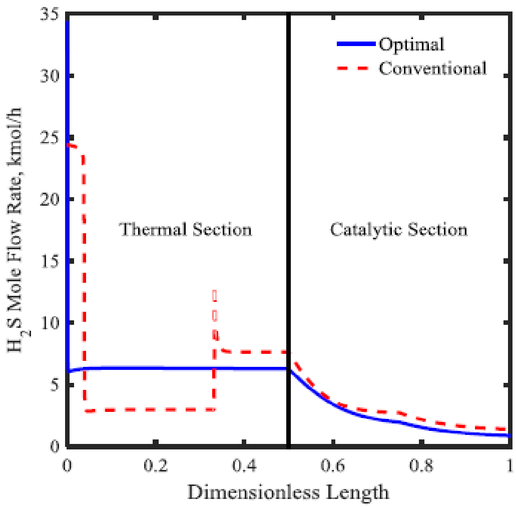


Figure 15: Lavan H_2S profile

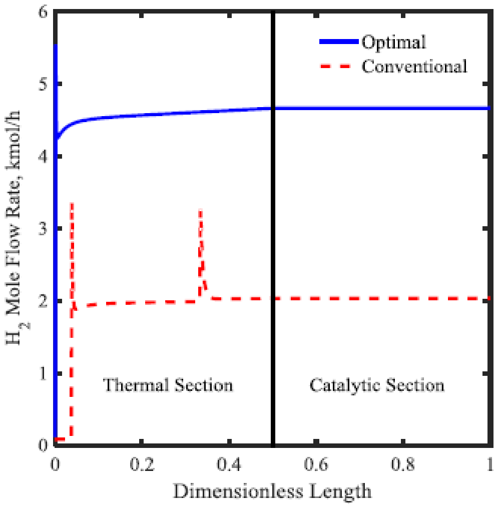
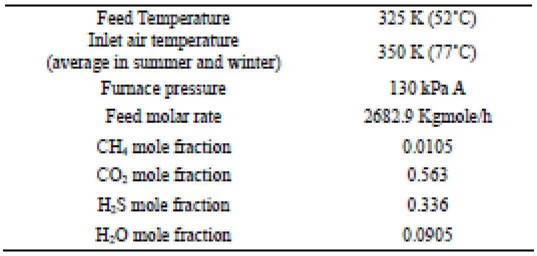


Figure 16: Lavan H_2 profile

1.3. Khangiran

The objective of this paper is to model the main reactions that take place in the Claus reactor furnace and compare it with actual data and simulated process. Since the most important point is the selection of suitable reaction conditions to increase the reactor performance, the model is formulated to predict the performance of the Claus plant. To substantiate the theoretical model, the authors used actual process condition and feed composition in Khangiran Gas Refinery. Model equations have been solved by using MATLAB program. Results from MATLAB are compared with actual plant data. The AAD (Average Absolute Deviation) of modeling results with actual data is 2.07%: error values are very little and show accuracy and precision of modeling and simulation. The predicting curve for different parameters of the reactor furnace according to variable conditions and specifications are then given.



Feed Temperature	325 K (52°C)
Inlet air temperature (average in summer and winter)	350 K (77°C)
Furnace pressure	130 kPa A
Feed molar rate	2682.9 Kgmole/h
CH_4 mole fraction	0.0105
CO_2 mole fraction	0.563
H_2S mole fraction	0.336
H_2O mole fraction	0.0905

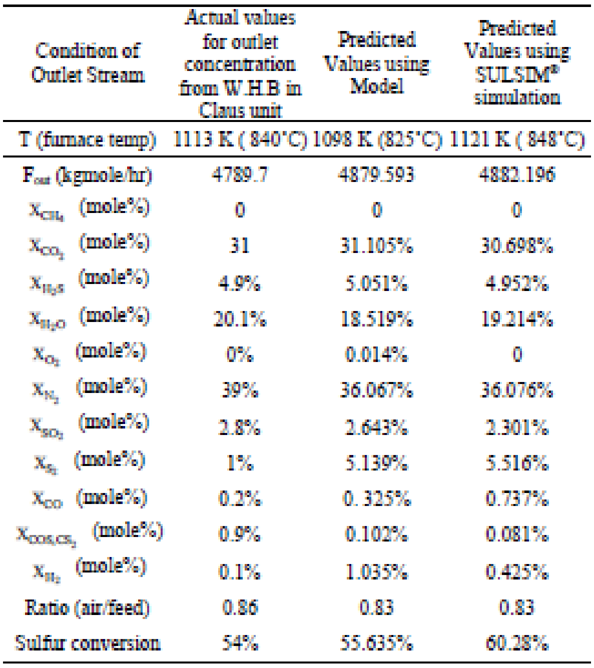
Table 8: Khangiran SRU specifications

The basic structure of the model consist of the equations of mole and energy conservative rule the furnace, which are related to each other and are function of molar conversion of H_2S in equilibrium reaction and temperature. In order to model the reactor, a steady-state simulation has been used for mole and energy balance. Sames and Paskal presented empirical correlations to predict the fraction of H_2, CO, COS, CS_2 and Sulfur (as S) in the effluent of the Claus furnace. The correlations were obtained from more than 300 tests on 100 different Sulfur trains; with different flow configurations processing acid gas feed streams [12]. The authors use these equations to model the furnace and mole balance. In this work, furnace pressure is 130 kPa (absolute) and pressure drop (ΔP) is 10 kPa. Using empirical equations and applying in the mole balance for the compounds, we get the mole balance equations, for each compound.

In order to verify the model, we compare the output result from reaction furnace, model values and simulation results obtained with industrial software SULSIM. Furnace temperature from the model (1098 K) is lower than actual temperature of reaction furnace (1113 K); 15°C temperature difference is negligible and it results into 1.35% error. Simulated results shows that predicted temperature is 1121 K and higher than the real Claus furnace temperature. Error occurred using software is lower (0.72%). In this case simulation is more reliable. Additionally, Sulfur conversion obtained from model results (56.635%) is in good agreement with conversion of Sulfur in Claus plant (54%) in gas refinery. On the other hand, results from simulation indicate that Sulfur conversion is 60.28%.

As it is presented in Table 9, in the furnace effluent there is Sulfur vapor: it is due to the high temperature in the furnace. Since Sulfur production is high compared with actual plant; air consumption is low,

O_2 in Claus plant damages some process fundamental equipments (catalyst exchanger) and thus must be minimized. Predicted concentration is 0.014% and error value is acceptable.



Condition of Outlet Stream	Actual values for outlet concentration from W.H.B in Claus unit	Predicted Values using Model	Predicted Values using SULSIM® simulation
T (furnace temp)	1113 K (840°C)	1098 K (825°C)	1121 K (848°C)
F_{out} (kgmole/hr)	4789.7	4879.593	4882.196
X_{CH_4} (mole%)	0	0	0
X_{CO_2} (mole%)	31	31.105%	30.698%
X_{H_2S} (mole%)	4.9%	5.051%	4.952%
X_{H_2O} (mole%)	20.1%	18.519%	19.214%
X_{O_2} (mole%)	0%	0.014%	0
X_{N_2} (mole%)	39%	36.067%	36.076%
X_{SO_2} (mole%)	2.8%	2.643%	2.301%
X_{S_2} (mole%)	1%	5.139%	5.516%
X_{CO} (mole%)	0.2%	0. 325%	0.737%
X_{COS,CS_2} (mole%)	0.9%	0.102%	0.081%
X_{H_2} (mole%)	0.1%	1.035%	0.425%
Ratio (air/feed)	0.86	0.83	0.83
Sulfur conversion	54%	55.635%	60.28%

Table 9: Comparison between Khangiran plant data with model and simulation results

COS and CS_2 actual values are greater than the ones obtained using the developed equilibrium model and the simulation software. COS forms in the WHB (Sames, 1990) following the reaction:

$$CO + \frac{1}{2} S_2 \rightarrow COS$$

Predicted S_2 content in both methods is greater than plant data. There is a big difference between real and predicted values for S_2; that may be due to the formation of liquid Sulfur in WHB. According to Table 9 and comparison between results and plant data, and also neglecting the error in CO and H_2 predicted concentrations, average error is about 3.5% and 5.36% for model and SULSIM® simulation; also AAD (Average Absolute Deviation) in comparing actual data with modeling and simulation results are 2.07% and 4.92%, respectively. We can conclude that the model is more efficient and applicable for other Claus plants with different inlet composition.

Figure 17 shows reaction furnace predicted temperature vs inlet H_2S content using model and simulation software. Both simulation and model have similar trend. According to the figure, model predicts that 1% increase in H_2S content will result into 7.5°C increase in furnace temperature. In order to combust hydrocarbons and aromatics, furnace temperature must be 1050°C. According to the model, if inlet gas stream contains more than 26% H_2S in current plant, temperature will rise higher than 1050°C (1323 K). Figures 18 and 19 illustrate the predictions of model and simulation for H_2S and Sulfur conversion in furnace again vs increase in H_2S content in feed. As model predicts, for one percent of the mole fraction of H_2S in feed stream, Sulfur conversion increases by 0.54% in reaction furnace and S2 mole fraction in outlet gas stream increases 0.12%.

Figure 18 indicates that lower H_2S concentrations in the feed correspond to lower furnace temperatures; while higher concentrations bring higher furnace that promote H_2S cracking and conversion in the unit.

The effect of preheating on the furnace can be predicted in the same way. Furnace temperature increases 4.4°C following 10°C increase in inlet temperature. H_2S conversion increases by 0.156% when inlet temperature increases 10°C. Also H_2S conversion in furnace rises from 72.66% to

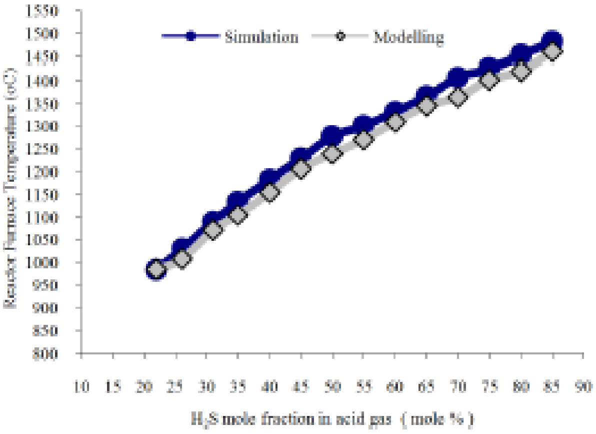


Figure 17: Simulation and model estimation for Modified Claus furnace temperature vs. H_2S content in the feed.

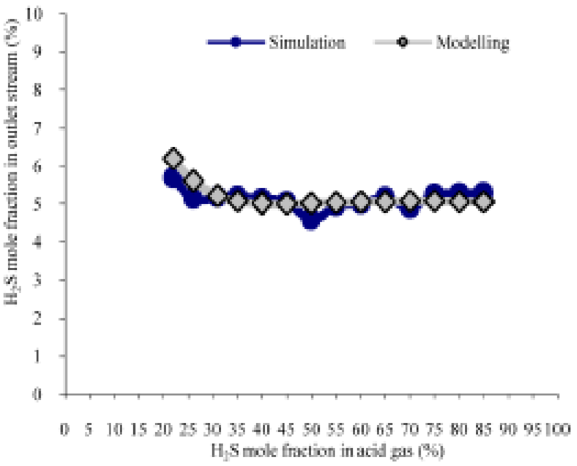


Figure 18: Simulation and model estimation for H_2S % at the outlet vs. H_2S content in the feed.

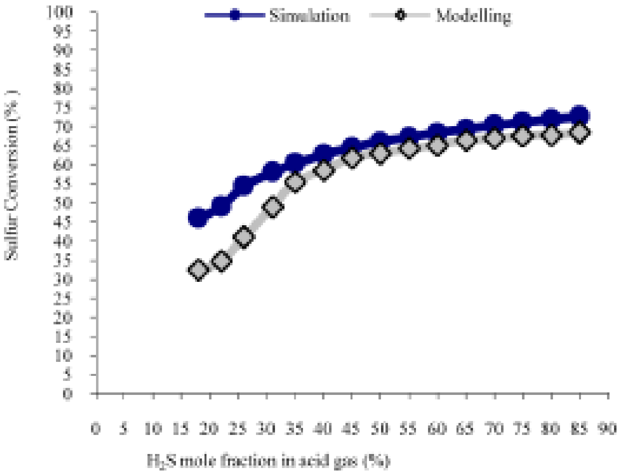


Figure 19: Simulation and model estimation for Sulfur conversion (%) at the outlet vs. H_2S content in the feed.

75.74% with preheated feed (252°C). Also 10°C increase in temperature of inlet air results into 2.52°C increases in furnace temperature, 0.103% increase in H_2S conversion and 0.153% increase in Sulfur conversion. If both acid gas feed and inlet air preheated separately and equally 10°C, reaction furnace temperature increases 7.1°C, H_2S conversion 0.21% and Sulfur conversion 0.31%.

From a theoretical point of view, there is an optimal temperature in the furnace reactor to get the more efficient performance, maximizes Sulfur production and H_2S conversion as reported in the previous sections for Claus process. A solution for this problem is fuel gas injection in the furnace to increase the flame temperature Calculation showed that 2000 m^3/h fuel injection in acid gas feed (50000 m^3/h) cause furnace temperature to increase of about 130°C. More hydrocarbon content in the feed will nevertheless produce more CS_2 and COS in the furnace. It was also observed that fuel injection led to reduction of the plant productivity as acid gas stream is diluted.

2. Simulation Tools

2.1. STRESS software

STRESS® (Sulphur Thermal Reactor Simulation Software), is the computational tool used to simulate the thermal section of a Claus process, exploiting, as a basis, the DSMOKE software, an earlier version of OpenSMOKE++, presented in the next chapter.

It consists of a graphical user-friendly interface and of a code portion that launches a DOS window for the real numerical simulation. Code includes a chemical interpreter, that allows to read and acquire the kinetic mechanism, thermodynamic properties of the various species involved and the kinetic model of the reactor.

This software requires, as input, in the GUI, the following specifications:

- Molar of mass flow rate of the acid gas to be treated;
- Volumetric/molar composition, i.e. volumetric or molar fractions, of the inlet acid gas;
- Molar or mass flow rate of combustion air;
- Volumetric/molar composition, i.e. volumetric or molar fractions, of the combustion air;
- Heat converter type (“One zone” or “Split Flow”);
- Thermal converter and furnace characteristics dimensions.

2.2. The OpenSMOKE++ Suite

OpenSMOKE++ is a general framework for numerical simulations of reacting systems with detailed kinetic mechanisms, including thousands of chemical species and reactions. The framework is entirely written in object-oriented C++ and can be easily extended and customized by the user for specific systems, without having to modify the core functionality of the program. The OpenSMOKE++ framework can handle simulations of ideal chemical reactors (plug-flow, batch, and jet-stirred reactors), shock-tubes, rapid compression machines, and can be easily incorporated into multi-dimensional CFD codes for the modeling of reacting flows. OpenSMOKE++ provides useful numerical tools such as the sensitivity and rate of production analyses, needed to recognize the main chemical paths and to interpret the numerical results from a kinetic point of view. Since simulations involving large kinetic mechanisms are very time consuming, OpenSMOKE++ adopts advanced numerical techniques able to reduce the computational cost, without sacrificing the accuracy and the robustness of the calculations. The OpenSMOKE++ Suite is a collection of standard solvers for performing kinetic analyses with detailed kinetic mechanisms, with hundreds of species and thousands of reactions. The word solver has to be intended as an independent program, built with the aim to perform a specific task (for example to simulate a batch reactor, or to model a shock-wave, etc.). Thus, in the following, the OpenSMOKE++ Suite definition will be used to refer to the collection of OpenSMOKE++ standard solvers. The list of available solvers includes:

1) a kinetic pre-processor, a utility which is able to read, pre-process and analyze kinetic mechanisms written in the CHEMKIN format. Its main purpose is to rewrite the kinetic scheme in a XML format which can be efficiently used by the OpenSMOKE++ Suite solvers;

2) a collection of solvers (i.e. independent executable files), one for each system to simulate. In other words, the OpenSMOKE++ Suite provides the solver dedicated to the simulation of plug _ow reactors, the solver dedicated to the simulation of batch reactors, and so on. These solvers are completely independent from each other, but need the same pre-processed kinetic mechanism in XML format generated by the kinetic pre-processor described above;

3) a graphical post-processor, to easily post-process the simulation results. It is able not only to plot the usual profiles of temperature, pressure, composition, etc. along the time or space coordinate, but it is extremely useful to rapidly perform sensitivity analyses, rate of production analyses, and to draw flux diagrams. The figure below show a schematic diagram of the OpenSMOKE++ Suite.

2.2.1. How the OpenSMOKE++ Suite works

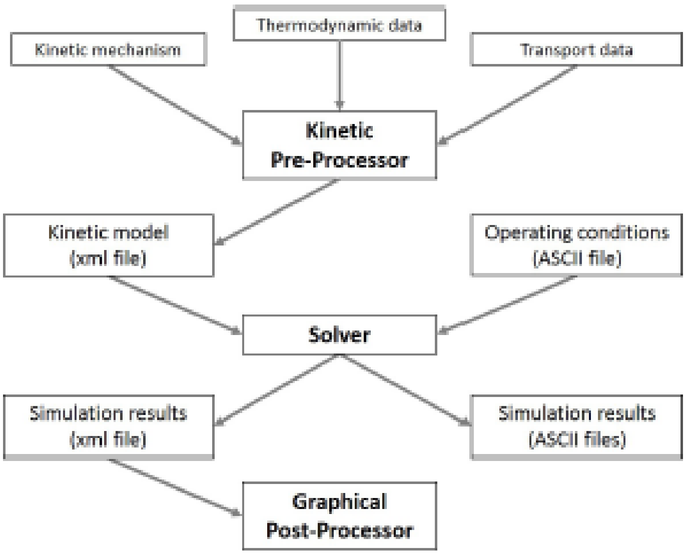


Figure 20: The OpenSMOKE++ Suite structure.

All the OpenSMOKE++ Suite solvers, together with the kinetic pre-processor, do not have any graphical interface. They are based on input _les in ASCII format. Only the graphical post-processor works with a GUI (Graphical User Interface), based on the QT libraries. This means that if you want to pre-process a kinetic mechanism (together with the thermodynamic and transport properties) or you want to simulate a reactor, your input conditions must be provided through one or more input files. These are simple ASCII files which can be written and modified using a generic text editor, like Notepad, Notepad++, Microsoft Word, gedit, etc.

Before using it in OpenSMOKE++ Suite, the kinetic scheme has be be pre-processed and re-written in a new format. In order to preprocess a kinetic scheme, the OpenSMOKE++ Suite provides the

OpenSMOKEpp_CHEMKIN_PreProcessor utility. The user has to supply the _les containing the thermodynamic data, the kinetic mechanism, and (optionally) the transport data. For the simulation of ideal reactors, usually the transport data are not needed and therefore the user could choose to preprocess only the thermodynamic and the kinetic data. This is useful, since in many cases the transport data are not available. In order to run the OpenSMOKEpp_CHEMKIN_PreProcessor utility, the user has to write an input _le containing the instructions (i.e. the dictionary) to perform the preprocessing and/or additional useful operations (checking of the thermodynamic properties, post-processing of reaction rates, etc.).

The ideal reactor simulations can be performed using the same approach used for pre-processing the kinetic mechanism, as described in Chapter 4. Thus, the user has to supply a proper dictionary containing the instructions and the options for running the simulation under investigation. As usual, the solver can be run using instructions from the command line.

2.3. Aspen Hysys

HYSYS is an interactive process engineering and simulation program. It is a powerful software for simulation of chemical plants and oil refineries. It includes tools for estimation of physical properties and liquid-vapor phase equilibrium, heat and material balances, as well as simulation of many types of chemical engineering equipment.

As a user-friendly computer software package, developed by Hyprotech, it combines comprehensive data regression, thermodynamic database access (TRC, DIPPR, DDB, API, PDS) and the Mayflower distillation technology to enable the design and analysis of separation systems, including azeotropic and extractive distillation and non-ideal, heterogeneous and multiple liquid phase systems.

Other than being widely popular in academic and industrial contexts, Hysys turns out to be very useful in the proceedings of this work, as it not only allows to quickly simulate different process configurations and equipment performances but also to create links and pass data to different simulation environments (like Matlab), more suited to represent complex reactor systems. Hysys itself, in fact, does not provide the fundamental kinetic mechanisms necessary to obtain reliable results in terms, for example, of reactant conversions and temperature profiles.

3. Claus Thermal Reactor Model

The furnace, as well as the RTR, were implemented in OpenSMOKE++ by means of two Plug Flow Reactors in series, the first one representing the combustion chamber and the second one representing the Waste Heat Boiler (WHB). The equation of the model are largely simplified as neither the laminar flame or the diffusive phenomena inside the units are considered (although they may eventually be subject of a dedicated study). The massive material balance fed to the OpenSMOKE++ solver is therefore reduced to the following form:

$$\frac{d\omega_i}{d\tau} = \sum_{j=1}^{NR} \nu_{ij} r_j MW_i \quad i = 1, \dots NC$$

where ω_i stands for the massive fraction of the i-th component; τ for the residence time in the reactor [s]; ν_{ij} for the stoichiometric coefficient of the the i-th component in the j-th reaction; r_j for the reaction rate of the j-th reaction [kmol/m^3/s] and MW_i for the molecular weight of the i-th component. The same logic can be adopted to derive the energy balance for the system in exam:

$$C_P \frac{dT}{d\tau} = \sum_{j=1}^{NR} \left(-\Delta H_{R_j}^0\right) r_j + \frac{UA}{V}(T_{EXT} - T) \quad i = 1, \dots NC$$

where C_P stands for the heat capacity of the reacting mixture [kcal/kmol/K]; T for the system temperature [K]; τ for the residence time in the reactor [s]; $-\Delta H^0_{Rj}$ for the heat released as enthalpy change due to the whole set of reactions [kcal/kmol]; r_j for the reaction rate of the j-th reaction [kmol/m^3/s]; U for the global thermal exchange coefficient [kW/m^2/K]; A for the available exchange area [m^2]; V for the volume of the system [m^3] and T_{EXT} for the temperature of the external medium [K] (e.g. the cooling fluid used in the WHB).

4. Validation Results

4.1. Shiraz: Data Validation

WHB specifications were not reported and, as a matter of fact, were not necessary to compare the obtained data since the paper showed the results for the outlet section of the furnace. Arbitrary but plausible values have then been set. For heat loss and thermal coefficient estimation,, which on the contrary are not negligible and affect the outcome of the simulation, plausible data have been computed through similitude with other units of the same kind. STRESS software yields good results for temperature and for many species; however, CO and COS molar fractions are way less than what we should expect. This may be due to the fact that STRESS software adopts an equilibrium-based model which is not able to simulate the behavior of these two compounds, probably governed by kinetics as the paper suggests, but the reason could also be an error in the data since such high values at the furnace outlet (especially regarding COS) are hardly compatible with industrial practice. The very high SO_2 fraction reported in the paper also seems unlikely from a TD point of view, as it is not consistent with the equilibrium combustion (flame) obtained with this set of initial data. S_2 at the outlet is also too much: STRESS does not include association reactions of elementary Sulfur to its heavier forms that take place in the WHB, where temperature decreases and favors condensation; furthermore when acid gas feed is higher than air feed it is not able to simulate well its recombination reactions to give H_2S: as a result S_2 molar fraction continues to increase.

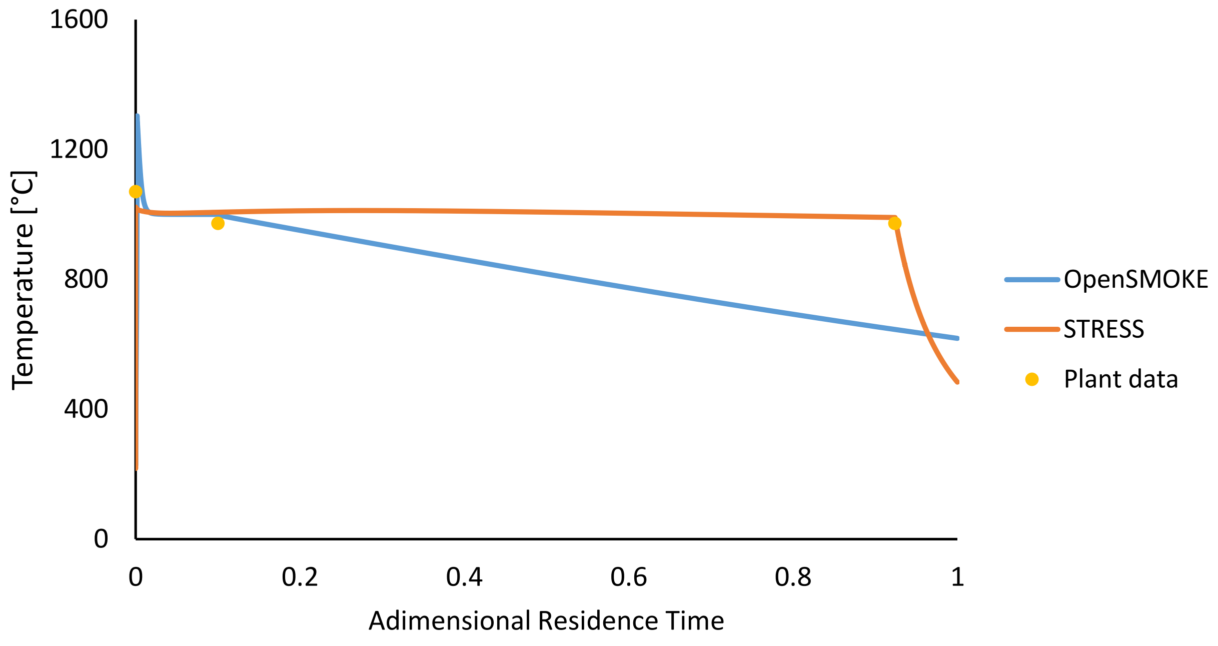


Figure 21: Shiraz Furnace: STRESS and OpenSMOKE++ temperature profiles

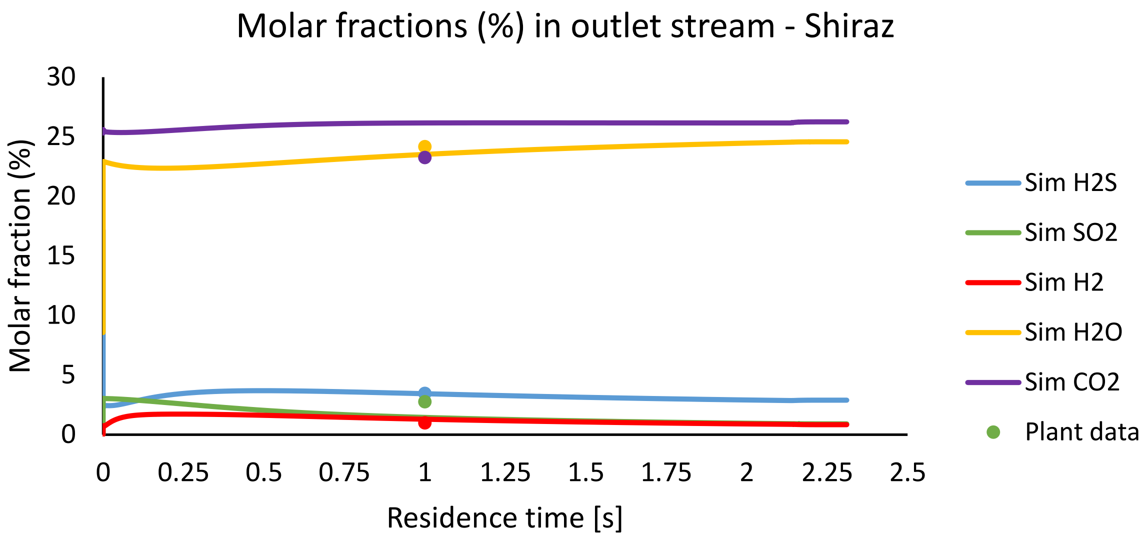


Figure 22: Shiraz Furnace: STRESS main species profiles

OpenSMOKE++ simulation originates slightly worse, but nonetheless acceptable, results. In particular the implemented model shows weaknesses when it comes to thermodynamic equilibrium calculations as the flame section of the combustion chamber was not explicitly considered: the TD counterpart of the kinetic scheme is not yet available. A workaround was found setting an environmental temperature in the range of 1100-1300 K and a very high thermal exchange coefficient. In this way, a higher maximum temperature is obtained together with realistic outlet stream composition.

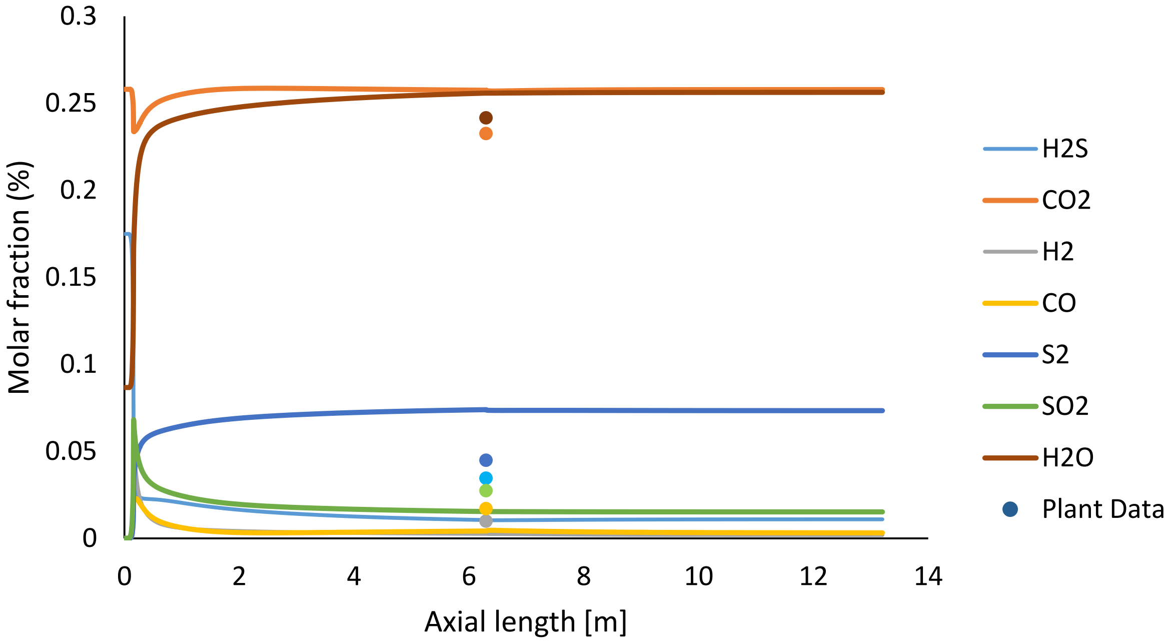


Figure 23: Shiraz Furnace: OpenSMOKE++ main species profiles

The comparison between the two simulation tools has been carried out by means of a peculiar accuracy measure, the SMAPE (Symmetric Mean Absolute Percentage Error) in place of the usual MAPE (Mean Absolute Percentage Error). The latter turns out to be misleading in the case of small denominators or negative error (actual value A_t smaller than forecast value F_t). SMAPE overcomes this problem proposing a different definition:

$$SMAPE = \frac{100}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{|A_t| + |F_t|}$$

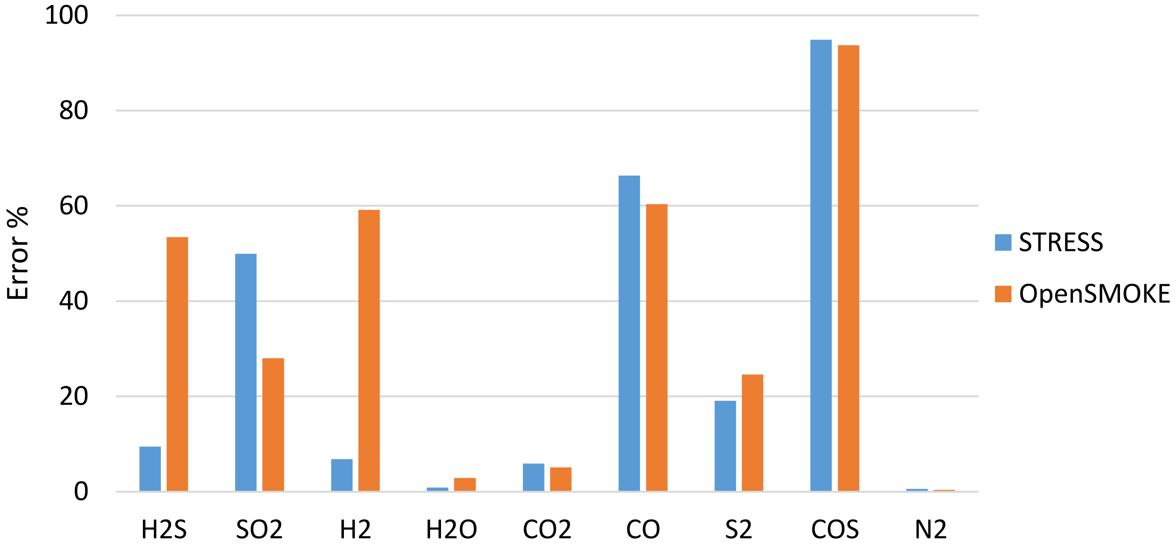


Figure 24: Shiraz Furnace: STRESS and OpenSMOKE++ sMAPE

It is quite apparent just looking at Fig. 24 that STRESS yields better results compared to OpenSMOKE++, and this is confirmed by a SMAPE value of 28.2% against 36.4%.

Overall, both simulation yield acceptable results and different applications for this industrial stream of acid gases will be further investigated. Finally, the table below reports outlet stream conditions (composition and temperature) used for comparison (Plant vs STRESS vs OpenSMOKE++).

	PLANT DATA	STRESS	OpenSMOKE++
Pressure [atm]	1.69	1.69	1.69
Furnace Outlet Temp [°C]	972	990	1000
H_2S (%)	3.4512	2.8562	1.048
SO_2 (%)	2.7345	0.9124	1.537
H_2 (%)	0.9836	08582	0.2523
H_2O (%)	24.1414	24.5370	25.5796
CO_2 (%)	23.2449	26.1516	25.7414
CO (%)	1.6947	0.3424	0.4191
S_2 (%)	4.4742	6.5835	7.3915
COS (%)	1.8296	0.048	0.05936
CS_2 (%)	-------	--------	0.00214
N_2 (%)	37.2688	37.7107	37.7420

Table 10: Shiraz Furnace: outlet streams comparison

4.2. Lavan: Data Validation

Lavan refinery SRU operates a split-flow furnace, where a part of the acid gas feed is injected in the middle of the combustion chamber after the burner flame, in order to better prevent complete oxidation of H_2S. In agreement with the authors' choice, a simple straight-through configuration was adopted, since this allows to get a higher Sulfur recovery (different process simulations confirm this assumption). The inlet acid gas streams were re-united in a single one; its temperature is a weighted average of the two previous ones. As seen before, WHB specifications and furnace heat loss flux were adjusted on the basis of similar industrial data in order to better represent the real plant operation. The agreement between the two sets of data is good, except for CO and COS; a plausible reason has already been descripted. Concerning temperature, the relative error committed is definitely acceptable; the flame temperature is really off but this could very well be an error in the paper, as a value like this one is hardly consistent with the thermodynamic equilibrium for the system in exam (which is well represented by STRESS).

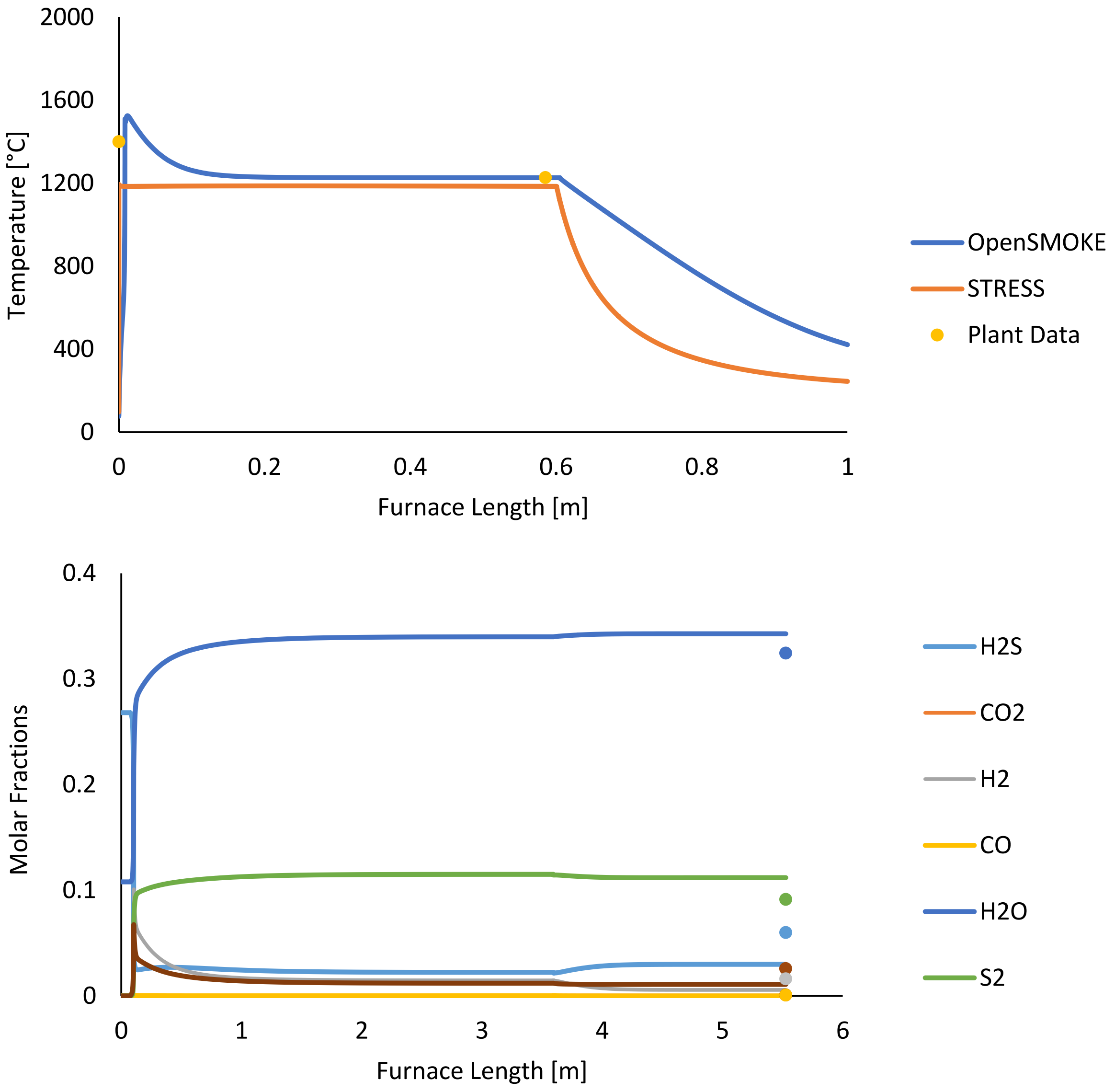


Figure 25 and 26: Lavan Furnace: STRESS and OpenSMOKE++ temperature profiles; STRESS composition profiles

Again, OpenSMOKE++ performs generally worse but still in an acceptable manner. CO and COS still represent the most critical component from a simulation point of view, because of the complex kinetics in the case of the second one and of the very low concentration of CO_2 in the inlet stream as for the first one. The main synthesis route towards CO is, in fact, the reduction of carbon dioxide through reaction with H_2S so here a reactant is practically missing. The value reported in the paper and related to the industrial plant seems then definitely too high.

One more time, the model forecasts fewer hydrogen than the actual one. This means that oxidation of H_2S tends to go towards completion and is not partial, if we exclude the very first zone of the combustion chamber, occupied by the flame.

SMAPE accuracy indicator reports a value of 26.4% for the simulation carried out with STRESS and 40% for the OpenSMOKE++ one. Those result are comparable with the ones obtained in the chaper before and so they confirm the goodness of our kinetic model. Anyway the incredibly high H_2S/CO_2 ratio at the inlet does not suit further investigation directed towards an AG2S™ application: at best only H_2 will be the valuable product to come out from the process and the optimization will lead to different specifications with respect to the one needed to produce syngas.

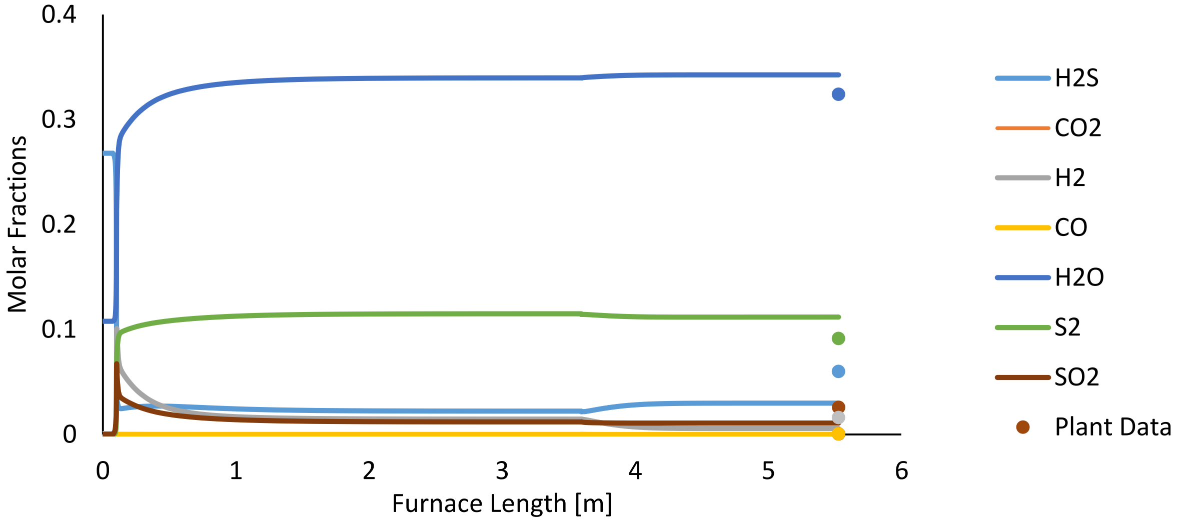


Figure 27: Lavan Furnace: OpenSMOKE++ main species profiles

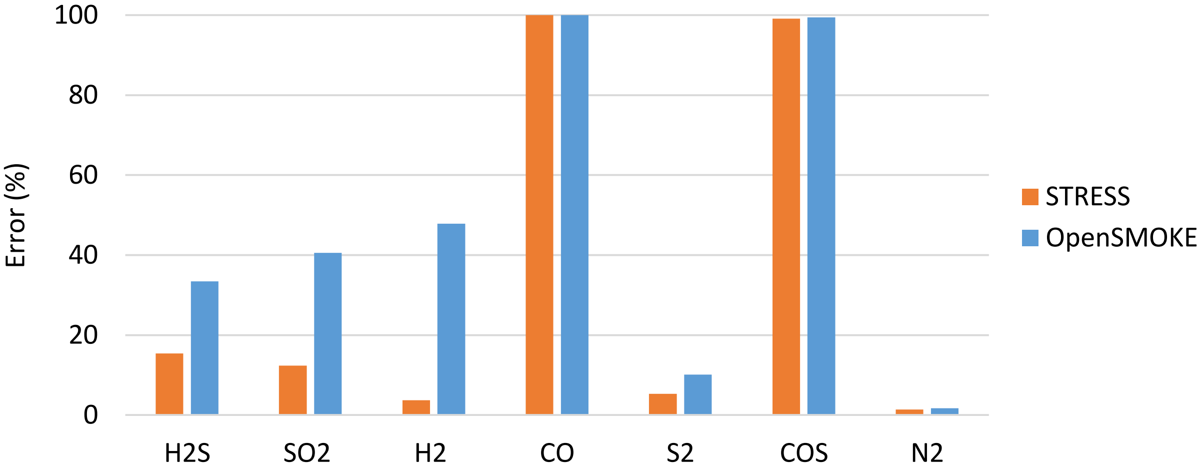


Figure 28: Lavan Furnace: STRESS and OpenSMOKE++ sAPE

	PLANT DATA	STRESS	OpenSMOKE++
Pressure [atm]	1.48	1.48	1.48
Furnace Outlet Temp [°C]	1227	1185	1000
H_2S (%)	5.9851	4.3887	2.7862
SO_2 (%)	2.5834	2.0140	0.9269
H_2 (%)	1.6087	1.4950	0.3908
H_2O (%)	-------	32.4168	34.6737
CO_2 (%)	-------	0.0759	0.0459
CO (%)	0.05857	0	8E-09
S_2 (%)	9.1347	10.1605	11.4352
COS (%)	0.01796	0.0001	0.000214
CS_2 (%)	-------	--------	1.6E-11
N_2 (%)	48.1188	49.4491	49.7021

Table 11: Lavan Furnace: outlet streams comparison

4.3. Khangiran: Data Validation

The reported values refer only to the composition of the inlet acid gas stream, its rate and its temperature. Composition and flow rate of the air stream was not indicated, but the authors found a 0.86 air/feed ratio the exit of the WHB; coupling this data with the total molar flux at the outlet and using the reported fraction of N_2, which is inert, a value for the inlet air flux has been calculated (x_{N2}=0.79 and x_{O2}=0.19 was the considered air composition). Furnace and WHB dimensions were assumed, scaling down other specifications found in literature on the base of the respective molar fluxes. In this case OpenSMOKE++ performances are better than the ones observed using STRESS software. That is because the simple, 2-PFR model implemented in OpenSMOKE++ with the aid of Matlab is able to forecast the thermal behavior of the combustion chamber, bringing its outlet temperature very close to the real plant one.

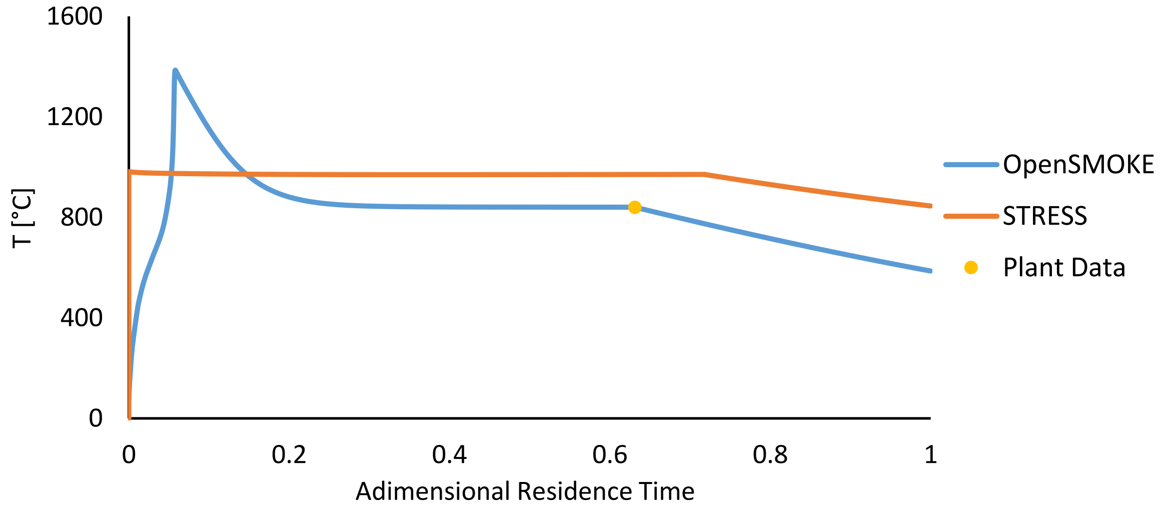


Figure 29: Khangiran Furnace: STRESS and OpenSMOKE++ temperature profiles;

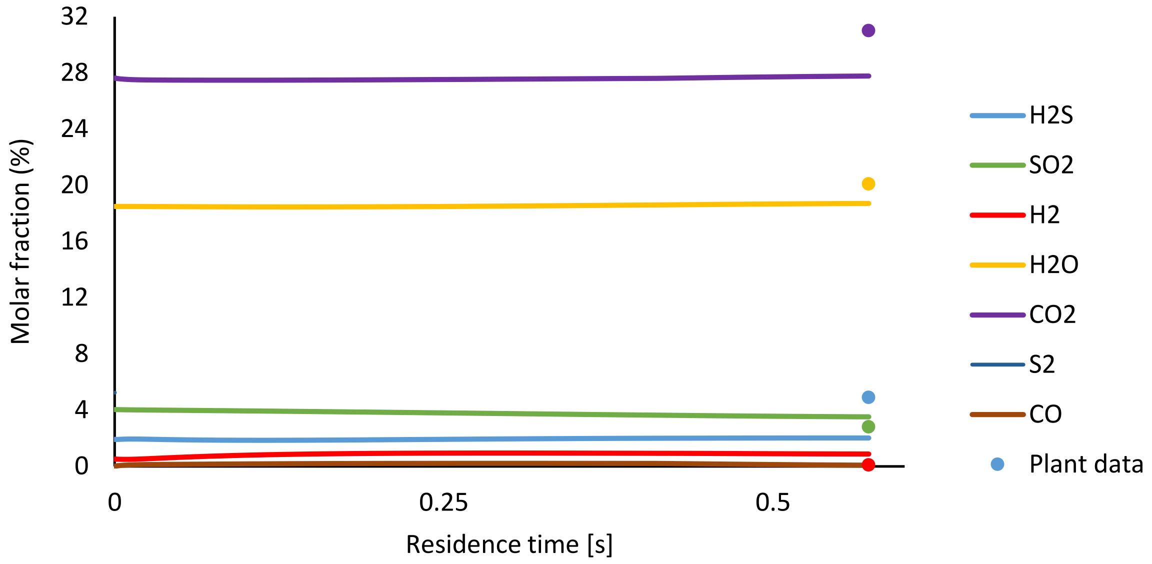


Figure 30: Khangiran Furnace: STRESS main species profiles

OpenSMOKE++'s great accuracy in reproducing the temperature profile of the furnace (except for the flame zone, as a thermodynamic-based mechanism was not implemented) is reflected in the various species profiles along the whole SRU. COS behavior, for example, gets much closer to the real one; the other results are still pretty much comparable.

SMAPE value for STRESS simulation is 39.1% while OpenSMOKE++ yields a preferable 37.8%; both simulation tools' performances are then acceptable (although slightly worse than in the two previous case studies) confirming the applicability of our kinetic scheme to real industrial SRU units and making this acid gas feed stream available for further investigation.

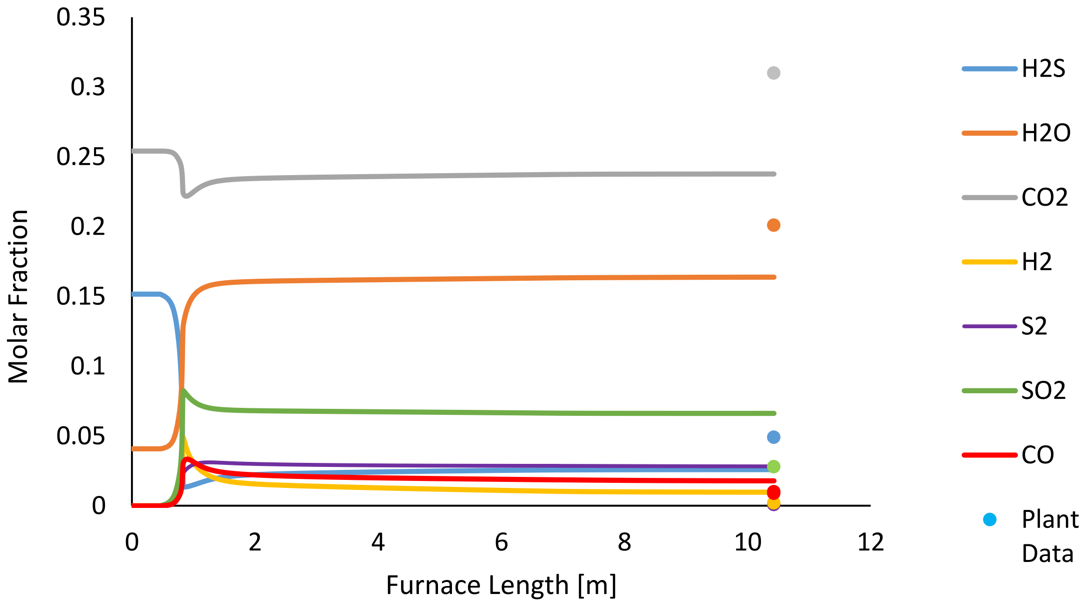


Figure 31: Khangiran Furnace: OpenSMOKE++ main species profiles

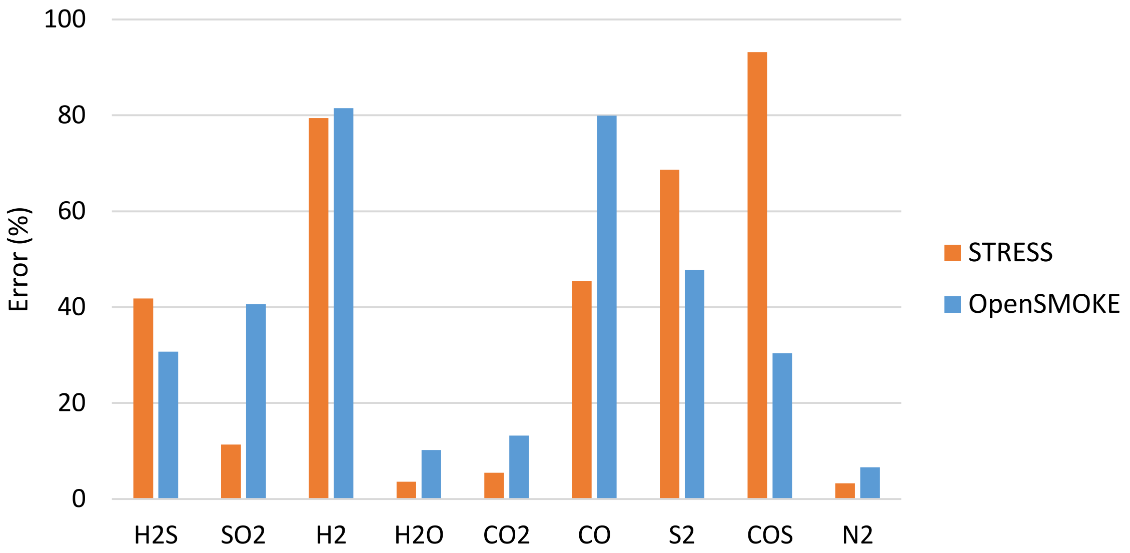


Figure 32: Khangiran Furnace: STRESS and OpenSMOKE++ sAPE

Finally, the Table 12 reports the results of the two simulations alongside the one referred to the real plant found in the paper. A critique can be moved towards this last set of data especially concerning the Sulfur (S_2) concentration in the outlet stream: being this specie a product of the process, a value of 1% seems definitely too low and definitely not realistic.

	PLANT DATA	STRESS	OpenSMOKE++
Pressure [atm]	1.283	1.283	1.283
Furnace Outlet Temp [°C]	840	971	840
H_2S (%)	4.9	2.0117	2.5975
SO_2 (%)	2.8	3.5171	6.6253
H_2 (%)	0.1	0.8709	0.9784
H_2O (%)	20.1	18.7041	16.3853
CO_2 (%)	31	27.7724	23.7751
CO (%)	0.2	0.0751	1.7922
S_2 (%)	1	5.3833	2.8296
COS (%)	0.9	0.0319	0.4805
CS_2 (%)	-------	--------	0.0016
N_2 (%)	39	41.6336	44.5061

Table 12: Khangiran Furnace: outlet streams comparison

PART 2 – CLAUS vs AG2S™ COMPARISON

The aim of the second part to this work is to properly set up and run a model which is able to reproduce the proposed configuration for the thermal section of the AG2S™ process; i.e. a furnace vary similar to the Claus one preceded by an heat exchanger having a pre-heating function. In this way not only flame temperature increases, but so does also H_2S reducing power towards CO_2, hopefully generating a better Sulfur recovery coupled with a greater selectivity directed to syngas production.

The preheater, for economical and process reasons, operates the thermal exchange between two process fluids; one coming out from the WHB and the other one being the fresh feed for the furnace (i.e. the acid gas stream). What can be accomplished is also then the cool-down of the stream that goes to the catalytic section, operation that usually needed an external coolant in order to bring temperature around the required one (circa 300°C). The acid gas feed is usually available at quite low temperatures (max 200°C) so in order to reduce the duty and the dimensions of the pre-heating unit another small exchanger can be anticipated by a small exchanger working with an external fluid able to increase the temperature at least to 250°C.

1. RTR Model

In order to manage the task of building a simulation architecture able to contemplate, at the same time, standard unit operations like heat exchange and the relative equipment together with reacting systems having complex kinetic mechanisms and the reactor environment, an integration between Aspen Hysys and OpenSMOKE++ has been designed.

The first software is, in fact, capable to quickly run a great number of standardized computation regarding a whole lot of chemical industry unit operations (ranging from simple heat exchanger to distillation columns and so on) but when it comes to reacting system its applicability is limited to thermodynamic equilibrium situations (the so-called Gibbs Reactor) or ideal PFRs / CSTRs.

Meanwhile OpenSMOKE++, while appositely studied to perform computations upon personalized kinetics in a big range of different reactors, makes simulations of heat exchangers and similar equipment not impossible but unnecessarily complicated. The ideal configuration is then the one that combine the positive traits of both instruments, forcing them to interact with each other.

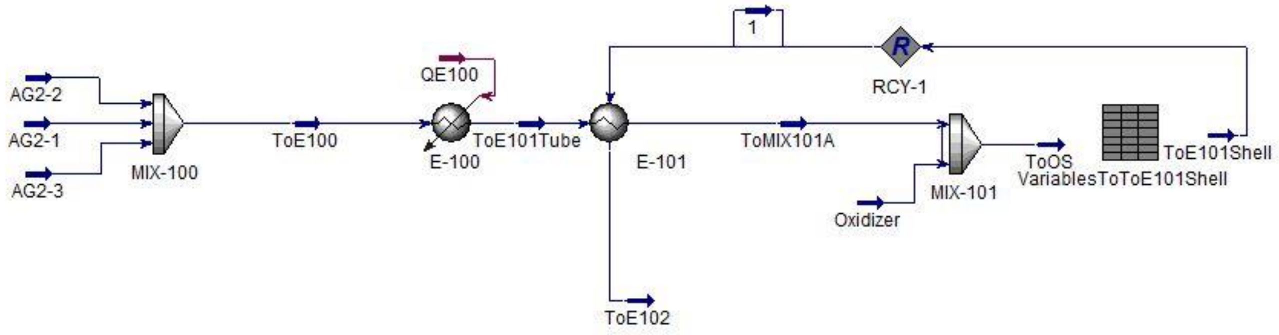


Figure 33: Aspen Hysys scheme for RTR simulation

1.1. Kinetic Scheme

A new and more complete kinetic scheme of the AG2S™ process has recently been developed at Politecnico di Milano. The previous studies were based mainly on the development of detailed kinetic models of the reacting systems in the furnace, to improve the prediction capacity for H_2S pyrolysis first, and then for the interaction of this compound with O_2.

The new study was carried out with a block-kinetic approach to construct the detailed mechanism; restarting from pyrolysis of H_2S, key block for this novel technology, continuing through the oxidation of H_2S, that provides the heat required to activate the pyrolysis, and concluding with the study of the interaction of basic Sulfur compound with C1 species to produce pollutant agents like COS and CS_2.

This work was finally integrated with the detailed mechanism of pyrolysis, partial oxidation and combustion of hydrocarbon compounds up to 3 C atoms; performed by the CRECK Modelling group with the new ARAMCO approach, that is considering a different, from the Lindemann-Troe, dependence from pressure; it is remarkable to denote the importance of this section due to the reduction reactions that convert carbon dioxide to carbon monoxide.

The whole set of reactions with their parameters for the Arrhenius expression, are reported in the tables of Appendix A.

1.2. Primary Structure: Aspen Hysys

When building a process scheme composed by many parts, simplicity and user-friendliness make Hysys the go-to choice for modeling and simulation.

In this case, to represent a thermal furnace, the following set-up has been developed. Three streams represent the acid gases at the inlet of the process: the composition of the mixture has been split between H_2S (“AG2S™ -1”), CO_2 (“AG2S™ -2”) and all the other components(“AG2S™ -3”; for simplicity and velocity reason only H_2O has been considered). This comes particularly useful to investigate the initial ratio between hydrogen sulfide and carbon dioxide and correlate this information to the overall performance of the unit in terms of conversion and not only (this topic will be further discussed in Part 4).

The streams are re-united by means of a fictional mixer and then the product is pre-heated by two exchanger. The second one implies the use of two different process streams, in which one of them is the outlet of the WHB. This configuration generates a convergence problem: Hysys’ recycle tool (RCY-1) allows to set a first-try temperature value necessary to solve the E-101 unit but after one simulation of the thermal section is complete a new temperature is computed, thus changing the exchanger performances and the initial conditions for the simulation itself. The convergence problem will be addressed and resolved in the next chapter, at the extern of Hysys where the core of the model lies, i.e. Matlab.

Matlab is, in fact, the environment chosen to collect data from Hysys and print it into simple ASCII files that represent the input of the OpenSMOKE++ solvers. Fig 34 graphically schematize the flow of data.

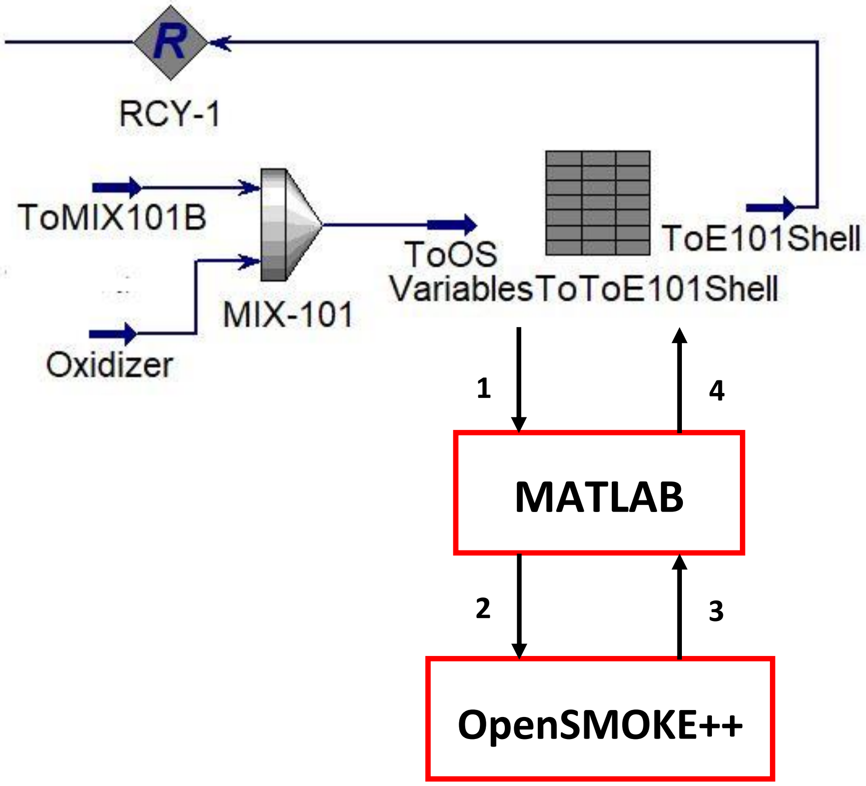


Figure 34: Particular of interaction between the three simulation environments

1.3. Secondary Structure: Matlab

The advanced programming language, which Matlab is built upon, makes it possible to integrate the environment with a wide range of very different tools. The instruments needed to perform passages 1 and 2 (and respectively 4 and 3) pictured in Fig 34 are nonetheless very different; below follows detailed explanation of both of them.

1.3.1. Integration between Matlab and Aspen Hysys.

The Microsoft Component Object Model (COM) is a platform-independent, distributed, object-oriented system for creating binary software components that can interact. COM is the foundation technology for Microsoft's OLE (compound documents), ActiveX (Internet-enabled components), as well as others.

To understand COM (and therefore all COM-based technologies), it is crucial to understand that it is not an object-oriented language but a standard for objects. COM specifies an object model and programming requirements that enable COM objects (also called COM components, or sometimes simply objects) to interact with other objects. These objects can be within a single process, in other processes, and can even be on remote computers. They can be written in different languages, and they may be structurally quite dissimilar, which is why COM is referred to as a binary standard; a standard that applies after a program has been translated to binary machine code.

COM only defines the essential nature of a COM object. In general, a software object is made up of a set of data and the functions that manipulate the data. A COM object is one in which access to an object's data is achieved exclusively through one or more sets of related functions. These function sets are called interfaces, and the functions of an interface are called methods. Further, COM requires that the only way to gain access to the methods of an interface is through a pointer to the interface.

Besides specifying the basic binary object standard, COM defines certain basic interfaces that provide functions common to all COM-based technologies, and it provides a small number of functions that all components require.

- A COM object is a software component that conforms to the Component Object Model. COM enforces encapsulation of the object, preventing direct access of its data and implementation. COM objects expose interfaces, which consist of properties, methods and events.
- A COM client is a program that makes use of COM objects. COM objects that expose functionality for use are called COM servers. COM servers can be in-process or out-of-process. An example of an out-of-process server is Microsoft Excel spreadsheet program.
- A Microsoft ActiveX control is a type of in-process COM server that requires a control container. ActiveX controls typically have a user interface. An example is the Microsoft Calendar control. A control container is an application capable of hosting ActiveX controls. A MATLAB® figure window or Aspen Hysys application are examples of control containers.

MATLAB can be used as either a COM client or a COM Automation server. The first one is the case that suits our purposes. A COM client is a program that manipulates COM objects. These objects can run in the MATLAB application or can be part of another application that exposes its objects as a programmatic interface to the application (e.g. Excel, Aspen Hysys)..

Using MATLAB as a COM client provides two techniques for developing programs in MATLAB:

1) Including COM components in your MATLAB application.
2) Accessing existing applications that expose objects via Automation.

In a typical scenario, MATLAB creates ActiveX controls in figure windows, which are manipulated by MATLAB through the controls' properties, methods, and events. This is useful because there exists a wide variety of graphical user interface components implemented as ActiveX controls.

MATLAB COM clients can access applications that support Automation, such as the Excel spreadsheet program. In this case, MATLAB creates an Automation server in which to run the application and returns a handle to the primary interface for the object created.

MATLAB software provides two functions to create a COM object:

1) `actxcontrol` — Creates a Microsoft ActiveX control in a MATLAB figure window.
2) `actxserver` — Creates an in-process server for a dynamic link library (DLL) component or an out-of-process server for an executable (EXE) component.

The following diagram shows the basic steps in creating the server process; highlighted are the steps implemented in order to actually connect Matlab to Aspen Hysys done in this work:

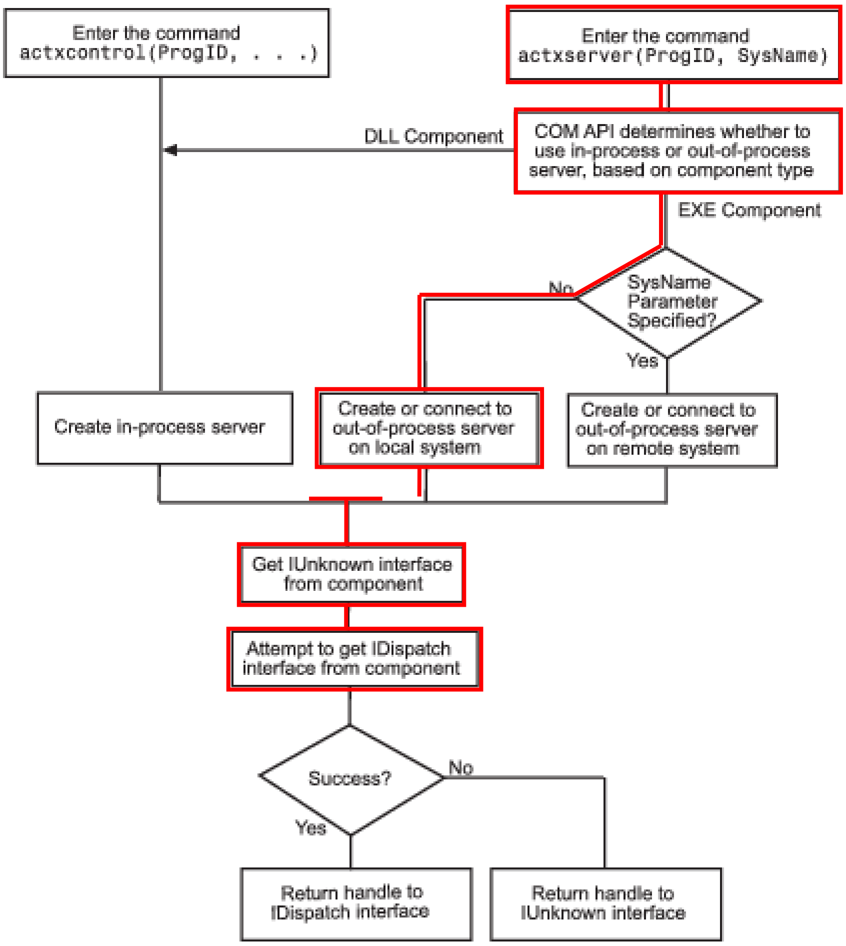

Enter the command
actxcontrol(ProgID, . . .)
Enter the command
actxserver(ProgID, SysName)
DLL Component
COM API determines whether to use in-process or out-of-process server, based on component type
EXE Component
No
SysName Parameter Specified?
Yes
Create in-process server
Create or connect to out-of-process server on local system
Create or connect to out-of-process server on remote system
Get IUnknown interface from component
Attempt to get IDispatch interface from component
Success?
No
Yes
Return handle to IDispatch interface
Return handle to IUnknown interface

Figure 35: Logical steps to perform in order to manage Aspen Hysys or other COM based program data from Matlab

Following this procedure it is possible to access and modify the data belonging to the object (streams) handled by Hysys: for example, the commands

```
h = actxserver('Hysys.Application');
hyCase = h.Activedocument;
f = hyCase.Flowsheet;
g = f.MaterialStreams;
g.Item(‘AG2S-1’).MassFlowValue
```

set up Hysys.exe as the external server, call its active flowsheet and the value of the flow rate in massive terms of the stream AG2S™ -1.

1.3.2. Integration between Matlab and OpenSMOKE++

The built-in functions **`fopen`**, **`fprintf`**, **`dos`**, **`importdata`** can easily be combined in order to create a text file, write data into it, execute DOS commands and read their output. Being ASCII files the operational basis for OpenSMOKE++’s solver to work, this is all it is needed to integrate the two simulation environments.

1.4. Numerical Convergence

As explained earlier, a convergence problem arises from this model, thanks to the presence of a “recycle” of the WHB outlet stream, used as hot fluid in the process pre-heater before the inlet of the furnace. It of interest, then, to secure that the results of the simulation come close to the real ones, excluded a certain tolerance ε. As those results come from unknown equations (i.e. not explicitly written and solved directly by Aspen Hysys), a simple possible solution in represented by the successive substitution method.

The method of successive substitutions (also called fixed point iteration) is perhaps the simplest method of obtaining a solution to a nonlinear equation. This technique begins by rearranging the basic F x() 0 equation so that the variable x is given as some new function of the same variable x . The original equation is thus converted into an equation of the form:

$$x = G(x)$$

Starting with an initial guess, an iterative procedure is then initiated where the value of x is substituted into the right hand side function and a new value of G(x) is calculated. The procedure is iterated until there is no further change in x or until the change is less than some desired accuracy. However, the convergence is slow, and follows what is known as “linear” convergence. This means that as the true solution is approached, the error at each iteration is some linear fraction of the error at the previous iteration. Let’s now study the convergence rate and the condition for convergence of the successive substitution method. Let’s let

$$x_1 = x_0 + \delta x_1$$

where x_0 is the true solution value and

$$\delta x_1 = x_1 - x_0$$

is the error in the solution value at step 1 in the iteration. The next approximation to the solution is obtained from:

$$x_2 = G(x_1) = G(x_0 + \delta x_1) = G(x_0) + G'(x_0)\delta x_1$$

where the first term in a Taylor series has been used to approximate the value of the function $G(x_0 + \delta x_1)$. Since the new calculated value must be of the form: $x_2 = x_0 + \delta x_2$ and $x_0 = G(x_0)$, it is seen that the error in the value at the next iteration is:

$$\delta x_2 = G'(x_0)\delta x_1$$

or

$$\varepsilon = \frac{\delta x_2}{x_0} = G'(x_0)\varepsilon_1$$

Thus the error or relative error at each iteration is linearly proportional to the error or relative error at the previous iteration and the criteria for a smaller error with each iteration is that:

$$|G'(x_0)| < 1$$

The magnitude of the derivative of the successive substitution function evaluated at the solution point must be less than unity for the error to decrease with each iteration. If the derivative is negative, the calculated values will oscillate about the solution point while if it is positive the calculated values will monotonically approach the solution point.

This convergence method has three main problems: the convergence criteria, the selection of a guess value and the criteria for selecting a correct damping factor. The first one is related to the criteria convergence, since is very difficult to decide a correct convergence criterion that permits to define a good convergence region, not too big and not too small. For this kind of problem, the criterion is a congruence criterion where the variables of two streams have to be equal and are evaluated by the black-box solver (the simulators). Since the discrepancy between two iterations is not too big and is not so impacting on the computational time, the only question remains in how much big can be this discrepancy value. The solution will be good enough when it approaches 0.

Regarding the guess values ,the issue consists in selecting the starting guess point, afflicted by the convergence region. Because with a guess point out from a possible convergence region can take to a divergence of the problem or to converge in another convergence region that could be not the desired one. The last one, the oscillation problem and the damping factor selection criteria, is extremely affecting the convergence time of the algorithm. It can happen that the problem oscillates between two possible values and a damping factor is sometimes required to decrease the possibilities to have an oscillation, but how to choose a suitable value remains unsolved.

To solve the last issue presented before and to improve the resolution, a modification to this algorithm and to the simulator has been done. The first one is the introduction of an anti-ringing criterion, with a similar logic of the SSC on the damping factor, where starting from a first value of the damping factor of 1, every iteration a comparison of the oscillation factor with its criteria is done. This anti-ringing criterion checks and acts every three iterations and compare the 3 discrepancies between the 4 consecutives iterations. If it seems that the first value and the third value are sufficiently equal, then the damping factor is reduced, if not then it continues through the sequent iteration until convergence.

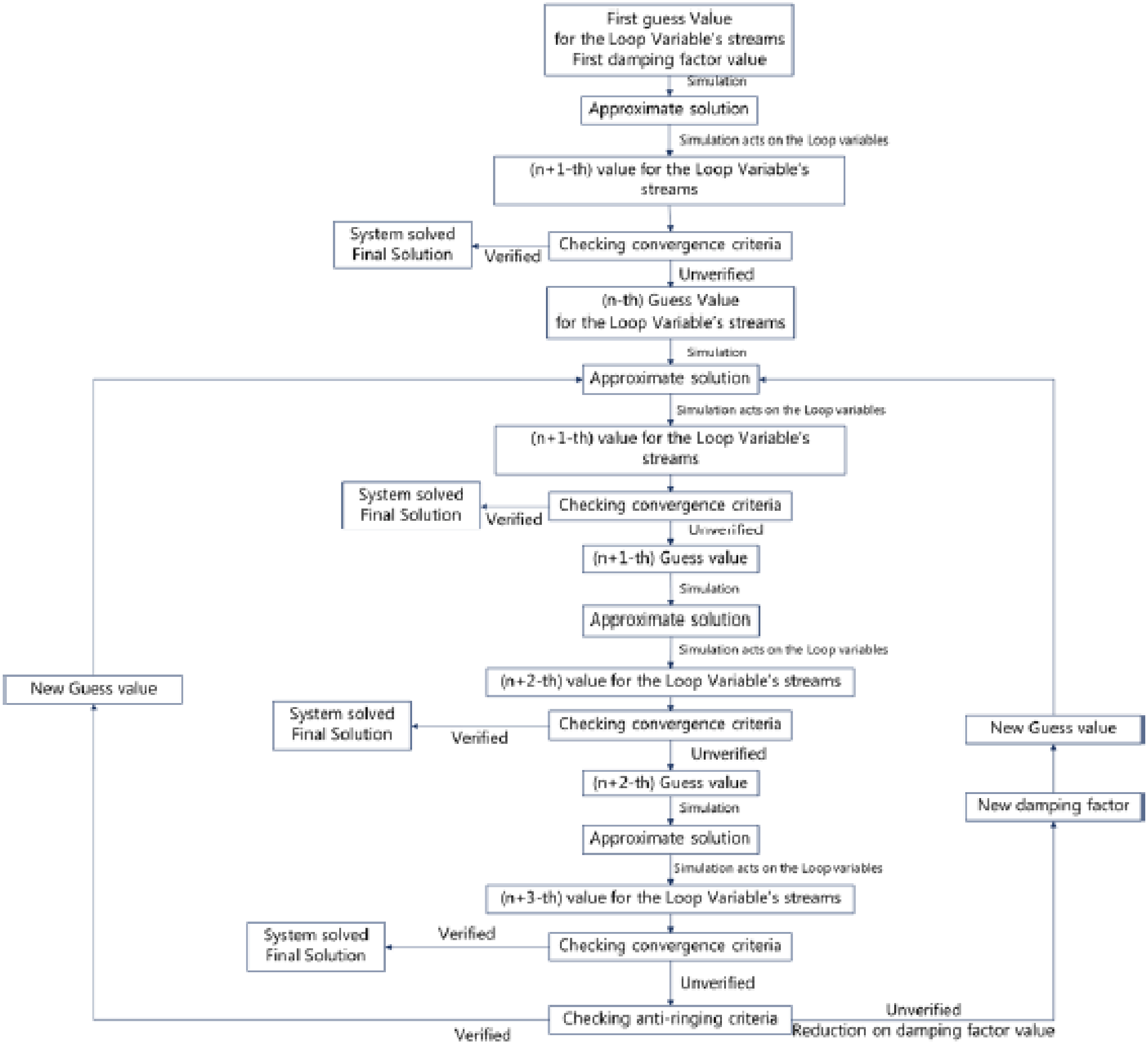

First guess Value
for the Loop Variable's streams
First damping factor value
Simulation
Approximate solution
Simulation acts on the Loop variables
(n+1-th) value for the Loop Variable's streams
System solved
Final Solution
Verified
Checking convergence criteria
Unverified
(n-th) Guess Value
for the Loop Variable's streams
Simulation
Approximate solution
Simulation acts on the Loop variables
(n+1-th) value for the Loop Variable's streams
System solved
Final Solution
Verified
Checking convergence criteria
Unverified
(n+1-th) Guess value
Simulation
Approximate solution
Simulation acts on the Loop variables
New Guess value
(n+2-th) value for the Loop Variable's streams
System solved
Final Solution
Verified
Checking convergence criteria
Unverified
New Guess value
(n+2-th) Guess value
Simulation
New damping factor
Approximate solution
Simulation acts on the Loop variables
(n+3-th) value for the Loop Variable's streams
System solved
Final Solution
Verified
Checking convergence criteria
Unverified
Checking anti-ringing criteria
Unverified
Reduction on damping factor value
Verified

Figure 36: Block representation of the successive substitution algorithm integrated with an anti-ringing criterion

2. AG2S™ Process Simulation

The process, as Manenti suggests [12], is operated using a pure stream of O_2 as oxidizer. This choice can immediately lead to some assumptions: the furnace, adopted from a Claus plant, will be over-dimensioned and the maximum temperature reached inside the combustion chamber will be higher than before. While this is undoubtedly beneficial from an acid gas conversion (and Sulfur recovery) point of view, especially If coupled with the pre-heating of the inlet feed, it will lead to lower yields in terms of syngas production favoring complete oxidations towards water and SO_2 (so Sulphur by reflex).

2.1. Shiraz: Claus vs AG2S™

The H_2S/CO_2 ratio for the fresh acid gas stands around 0.524. Flame temperature is consistent with the one reached inside the classic combustion chamber for the Claus process; here high thermal levels are kept for basically the whole volume of the burner thanks to the null inert gas presence. The WHB is still able to bring the outlet temperature down to the usual values; here the temperature has to be lowered in the shorter residence time possible because, as we see, re-oxidation to CO_2 is very fast and happens at the expenses of the fresh produced syngas (and SO_2) too.

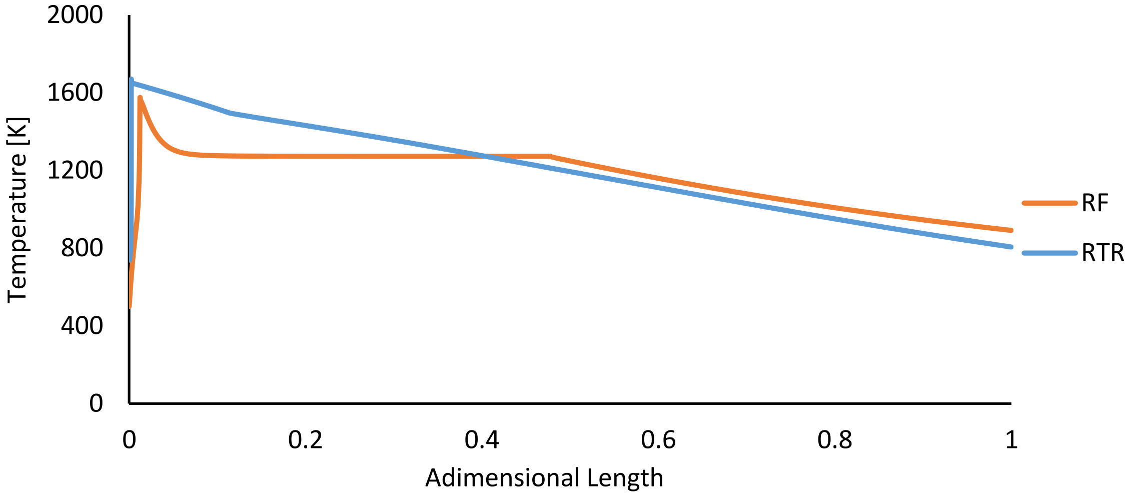


Figure 37: Claus vs AG2S™ temperature profile comparison for the Shiraz furnace

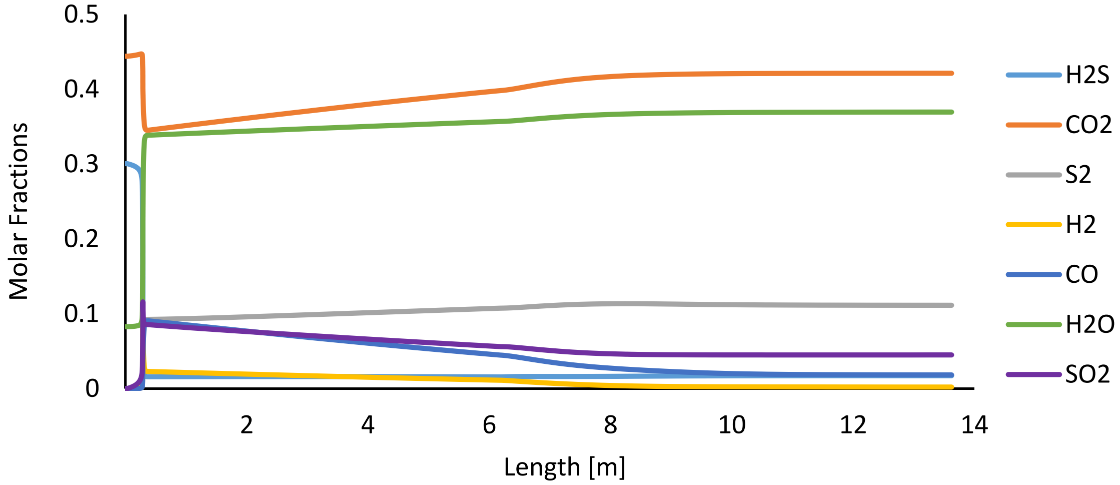


Figure 38: AG2S™ composition profiles in the Shiraz furnace

As anticipated before, high temperatures cause a positive shift in water production, which result much higher than before. CO_2 concentration also rises, although not at the same extent as H_2O, as a result of complete oxidation reaction; the same can be said for S_2. H_2 production, after an initial spike, rapidly stops as the specie is converted to water, following a thermodynamically favorable path.CO presents an analogous profile with a less steep decrease rate. The strong oxidation environment also affects SO_2 synthesis, more relevant than in the Claus furnace: its stability in the flame zone is a downside of the process, as Sulfur oxides represent a cost since they have to be carefully removed in the catalytic and washing sections, in order to avoid their emission to the atmosphere.

Outlet stream composition comparison is presented below:

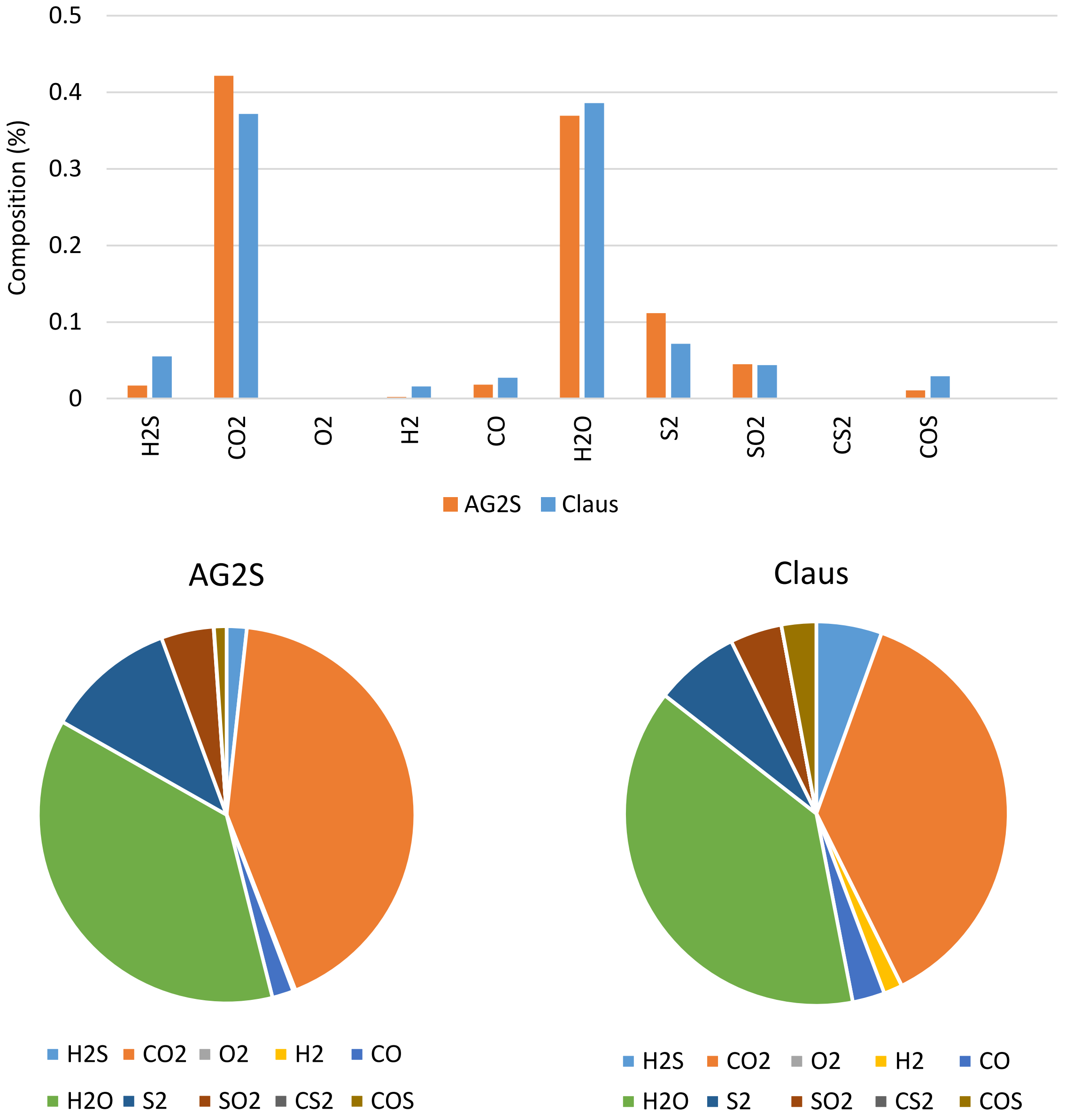


Figure 39: Claus vs AG2S™ composition of the outlet stream comparison for the Shiraz furnace

Overall, this process configuration shows a 6% selectivity towards syngas while the yields stands around 2.3% (molar values). Acid gas combined conversion reaches 40%. The first impression is that, while RTR seems promising as it broaden the field of applicability of SRUs for a larger range of acid gas compositions, its aim to obtain a somewhat high selectivity to syngas cannot be accomplished at the same working conditions of a standard Claus thermal furnace. Further studies will later expand this concept.

Table 13 sum up the inlet and outlet conditions considered in this simulation:

INLET	**ACID GAS**	**OXIDIZER**
H_2S	222 kmol/h	---
CO_2	327.46 kmol/h	---
H_2O	61 kmol/h	---
O_2	---	127.4 kmol/h
Temperature	218 °C	220°C
Pressure	1.48 atm	1.48 atm

Table 13: Shiraz Furnace: composition and condition of the inlet stream

OUTLET (molar fractions)	**Claus**	**AG2S™**
H_2S	5.517117441	1.719116
CO_2	37.15967293	42.13685
O_2	0	9.20323E-12
H_2	1.572468326	0.199763
CO	2.709138287	1.823151
H_2O	38.59286549	36.95317
S_2	7.1524845	11.13149
SO_2	4.371461946	4.49193
CS_2	0	2.48E-02
COS	2.924791086	1.07
T_{MAX}	1300 °C	1395 °C

Table 14: Composition of the outlet stream of the Shiraz furnace (Claus vs AG2S™)

2.2. Khangiran: Claus vs AG2S™

Once again, the acid gas feed is richer in CO_2 than in H_2S: this causes a consistent oxidation of hydrogen sulfide to S_2 (with H_2 as byproduct) at the expenses of carbon dioxide in the preheating zone. Following a rapid increase of temperature, the favored reduction pathway becomes the one to CO (CO_2 would be too stable to undergo this reaction); hydrogen is oxidized to water and Sulfur to SO_2. Excluding the carbon monoxide backwards reaction, in the WHB no significant composition changes are spotted: as a matter of fact, the cool-down in this zone is very limited (only 150 degrees); from a certain point of view this is beneficial since syngas quantity does not decrease, but so does SO_2.

Similarly to what has been stated before, the big temperature increase is related to the very exothermic reaction of H_2O production. The reacting mass to be heated is one order of magnitude bigger than before: for this reason the combustion mechanism starts way after the flame inside the combustion chamber; this configuration may not be optimal from an energetic point of view as fuel is basically wasted in a pure direct thermal exchange application.

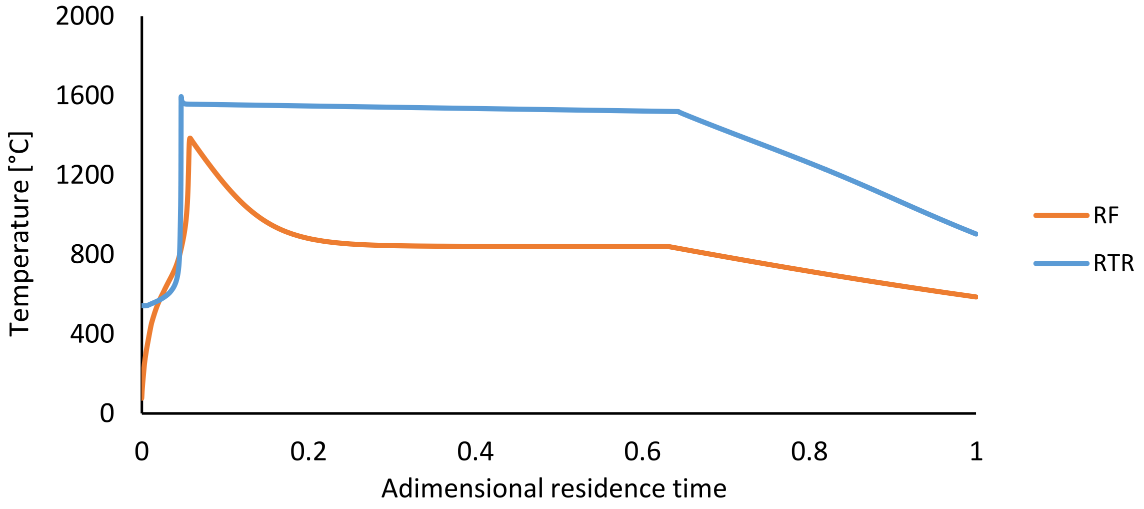


Figure 40: Claus vs AG2S™ temperature profile comparison for the Khangiran furnace

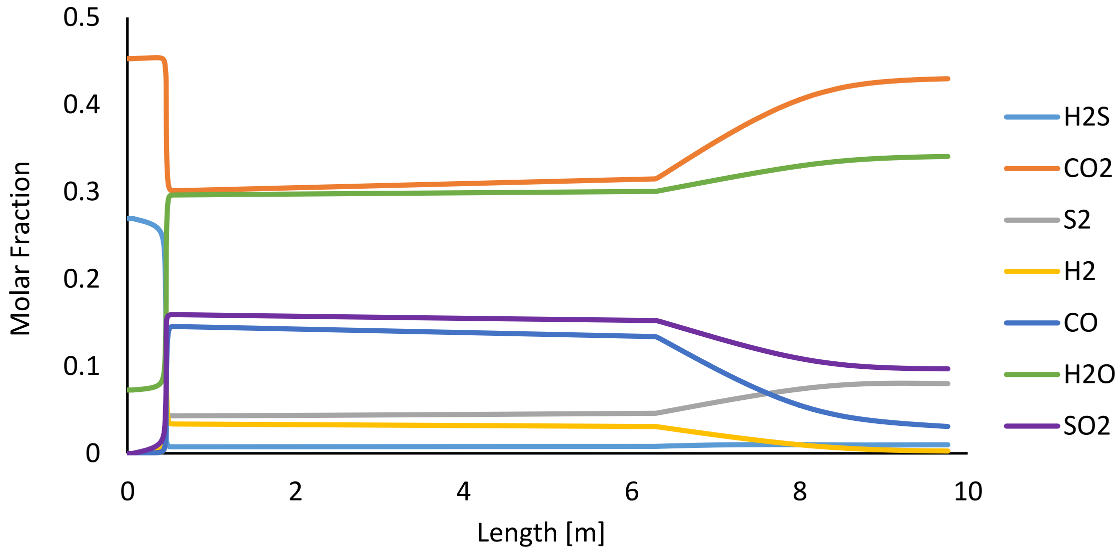


Figure 41: AG2S™ composition profiles in the Khangiran furnace

The peculiar temperature profile reflects on the reactivity of the mixture: significant conversion of CO_2 and mainly H_2S starts towards the end of the combustion chamber, as well as syngas formation. Paradoxically, this characteristic is attractive since it does not give time to H_2 and CO to recombine into more thermodynamically stable species, only in the WHB (as usual) the reduction of temperature enables this reaction path.

Overall hydrogen an carbon monoxide production is already preferable in the AG2S™ process with respect to the Claus configuration. H_2S conversion, SO_2 and COS reduction contribute in making RTR a very good solution in this case. This motivates us in further investigating the potential of the novel process layout.

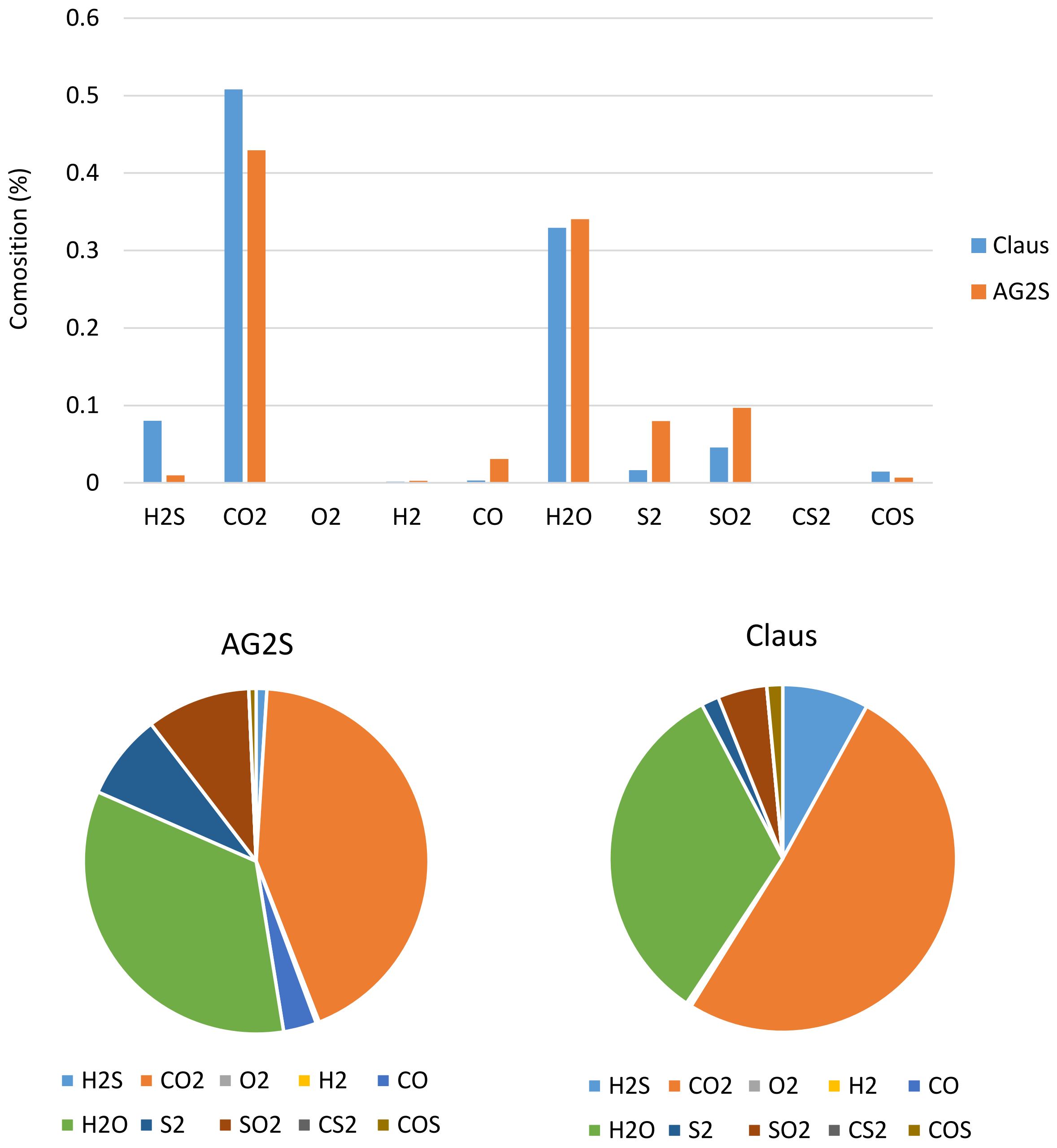


Figure 42: Claus vs AG2S™ composition of the outlet stream comparison for the Khangiran furnace

Molar selectivity reaches 8.5% while yield is up to 3.1% (1.7% massive). The two parameters combine to give a global acid conversion of circa 36%. Although better than in the case before, this result could surely benefit from an optimization of the operating conditions, done in the next chapters.

Finally, the tables below report the inlet streams conditions and compare the outlet of both process solutions:

INLET	ACID GAS	OXIDIZER
H_2S	901.45 kmol/h	---
CO_2	1510.47 kmol/h	---
H_2O	242.80 kmol/h	---
O_2	---	526.19 kmol/h
Temperature	52 °C	77 °C
Pressure	1.283 atm	1.283 atm

Table 15: Khangiran Furnace: composition and condition of the inlet stream

OUTLET (molar fractions)	Claus	AG2S™
H_2S	8.032786885	3.508496268
CO_2	50.81967213	46.26891967
O_2	0	1.27303E-11
H_2	0.163934426	0.527051385
CO	0.327868852	1.900362334
H_2O	32.95081967	33.0448204
S_2	1.639344262	10.75336983
SO_2	4.590163934	3.112079443
CS_2	0	0.028765901
COS	1.475409836	0.856134356
T_{MAX}	1000 °C	1224 °C

Table 16: Composition of the outlet stream of the Khangiran furnace (Claus vs AG2S™)

PART 3 – NUMERICAL OPTIMIZATION: ALGORITHM ARCHITECTURE

Optimization of a process under certain constraints (operational, economical, normative, etc.) is a core part of industrial engineering. Formulations of the problem may vary greatly but the aim remains to find the minimum of a function F(**x**) involving n_V variables, **x** ∈ R^{nV} , with $n_V > 1$. For maximization problems, all that needs to be done is to invert the sign of the objective function.

If we consider the multidimensional problem of finding the vector **x** that minimizes/maximizes the function F(**x**), many algorithms are based on the detection of a search direction, $\mathbf{p}_i$, across which the optimum should be searched for, starting from the initial guess, $\mathbf{x}_i$:

$$F(t) = F(\boldsymbol{x}_i + t\boldsymbol{p}_i)$$

Thus, the multidimensional problem is reduced to a one-dimensional optimization and it may therefore be an idea to use general one-dimensional program to handle multidimensional problems. General programs for one-dimensional optimization are rarely included in the most recent implementations of multidimensional optimization programs. Certain special one-dimensional methods, which are dedicated to multidimensional problems, are usually adopted instead.

Two kinds of algorithms are proposed in the one-dimensional case: the methods that compare different values of the function (comparison methods) and the methods that approximate the function with simpler functions to search for the minimum. Comparison methods such as Fibonacci's method and golden section search exploit function unimodality within a specific interval of uncertainty with the aim of reducing it progressively.

In the multidimensional case, the methods based on the gradual reduction of the region of uncertainty quickly lose their efficiency as the problem's dimension increases. The one-dimensional case differs from other cases: as a method that reduces the region of uncertainty, it evenly reduces the uncertainty on variables along which the minimum is being sought. Therefore, when the region of uncertainty is reduced by a factor of 100, the uncertainty of the variable is evenly reduced by 100 in the one-dimensional case. In a two-dimensional problem, when the region of uncertainty is reduced by a factor of 100, the uncertainty on both the search variables is only reduced by 10. As problem dimensions increase, only slight improvements are obtained for the search variables, even though very high-performance methods are adopted to reduce the region of uncertainty: the efficiency gap between one- and multidimensional problems increases exponentially with the dimension of the optimization problem.

If we add to this the difficulty of defining function unimodality in a multidimensional space, it is easy to see why these methods are only adopted for one-dimensional problems.

The fact that methods that reduce the region of uncertainty cannot be used leads to another serious problem: it is not possible to know a priori how many iterations are necessary to achieve an assigned precision. Even with methods that approximate the function to be minimized by means of a simpler function, there are significant differences between one- and multidimensional cases. The number of points necessary to approximate the function with a quadratic function increases by n_V^2.

There is another disadvantage when $n_V > 1$. While in the one-dimensional case, the new function evaluation can be exploited at each iteration to satisfactorily upgrade the parabolic approximation; in the multidimensional case, we have useful information only on the search direction.

Lastly, when $n_V > 1$ a new difficulty arises with respect to the one-dimensional case that can make searching for the minimum very hard. As mentioned above, the function contours are strengthened along a direction when the function is quadratic and the Hessian condition number is large. Therefore, very narrow valleys may be present in the multidimensional case but not in the one-dimensional case. One algorithm may easily find the bottom of the valley, but may encounter huge difficulties in moving along the valley (towards the function minimum).

Therefore, the multidimensional minimization problem is significantly harder to solve than one-dimensional minimization.

1. Hard Mono and Multi-Dimensional Optimization Problems

Hard optimization issues require a change of strategy, in both one-dimensional and multidimensional cases. As the minimum has to be found in a set of minima, one initial essential modification is required. In the case of unimodal functions, only the values of abscissas t_i and of functions $f_i = F(t_i)$ to define the next iteration are collected, whereas when the function is multimodal, they have to be collected into two dedicated vectors, **t** and **f** . This is the only way to check for the existence of intervals that may contain a local minimum.

A second modification is also needed since the function or its derivatives may present either some discontinuities or non-evaluable regions in correspondence with t_i. Parabolic and cubic interpolation methods may be unreliable and, therefore, it is opportune to give a smaller weight to the prediction coming from them.

We assume that

1) the search must be carried out within the interval $l \le t \le u$ (l and u vectors of the lower and upper constraints);

2) n points are collected and sorted from the smallest t_i and all the points are distinct: $t_i < t_{i+1}$;

3) in correspondence with each point t_i, the value $f_i = F(t_i)$ is collected.

We can deduce that

- there is a minimum between $t_1 = l$ and t_2 , if $f_1 < f_2$;
- there is a minimum between t_{n-1} and $t_n = l$, if f $f_{n-1} > f_n$;
- there is a minimum between t_{i-1} and t_{n+1}, if $f_{i-1} < f_i < f_{i+1}$
- there could be a minimum between t_i and t_{i+1}, if $f_i = f_{i+1}$ or if the function is feasible in t_i but infeasible in t_{i+1}, or if the function is feasible in t_{i+1} but infeasible in t_i;
- a certain number of points can be inserted between t_i and t_{i+1} if the distance between them is particularly large.

In the special case of optimization problems with two variables, the following alternatives may be used: the problem is considered as the smallest multidimensional optimization or the problem is considered as a special case of one-dimensional optimization. In fact, it is possible to write the following relation:

$$\min_{x_1,x_2}\left(F(x_1,x_2)\right) = \min_{x_1}\left(\min_{x_2}(F(x_1,x_2))\right) = \min_{x_2}\left(\min_{x_1}(F(x_1,x_2))\right)$$

The third relation means that for each value of the variable x_2, a one-dimensional minimum of the function $F(x_1,x_2)$ is searched for with respect to the variable x_1. The variable x_2 is, in turn, used to minimize the function

$$\theta(x_2) = \min_{x_1}\big(F(x_1, x_2)\big)$$

This minimum can also be obtained by means of a one-dimensional search. For example, the Haupt function for the two-dimensional space is considered:

$$F_{HAUPT} = x_1 \sin(4x_1) + x_2 \sin(2x_2)$$

A one-dimensional search performed on the variable x_1 in correspondence with x_2 = 2:5 results in six distinct minima for the function F_{HAUPT}. The global minimum is equal to -7:184 in x1 = 9:823.

The function $\theta(x_2) = \min_{x_1}\big(F(x_1, x_2)\big)$ to be minimized with respect to x_2 is also multimodal. Its minimization, therefore, requires a robust algorithm for one-dimensional minimization.

2. Robust Optimization Methods

Many heuristic methods have been proposed, including genetic and other evolutionary algorithms (Goldberg, 1989; Holland, 1992; Schwefel, 1995), artificial neural networks (Haykin, 1994), tabu-search methods (Glover, 1989), simulated annealing (Kirkpatrick et al., 1983), particle swarm and population-based methods (Kennedy and Eberhart, 1995; Parsopoulos and Vrahatis, 2002), colony optimization (Dorigo and Maria, 1997). A description of a selection of these algorithms can be found in Haupt and Haupt (2004).

A new strategy proposed by Buzzi-Ferraris and Manenti (2010) is described below. This strategy exploits the features of the Optnov method combined with the ones of the Simplex method, originally developed to solve very narrow valley problems. Let us suppose we know a direction that is oriented more or less along the bottom of the valley. The mistake made by many strategies is to adopt such a direction as the axis along which we should search for the one-dimensional minimum. Such a search will prove ineffective, usually because the direction is imprecise and the valley is nonlinear. To exploit the search direction that detects the bottom of the valley in a more or less accurate manner, it is necessary to shift our point of view.

2.1. Simplex Method

The simplex method was originally proposed by Spendley, Hext & Himsworth, 1962 [49] and was subsequently improved and modified by Nelder & Mead, 1965 [40]. In this method, a set of N + 1 (with N = nv) different vertices $\boldsymbol{v}_0$, $\boldsymbol{v}_1$,… $\boldsymbol{v}_{N}$−1,$\boldsymbol{v}_N$ is called the simplex. Vertices are sorted to have increasing function values with respect to the index of vertices:

$F_0 \leq F_1 \leq \cdots \leq F_{N-1} \leq F_N$. The vertex $\boldsymbol{v}_0$ contains the best (minimum) value and $\boldsymbol{v}_N$ the worst value of the N + 1 vertices. The barycenter $\boldsymbol{v}_B$ of the vertices from 0 to N − 1 is calculated through the arithmetic mean of their coordinates by excluding the worst vertex $\boldsymbol{v}_N$.

The method is based on three fundamental operations: reflection, expansion, and contraction. Given N + 1 distinct initial vertices, the new vertex v_R is obtained by reflecting the worst vertex v_N with respect to the barycenter v_B:

$$v_R = v_B + \alpha(v_B - v_N) \quad \alpha > 0$$

The following cases may occur:

1) The function in v_R is better than v_0. In such a case, the reflection is to be expanded:

$$v_E = v_B + \gamma(v_R - v_B) \quad \gamma > 1$$

If F(v_E) < F(v_R), the point v_E is introduced in the simplex by replacing vN (the new vertices of the simplex are v_E, v_0, v_1, … , v_{N-1}). On the other hand, if F(v_E) ≥ F(v_R), the point v_R is introduced in the simplex by replacing v_N (the new vertices of the simplex are v_R, v_0, v_1, … , v_{N-1}).

2) The vertex v_R is worse than v_0 but better than v_{N-1}: $F_0 < F_R < F_{N-1} < F_N$. In such case v_R is introduced in the simplex by replacing vN (the new vertices of the simplex are v_0, … , v_R,… , v_{N-1}).

3) The vertex v_R is worse than v_0, better than v_N, but worse than v_{N-1}: $F_0 < F_{N-1} < F_R < F_N$. In this case, a contraction is performed:

$$v_C = v_B + \beta(v_R - v_B) \quad 0 < \beta < 1$$

4) The vertex v_R is worse than v_N: $F_0 < … < F_N < F_R$. In this case, a contraction in the opposite direction (with respect to the previous reflection) is carried out:

$$v_C = v_B + \beta(v_N - v_B) \quad 0 < \beta < 1$$

If the vertex v_C evaluated with the two equations above is better than v_N, this is replaced by the same v_C and the new vertices of the simplex are v_0, … , v_{N-1}, v_C or v_0, … , v_C, … , v_{N-1}.

Otherwhise the simplex is contracted in the neighborhood of the best vertex $v0$ by means of the formula:

$$v_i = v_i - \delta(v_i - v_0) \quad 0 < \delta < 1$$

The values suggested by Nelder & Mead are: α = 1, β = 0.5, γ = 2 and δ = 0.5. The method continues while the distance of vertices v_i from v_0 is larger than an assigned tolerance. Another stop criterion is to check that the function is not constant in all vertices of the simplex. The Simplex method has the following pros and cons:

Pros:

- The objective function does not have any special requirements: in fact, it can be non-derivable and discontinuous.
- It solves relatively narrow valleys.

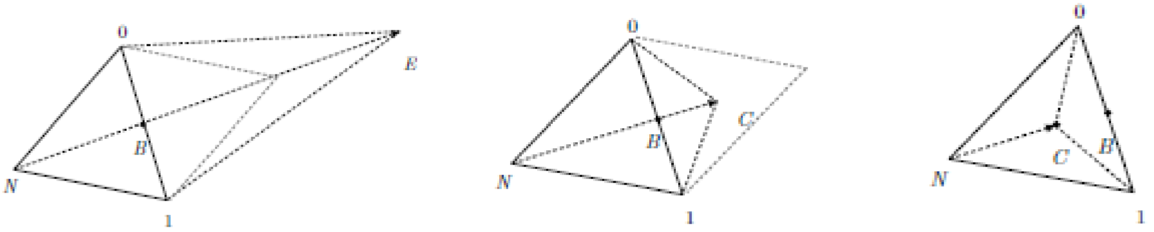


Figure 43: Graphical representation of expansion (a) and compression (b) operations in the Simplex method

- It is a good method when coupled with more performing algorithms.

Cons:

- It does not allow the minimum to be detected with a high level of accuracy except after a large number of calculations.
- It requires a lot of memory allocation when the number of variables is large.
- The method can be used only for small- and medium-scale problems.
- In the presence of narrow valleys, the vertices may collapse in a sub-space. If this happens, the method is unable to find the function minimum.
- It does not exploit the function information obtained during the search: it is not high-performance when the function is easy.

2.2. Optnov-Simplex Hybrid Method

The Optnov-Simplex hybrid method is based on some simple ideas that make it particularly robust for very narrow valleys:

- Any optimization algorithm can find the bottom of the valley by starting from a point outside the same valley;
- The line joining two points on the bottom of the valley is a reasonable valley direction; therefore, there is a good probability that a point projected along such a direction will be close to the valley.
- Nevertheless, this valley direction must not be used as the one-dimensional search direction, but rather as a direction along which a new point projection must be carried out.
- This new point should not be discarded even though it is worse than the previous one, rather it is the new starting point for a new search.
- This search must be performed in the subspace orthogonal to the valley direction to prevent the issue of small steps arising. This philosophy is particularly simple in object-oriented programming. The optimization problem is split into two different levels:

1) The first (outer optimizer) is managed by a single object that exploits the Optnov method to find a certain number, N, of points to initialize an even number of objects.

2) In the second (inner optimizer), each object uses a program to search for the minimum with a limited number of iterations starting from the point assigned by the outer optimizer. N is the number of points managed by the external optimizer to initialize an even number of optimization inner objects.

This philosophy is useful in solving all problems demanding algorithm robustness:

- When the function has many minima and we need to search for the global minimum.
- When the function has very narrow valleys (or steep walls).
- When the function is undefined somewhere.

This philosophy is particularly effective when several processors are available. Actually, each object of the inner optimization can be managed by its dedicated processor.

2.2.1. Outer Optimizer

Searching for the global optimum and tackling possible narrow valleys are both jobs for the outer optimizer. The following strategy is proposed to manage the N objects: three objects are required to apply the OPTNOV philosophy whereas the remaining N - 3 objects are selected by using the same techniques employed in optimal experimental design (Buzzi-Ferraris, 1999; Buzzi-Ferraris and Manenti, 2009, 2010). The outer optimizer collects initial and arrival points for each inner object, and selects the two points among them that have the best performances. If these two points are very close to each other, the best third, fourth, and so on, is selected in spite of the second to avoid any ill-conditioning while detecting the valley direction.

Distances δ1, δ2, and δ3 between points can be reduced or expanded depending on the results: for example, if the point III results in a better inner optimum, distances are expanded. Points from the fourth to the Nth are selected so as to have the farthest points compared to all the collected ones. This selection is efficiently carried out using techniques adopted and proven for the optimal design of experiments.

The following procedure is adopted as the stop criterion. At each iteration, the number of points in the neighborhood of the optimum (given a tolerance value) is checked. If such a number is reasonable (according to an assigned value), a possible solution is reached. Theoretically, the number of points should be in the order of magnitude of the optimization problem dimensions, but it is preferable to use smaller numbers when the optimization size is large.

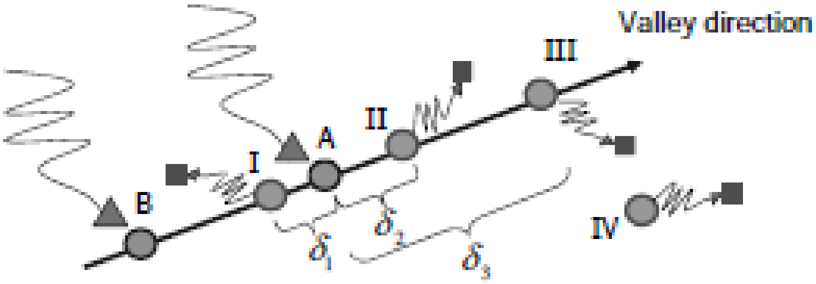


Figure 44: Graphical representation of the projections of the inner points as the outer optimizer evaluates them (OPTNOV-Simplex Method)

2.2.2. Inner Optimizer

The inner optimizer has the task of quickly finding a local minimum or a point on the bottom of the valley even when robustness is required. Here the Optnov method is adopted to improve the robustness of the Simplex method. The Optnov method iteratively selects a new starting point that is used by the Simplex method with a limited number of iterations.

The **BzzMinimizationRobust** (see following paragraph) class uses this hybrid Optnov–Simplex method. The hybrid Optnov–Simplex method has the following pros and cons.

1) The objective function does not have any special requirements: in fact, it can be non-derivable and discontinuous, and no quadratic functions can reasonably approximate it in the optimum.

2) It allows narrow valleys to be followed.

3) The function can be undefined in some regions and the domain cannot be analytically described.

4) It is inefficient with respect to other alternatives when robustness is not necessary.

5) The method can only be used for small- and medium-scale problems.

6) It does not effectively exploit parallel computing. Its computational performance does not improve significantly when several processors are available.

3. Optimization Tools

The algorithm described before can reach a very high degree of complexity and could be difficult to implement starting from scratch. Luckily, they are available (together with a great number of other ones related to different mathematic problems) in a pre-compiled and free library, called **BzzMath**, written in C++ language and in such a way to be as much user-friendly as possible. The Visual Studio C++ environment and the BzzMath library are now described.

3.1. The C++ Language

C++ is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation.

It was designed with a bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, web search or SQL servers), and performance-critical applications (e.g. telephone switches or space probes). C++ is a compiled language, with implementations of it available on many platforms. The C++ language has two main components: a direct mapping of hardware features provided primarily by the C subset, and zero-overhead abstractions based on those mappings.

A C++ program is a sequence of ASCII text files (typically header and source files) that contain declarations. They undergo translation to become an executable program, which is executed when the OS calls its main function. Certain words in a C++ program have special meaning, and these are known as keywords. Others can be used as identifiers. Comments are ignored during translation.

The entities of a C++ program are values, objects, references, functions, enumerators, types, class members, templates, template specializations, namespaces, parameter packs, etc. Preprocessor macros are not C++ entities. Entities are introduced by declarations, which associate them with names and define their properties. The declarations that define all properties required to use an entity are definitions. Definitions of functions include sequences of statements, some of which include expressions, which specify the computations to be performed by the program.

Names encountered in a program are associated with the declarations that introduced them using name lookup. Each name is only valid within a part of the program called its scope. Some names have linkage which makes them refer to the same entities when they appear in different scopes or translation units. Each object, reference, function, expression in C++ is associated with a type, which may be fundamental, compound, or user-defined, complete or incomplete, etc. Named objects and named references to objects are known as variables.

Functions are C++ entities that associate a sequence of statements (a function body) with a name and a list of zero or more function parameters.

A function can terminate by returning or by throwing an exception. A function declaration may appear in any scope, but a function definition may only appear in namespace scope or, for member and friend functions, in class scope. A function that is declared in a class body without a friend specifier is a class member function. Such functions have many additional properties,

Functions are not objects: there are no arrays of functions and functions cannot be passed by value or returned from other functions. Pointers and references to functions are allowed, and may be used where functions themselves cannot.

Besides function values, the function call expression supports pointers to functions, dereferenced pointers to member functions, lambda-expressions, and any variable of class type that overloads the function-call operator. Together, these types are known as FunctionObjects, and they are used ubiquitously through the C++ standard library, see for example, usages of BinaryPredicate and Compare.

The standard library also provides a number of pre-defined function object templates as well as the methods to compose new ones (including std::mem_fn, std::bind, and std::function). In the C++ programming language, the C++ Standard Library is a collection of classes and functions, which are written in the core language and part of the C++ ISO Standard itself.The C++ Standard Library provides several generic containers, functions to utilize and manipulate these containers, function objects, generic strings and streams (including interactive and file I/O), support for some language features, and functions for everyday tasks such as finding the square root of a number. The C++ Standard Library also incorporates 18 headers of the ISO C90 C standard library ending with ".h", but their use is deprecated. No other headers in the C++ Standard Library end in ".h". Features of the C++ Standard Library are declared within the std namespace.

A noteworthy feature of the C++ Standard Library is that it not only specifies the syntax and semantics of generic algorithms, but also places requirements on their performance. These performance requirements often correspond to a well-known algorithm, which is expected but not required to be used. In most cases this requires linear time O(n) or linearithmic time O(n log n), but in some cases higher bounds are allowed, such as quasilinear time O(n log2 n).

The main function is called at program startup after initialization of the non-local objects with static storage duration. It is the designated entry point to a program that is executed in hosted environment (that is, with an operating system). The parameters of the two-parameter form of the main function allow arbitrary multi-byte character strings to be passed from the execution environment (these are typically known as command line arguments), the pointers argv[1] .. argv[argc-1] point at the first characters in each of these strings. argv[0] is the pointer to the initial character of a null-terminated multi-byte strings that represents the name used to invoke the program itself (or an empty string "" if this is not supported by the execution environment). The strings are modifiable, although these modifications do not propagate back to the execution environment. The size of the array pointed to by argv is at least argc+1, and the last element, argv[argc], is guaranteed to be a null pointer.

The main function has several special properties:

I) It cannot be used anywhere in the program:

a) in particular, it cannot be called recursively;

b) its address cannot be taken;

II) It cannot be predefined and cannot be overloaded;

III) It cannot be defined as deleted or declared with C language linkage;

IV) The body of the main function does not need to contain the return statement: if control reaches the end of main without encountering a return statement, the effect is that of executing return 0;

V) Execution of the return (or the implicit return upon reaching the end of main) is equivalent to first leaving the function normally (which destroys the objects with automatic storage duration) and then calling `std::exit` with the same argument as the argument of the return. (`std::exit` then destroys static objects and terminates the program).

3.1.1. The BzzMath Library

Besides the Standard Library, C++ supports user-defined libraries that can be indicized and implemented by an IDE (Integrated Development Environment). An external library above which this whole work relies on is the BzzMath Library, developed by prf. Guido Buzzi-Ferraris at Politecnico di Milano. Preliminary activities started in 1964 with the implementation of linear and nonlinear system solvers, optimization and linear programming using procedural programming philosophy. The philosophy has radically changed in 1989, when the library coding was started in object-oriented programming with Turbo C++, allowing a significant improvement of numerical methods and easiness of implementation.

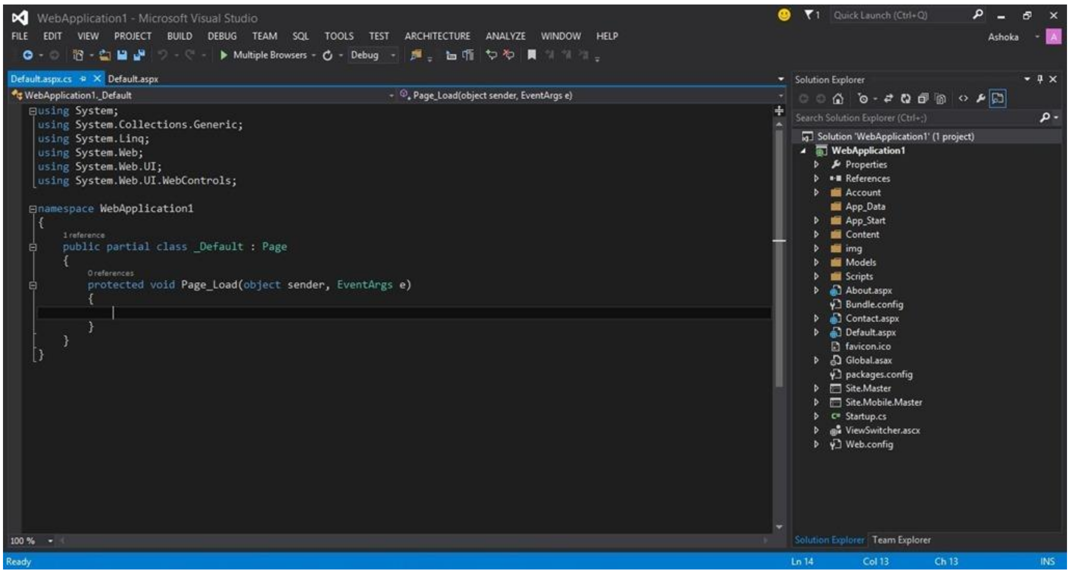
The library covers many fields of numerical analysis for Windows and Linux users, includes parallel computations automatically enabled when possible, and self-selects the best algorithms during computations. Among those covered fields, the one regarding the mono and multi-dimensional minimization is the one of most interest for the solution of the optimization problem related to the model presented in the chapter before: the **BzzMinimizationRobust** class will provide a particularly suitable algorithm.

3.2. Visual Studio

Microsoft Visual Studio is a suite of tools for creating software, from the planning phase through UI design, coding, testing, debugging, analyzing code quality and performance, deploying to customers, and gathering telemetry on usage. These tools are designed to work together as seamlessly as possible, and are all exposed through the Visual Studio Integrated Development Environment (IDE).

Although Visual Studio can be used to browse individual code files, more commonly the developer will be working on a project. A Visual Studio project is a collection of files and resources that are compiled to a single binary executable file for applications (for example, an .exe, DLL, or appx).

To build a project means to compile the source code and perform whatever steps are necessary to produce the executable. Different languages have different build operations, and regular websites don't build at all. Every compiler is completely configurable through the IDE. The output from the build, including an error or success messages, appears in the Output Window.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication1
{
    1 reference
    public partial class _Default : Page
    {
        0 references
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

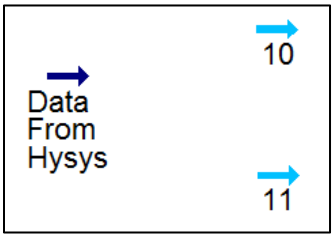
Figure 45: User interface of Visual Studio 2013

4. Optimization Algorithm

In order to actually find the best working conditions for the novel process model in exam, a suitable optimization algorithm has to be written and debugged. Since the solver for the minimization problem operates in the Visual Studio environment, the problem becomes the one of efficiently connecting Matlab (and therefore Hysys and OpenSMOKE++) to the IDE in which C++ code can be read. Integrating C++ to Matlab is a quite laborious operation that can be done in at least a couple of ways, exposed in the following paragraphs.

4.1. The MexFunction Method

In order to describe this procedure, a very simple example will be considered, like an exchange of important variables from a material stream of Hysys (eventually an energy stream too). As described in the figure below, the aim is to take information from the stream “Data From Hysys” that is totally defined (dark blue); elaborate them with the usage of C++ and back in Hysys to define streams “10” and “11” that are not complete at the moment (light blue). It’s important to say that with this procedure is possible to define only pure streams, not mixtures. In fact, in the example the stream “Data From Hysys” is a binary mixture and, after making calculations in C++, is split in stream “10” and “11”. The user must insert manually in the Hysys environment a composition value (i.e. molar or mass fraction) equal to 1 respectively for each component, one in stream “10” and the other in stream “11”.

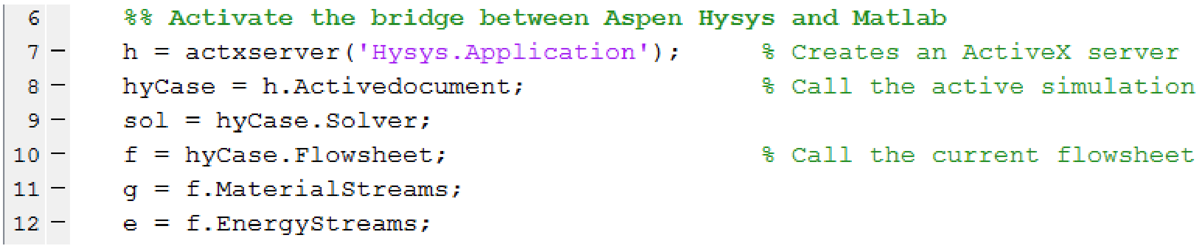


After this preliminary step of compiling the Aspen Hysys flowsheet is possible to start exchanging data between the software, the process will be divided in three different logic steps:

- 1 Step – Integration Aspen Hysys and Matlab
- 2 Step – Integration Matlab and C++
- 3 Step – Integration Matlab and Aspen Hysys

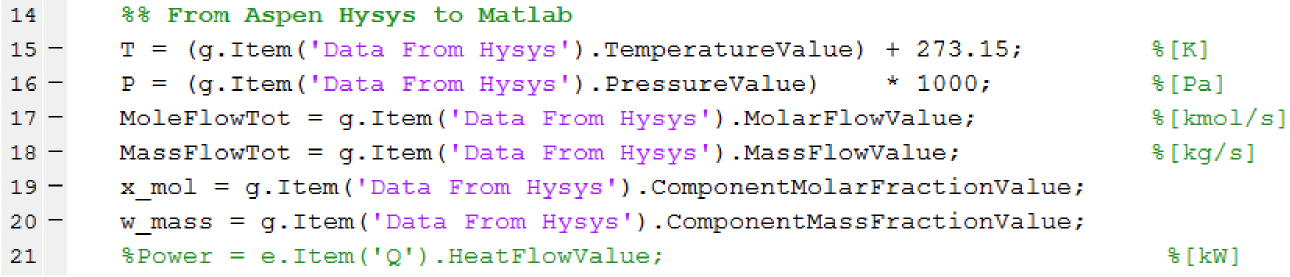
Caveat n.1: the procedure works only if only one Hysys flowsheet is opened.

The first step is to create a Matlab code. As mentioned Matlab is the global interface and to perform the exchange of data the user must run that specific Matlab code. Below there is an extract of the first part of the code, where a bridge between Hysys (the open flowsheet) and Matlab is created.



```
%% Activate the bridge between Aspen Hysys and Matlab
h = actxserver('Hysys.Application');      % Creates an ActiveX server
hyCase = h.Activedocument;                % Call the active simulation
sol = hyCase.Solver;
f = hyCase.Flowsheet;                     % Call the current flowsheet
g = f.MaterialStreams;
e = f.EnergyStreams;
```

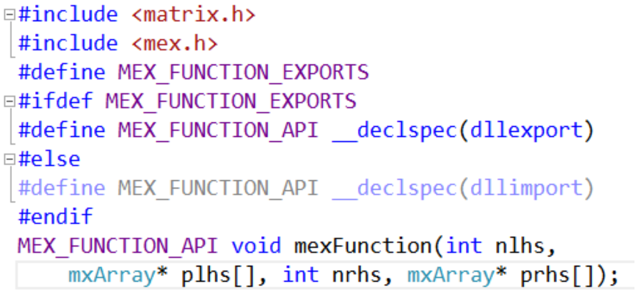
This is the moment to call from Hysys the desired variables belonging to a material or energy stream. In the present example only one material stream is present, so one possible command for the energy stream (i.e. heat flow) is commented as shown below.



```
%% From Aspen Hysys to Matlab
T = (g.Item('Data From Hysys').TemperatureValue) + 273.15;         %[K]
P = (g.Item('Data From Hysys').PressureValue)    * 1000;           %[Pa]
MoleFlowTot = g.Item('Data From Hysys').MolarFlowValue;            %[kmol/s]
MassFlowTot = g.Item('Data From Hysys').MassFlowValue;             %[kg/s]
x_mol = g.Item('Data From Hysys').ComponentMolarFractionValue;
w_mass = g.Item('Data From Hysys').ComponentMassFractionValue;
%Power = e.Item('Q').HeatFlowValue;                                  %[kW]
```

Now we leave the Matlab environment and open Microsoft Visual Studio. It will be shown at the end how to call the C++ code in Matlab script. It is worth to say that, as mentioned at the beginning, Matlab is the global user interface and so there is no the need to manually run the C++ code, it is done directly by the Matlab function. The following lines deal with the procedure to set up C++ for the final scope, that is to make calculations.

After proper initialization of the project, a suitable header must be created:



```
#include <matrix.h>
#include <mex.h>
#define MEX_FUNCTION_EXPORTS
#ifdef MEX_FUNCTION_EXPORTS
#define MEX_FUNCTION_API __declspec(dllexport)
#else
#define MEX_FUNCTION_API __declspec(dllimport)
#endif
MEX_FUNCTION_API void mexFunction(int nlhs,
    mxArray* plhs[], int nrhs, mxArray* prhs[]);
```

If necessary, additional libraries and dependencies must be located inside the working folder and their position specified inside Visual Studio

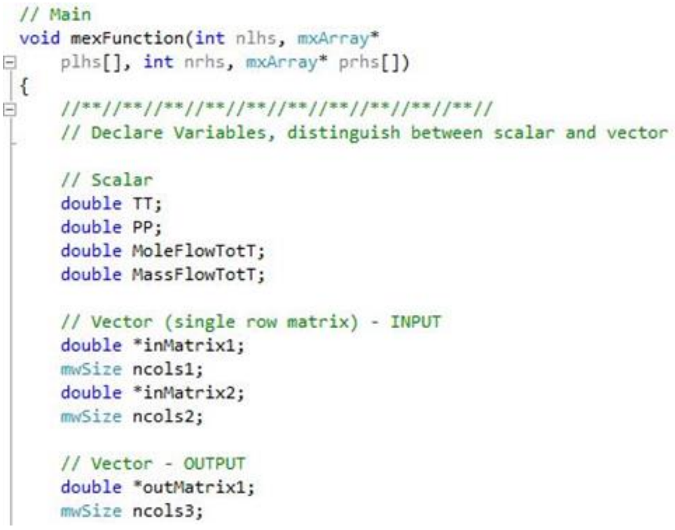
Once the required files are ready, we can start creating the code (i.e. the cpp file) that will read the input values from Matlab, perform the calculations and give back a certain number of outputs. This task is done by a function, called **mexFunction**, able to work with two vectors: "**prhs**" for the input elements and "**plhs**" for the output elements. In every position of those vectors is stored an object (that could be either a scalar, a vector or a matrix) which may come from Matlab, or created in the Visual Studio environment and must be transferred back.

In this example, the variables over which we want to make some computations are Temperature, Pressure, Molar and Mass Fluxes, Molar and Mass fractions of two components; so, 4 scalar values and 2 vectors. If we decide not to modify the mass flux and fractions, we are facing this situation:

$$[plhs] = \ mexFunction[prhs]$$

$$[Tnew\ Pnew\ MolarFluxNew\ x\_mol\_new\ ] \ = \ mexFunction[T\ P\ MolarFlux\ MassFlux\ xmol\ wmass]$$

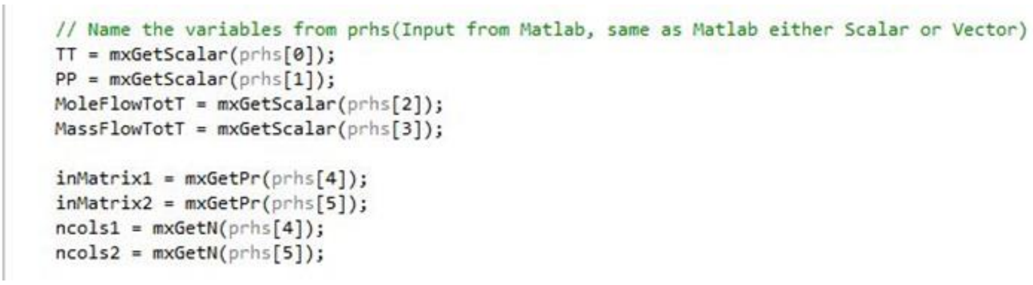
The concept behind the **mexFunction** is the same of the script that serves as "main" when working with Matlab as we can define the variables, perform computations and, ultimately, run it to perform a well-defined routine. After initializing the function and its content, therefore, the first thing to do is set up the input and output variables in the same form they appear in Matlab:



```
// Main
void mexFunction(int nlhs, mxArray*
    plhs[], int nrhs, mxArray* prhs[])
{
    //**//**//**//**//**//**//**//**//**//**//
    // Declare Variables, distinguish between scalar and vector

    // Scalar
    double TT;
    double PP;
    double MoleFlowTotT;
    double MassFlowTotT;

    // Vector (single row matrix) - INPUT
    double *inMatrix1;
    mwSize ncols1;
    double *inMatrix2;
    mwSize ncols2;

    // Vector - OUTPUT
    double *outMatrix1;
    mwSize ncols3;
```

Declaration of the scalar variables is straightforward once we know the kind of precision we are dealing with (integer, double, etc.). The four ones listed above represent the input values; initialization of the scalar outputs is done as we compute them.

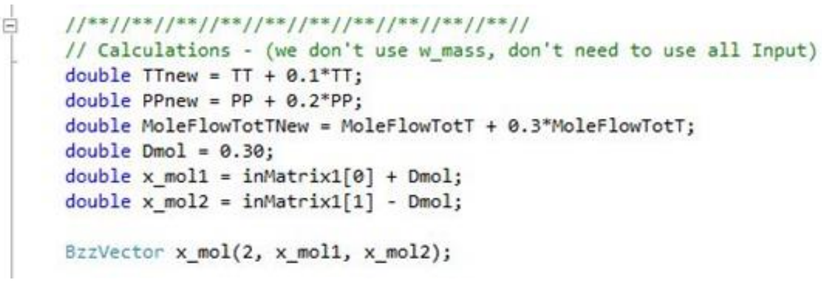
Arrays must be initialized at the very beginning of the function too. In this example, we expect to have 2 input vectors and 1 output vector so we define them by means of a pointer to the first element (e.g. “*inMatrix1”, * is the dereference operator) and their size. It is not necessary to specify yet the exact length of the vector, as it will be done later.



```
// Name the variables from prhs(Input from Matlab, same as Matlab either Scalar or Vector)
TT = mxGetScalar(prhs[0]);
PP = mxGetScalar(prhs[1]);
MoleFlowTotT = mxGetScalar(prhs[2]);
MassFlowTotT = mxGetScalar(prhs[3]);

inMatrix1 = mxGetPr(prhs[4]);
inMatrix2 = mxGetPr(prhs[5]);
ncols1 = mxGetN(prhs[4]);
ncols2 = mxGetN(prhs[5]);
```

To complete the initialization of our variables we need of course to supply them a value (or a set of values for an array): this is done accessing the input vector **prhs** through suitable commands like **mxGetScalar** or **mxGetPr**. The first one points to a real element of the **mxArray** and returns a double; while the second one points to the first value of a vector whose length is specified by the **mxGetN** command.

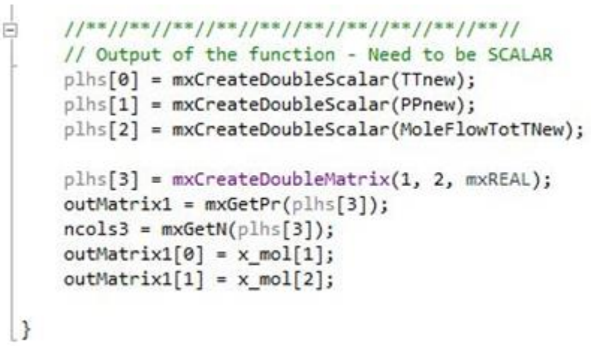
We are ready to perform all of the calculations we need.



```
//**//**//**//**//**//**//**//**//**//**//
// Calculations - (we don't use w_mass, don't need to use all Input)
double TTnew = TT + 0.1*TT;
double PPnew = PP + 0.2*PP;
double MoleFlowTotTNew = MoleFlowTotT + 0.3*MoleFlowTotT;
double Dmol = 0.30;
double x_mol1 = inMatrix1[0] + Dmol;
double x_mol2 = inMatrix1[1] - Dmol;

BzzVector x_mol(2, x_mol1, x_mol2);
```

It can be seen above how the output scalars are declared and computed at the same time. As for the output vector, it can of course be the results of different operations (for cycle, dot product, etc.), this time we simply chose to store in it the results of a sum and a difference, and to use an object called **BzzVector**, contained in the **BzzMath** library, just to show how to convert it into a vector of values of type double, as it is required by the mexFunction: **plhs** is, indeed, a vector of doubles. This operation turns out to be very useful if the solution of a problem goes through specific algorithms of the BzzMath library, like BzzOdeNonStiff and so on.

To conclude the routine, we have to assign every output to its position in the **plhs** vector so that they are transferred back to Matlab. We can specify which kind of element is going to be stored in a certain slot thanks to **mxCreateDoubleScalar** and **mxCreateDoubleMatrix** commands (basically the reverse of **mxGetScalar** and so on).



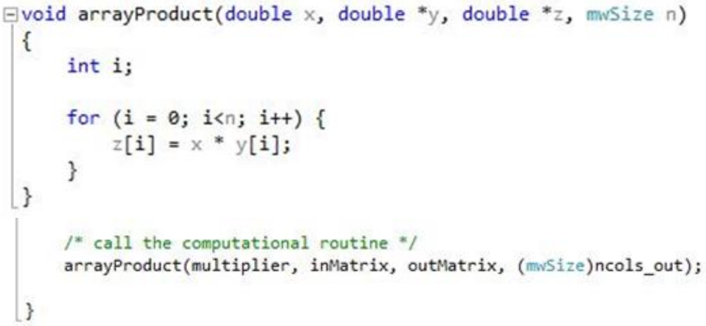
```
    //**//**//**//**//**//**//**//**//**//**//
    // Output of the function - Need to be SCALAR
    plhs[0] = mxCreateDoubleScalar(TTnew);
    plhs[1] = mxCreateDoubleScalar(PPnew);
    plhs[2] = mxCreateDoubleScalar(MoleFlowTotTNew);

    plhs[3] = mxCreateDoubleMatrix(1, 2, mxREAL);
    outMatrix1 = mxGetPr(plhs[3]);
    ncols3 = mxGetN(plhs[3]);
    outMatrix1[0] = x_mol[1];
    outMatrix1[1] = x_mol[2];

}
```

The first one is very simple and does not require further explanation. As for the second, after assigning the number of rows and columns and the kind of values (real vs complex) of the matrix, we operate the same way we did with the input ones, using **mxGetPr** to access the real data pointed to by its argument and **mxGetN** to determine the number of columns, giving a body to this initially empty array. To fill it with the same elements stored in the **BzzVector** (thus converting it to a doubles one) we need to pass them one by one.

Caveat n.4: every vector or matrix that goes back to Matlab has to be filled with values **<u>after</u>** its creation; so, if we have computations to do, they must be relegated to the final part of the code. From the point of view of the **mexFunction** not doing this procedure does not represent an error and is not signalled, nevertheless running it in Matlab will cause the program to crash.

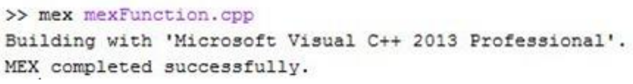
Caveat n.5: **BzzVector** starts enumerating its slots from the 1 position, while a vector of doubles starts from the 0 position; the correspondence between the two requires attention.

The last remark before returning to the Matlab environment is related to the use of subroutines: mexFunction works like a main script, so it can be used to call other algorithms that perform specific calculations. As an example, a code that performs a dot product and its caller at the end of mexFunction are reported:



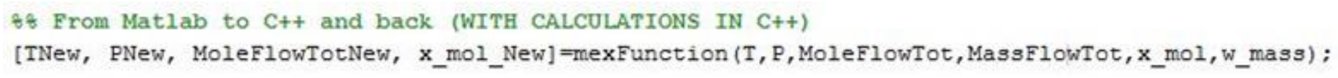
```
void arrayProduct(double x, double *y, double *z, mwSize n)
{
    int i;

    for (i = 0; i<n; i++) {
        z[i] = x * y[i];
    }
}

    /* call the computational routine */
    arrayProduct(multiplier, inMatrix, outMatrix, (mwSize)ncols_out);

}
```

We are headed back to Matlab. A different programming language cannot read the file **mexFunction.cpp** as is: the next step is the creation of a .mexw64 file that Matlab can use to communicate with the C++ code. The command "**mex –setup**" will allow us to choose the compilation language for this operation, for example Microsoft Visual C++ 2013 Professional or whatever version is installed in the personal computer. Then, running "**mex mexFunction.cpp**" will build the desired MEX file.



```
>> mex mexFunction.cpp
Building with 'Microsoft Visual C++ 2013 Professional'.
MEX completed successfully.
```

The last operation is, finally, writing a line in the script that starts this whole routine: the structure is exactly the one we have seen at the very beginning of this chapter.



```
%% From Matlab to C++ and back (WITH CALCULATIONS IN C++)
[TNew, PNew, MoleFlowTotNew, x_mol_New]=mexFunction(T,P,MoleFlowTot,MassFlowTot,x_mol,w_mass);
```

4.2. Matlab Compiler SDK Method

MATLAB Compiler SDK extends the functionality of MATLAB Compiler to build C/C++ shared libraries, Microsoft.NET assemblies, and Java classes from MATLAB programs. These components can be integrated with custom applications and then deployed to desktop, web, and enterprise systems.

Integrating compiled MATLAB functions into a C or C++ application requires the use of a combination of APIs (Application Programming Interfaces). MATLAB Compiler SDK uses APIs to initialize the MATLAB Runtime, load the compiled MATLAB functions into the MATLAB Runtime, and manage data that is passed between the C or C++ code and the MATLAB Runtime. The compiler generates some of the APIs based on the signatures of the compiled functions. MATLAB Runtime provides other APIs that are consistent for all applications.

The C++ library wrapper option allows to create a shared library from an arbitrary set of MATLAB files. MATLAB Compiler SDK generates a wrapper file and a header file. The header file contains all of the entry points for all of the compiled MATLAB functions. The following example writes a function wrapper for the functions **`addmatrix.m`**, **`multiplymatrix.m`** and **`eigmatrix.m`** in the form of a shared library.

In Matlab, enter the following command on a single line:

```
mcc -W cpplib:libmatrixp -T link:lib addmatrix.m multiplymatrix.m eigmatrix.m -v
```

The -`W cpplib:<libname>` option tells MATLAB Compiler SDK to generate a function wrapper for a shared library and call it `<libname>.` The -`T link:lib` option specifies the target output as a shared library. Note the directory where the product puts the shared library because will be needed later. For each MATLAB file specified on the MATLAB Compiler SDK command line, the product can generate two functions, the `mlx` function and the `mlf` function. Each of these generated functions performs the same action (calls the MATLAB file function). The two functions have different names and present different interfaces. The name of each function is based on the name of the first function in the MATLAB file (*addmatrix*, in this example); and the type depends on the command used: `mcc` generates a `mlf` function. This means that his function expects its input and output arguments to be passed in as individual variables rather than packed into arrays. If the function is capable of producing one or more outputs, the first argument is the number of outputs requested by the caller:

```
void Addmatrix(int nargout, mxArray** x, mxArray** y, mxArray* iterations,
mxArray* draw)
```

```
Addmatrix(1, &out, in1, in2)
```

In this call, the caller requests one output argument, the `out` vector, and provides two inputs, `in1` and `in2` vectors.

If the output variables you pass in to an `mlf` function are not NULL, the `mlf` function will attempt to free them using **`mxDestroyArray`**. This means that one can reuse output variables in consecutive calls to `mlf` functions without worrying about memory leaks. It also implies that one must pass either NULL or a valid MATLAB array for all output variables or your program will fail because the memory manager cannot distinguish between a non-initialized (invalid) array pointer and a valid array. It will try to free a pointer that is not NULL -- freeing an invalid pointer usually causes a segmentation fault or similar fatal error.

To use a MATLAB Compiler SDK generated shared library in a Visual Studio project:

1) Include the generated header file for each library in the application. Each generated shared library has an associated header file named libname.h.
2) Initialize the MATLAB Runtime proxy layer by calling **`mclmcrInitialize().`**
3) Use **`mclRunMain()`** to call the C function where your MATLAB functions are used. **`mclRunMain()`** provides a convenient cross platform mechanism for wrapping the execution of MATLAB code.
4) Initialize the MATLAB Runtime and set the global settings by calling **`mclInitializeApplication()`** API function. Call the **`mclInitializeApplication()`** function once per application, and it must be called before calling any other MATLAB API functions. It is possible to pass in application-level options to this function. **`mclInitializeApplication()`** returns a Boolean status code.
5) For each MATLAB Compiler SDK generated shared library that you include in the application, call the initialization function for the library. The initialization function performs library-local initialization. It unpacks the deployable archive and starts a MATLAB Runtime instance with the necessary information to execute the code in that archive. The library initialization function is named **`libnameInitialize().`** This function returns a Boolean status code.
6) Call the exported functions of each library as needed.
7) When the application no longer needs a given library, call the termination function for the library. The terminate function frees the resources associated with the libraries MATLAB Runtime instance. The library termination function is named **`libnameTerminate().`** Once a library has been terminated, the functions exported by the library cannot be called again in the application.
8) When the application no longer needs to call any MATLAB Compiler SDK generated libraries, call the **`mclTerminateApplication`** API function. This function frees application-level resources used by the MATLAB Runtime. Once you call this function, no further calls can be made to MATLAB Compiler SDK generated libraries in the application.

The following C++ code example calls the Admmatrix.m function after its export in the form of a shared library:

4.3. Method Selection Criteria

It is already apparent, from the presentations done in the chapters before, that both of the two methods offer feasible ways to connect functions written in Matlab with code that runs in a C++ environment. The basis upon which they are built are quite different and, in the end, make one particular method more attractive than the other.

The MexFunction method turns out to be useful when it is necessary to connect in series a Matlab and a C++ function: More in detail, it allows to call from Matlab a function defined in C++ after the latter is provided the tools to successfully convert data from a format to another.

The Matlab Complier SDK method, instead, allows to "wrap" the function written in Matlab in a shared library form that can be read and included by Visual Studio in a C++ function. This very peculiar optimization problem requires the BzzMath optimizer to interpret the output of an objective function and find the best direction in order to efficiently bring it to a minimum value: it is then more appropriate for the algorithm to be able to call the process model directly from the optimizer since its complexity reduces and the performances increase.

The following scheme will help in visualizing the relative complexity of the algorithm in terms of flow of data and make it obvious why the choice fell upon the second method.

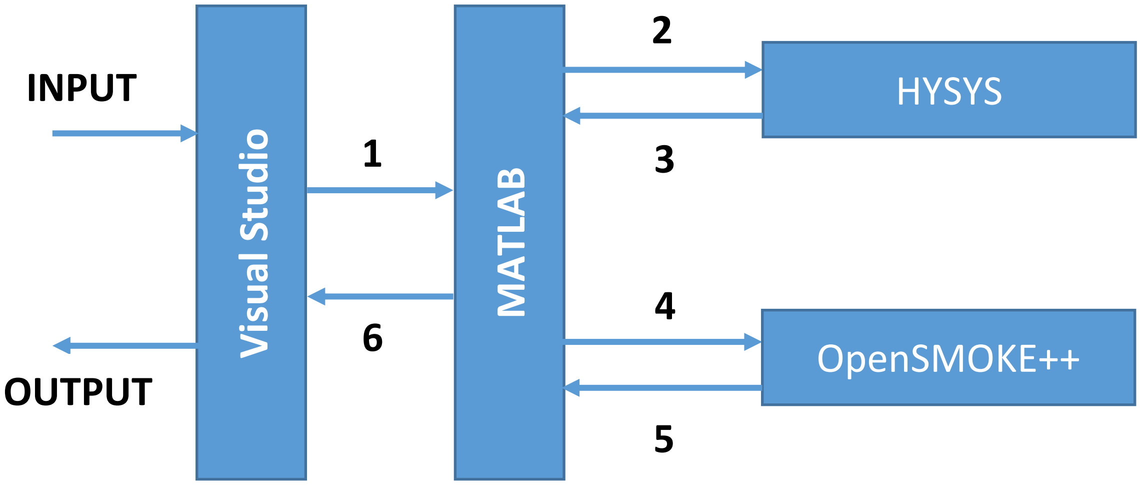


Figure 46: Data flow diagram of the algorithm implementing the Matlab Compiler SDK method

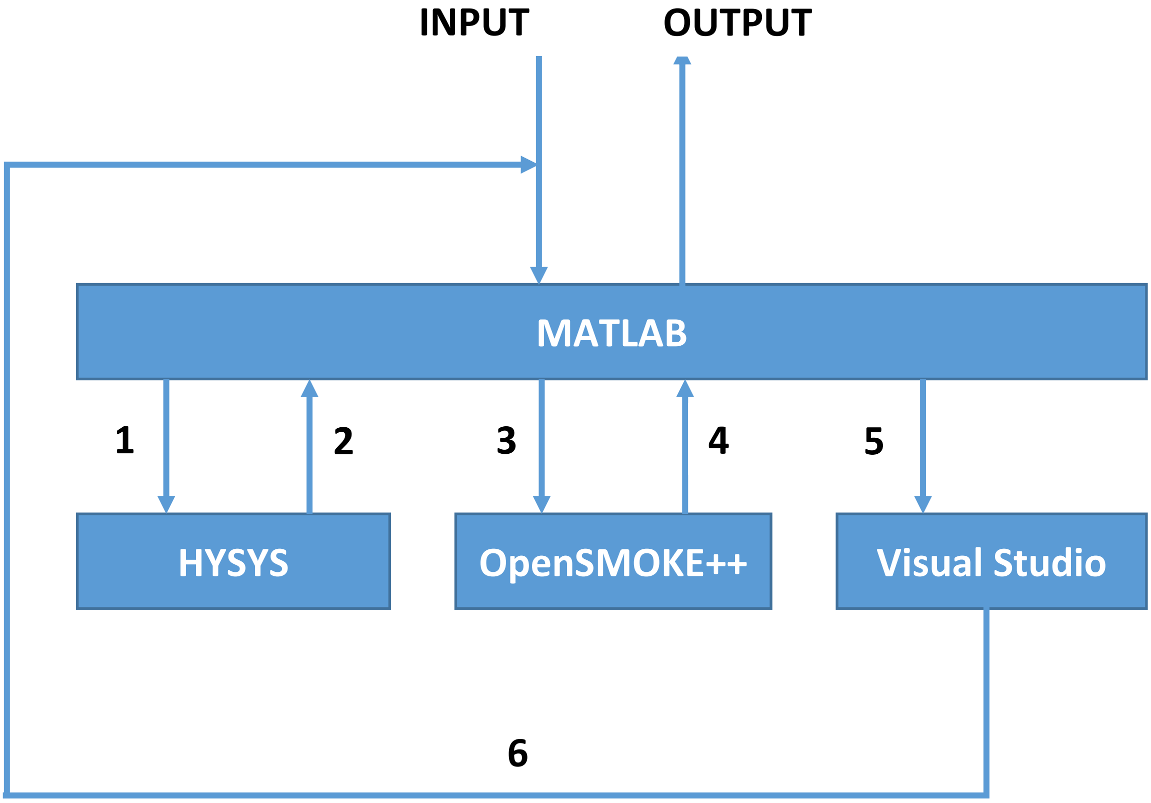


Figure 47: Data flow diagram of the algorithm implementing the MexFunction method

In Fig.46, INPUT represents the first guess values for the optimization while 6 and OUTPUT are respectively the objective function results and the solution of the optimization problem. The objective function in Fig. 47 is instead represented by stream 5.

It can be noted that the robust optimizer in the selected method runs at the highest possible algorithm level, thus avoiding a recycle of informations that unnecessarily complicates the whole structure. Furthermore, Matlab's importance in the architecture is comparable to Visual Studio's one only in the first method, as it rightfully should be, while in the second it is left to Matlab (which is not the environment where the optimizer operates) to control and dictate the performance of the process.

Everything that has been stated above contributes in selecting the architecture depicted in Fig. 46 as the one to be built in order to solve this optimization problem.

PART 4 – OPTIMIZATION PROBLEM: DEFINITION AND SOLUTION

Upon building an algorithm capable of taking on very complex and largely undefined (in terms of explicit model equations) optimization problems, the next natural step is the determination of the very function to be minimized. This means being able to write an objective function that represents the parameter (or the parameters) of interest in terms of one or more independent variables.

The first part of this chapter will deal with the selection of those variables between the ones with respect to which the novel process configuration shows greatest sensivity and then writing a suitable objective function to be the subject of exam by the robust optimizer.

1. Choice of the Optimization Variables

Literature search provided some fundamental inputs for the individuation of interesting process variables from an optimization point of view.

Groisil et al. [5] conducted an experimental research in a well established facility with the aim to examine the pyrolysis of diluted and non-diluted hydrogen sulfide at higher reactor temperatures (in the range of 1273-1573 K), studying the conversion of high concentration hydrogen sulfide stream to produce hydrogen and Sulfur at higher temperatures. The optimum operating conditions that favor hydrogen production and high conversion of hydrogen sulfide were then investigated: the results showed the significant role of reactor residence time and reactor temperature, as well as inlet concentration of hydrogen sulfide injected into the reactor. A high reactor temperature significantly reduces the reactor residence time required for high conversion of hydrogen sulfide. At temperatures above 1373 K, conversion of hydrogen sulfide reached an asymptotic steady state value at a residence time of approximately 0.8s, while more than 1.2s residence time was required to attain such a steady state asymptotic value at 1273 K.

Increase in the inlet concentration of hydrogen sulfide enhanced the radical pool species and stimulated recombination reactions that hindered hydrogen production and hydrogen sulfide conversion. This was apparent from the increased deviation between equilibrium and experimental data as the inlet concentration of hydrogen sulfide and temperature of the reactor were increased. The results provided here assists in alternative means of acid gas treatment with simultaneous hydrogen and Sulfur production as well as mitigation of health and environment issues.

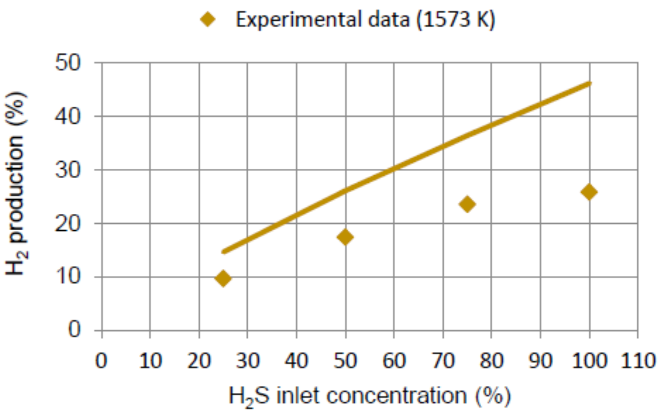


Figure 48: H_2 production with change in acid gas composition (H_2S diluted in N_2) at 1573K reactor temperature. [9]

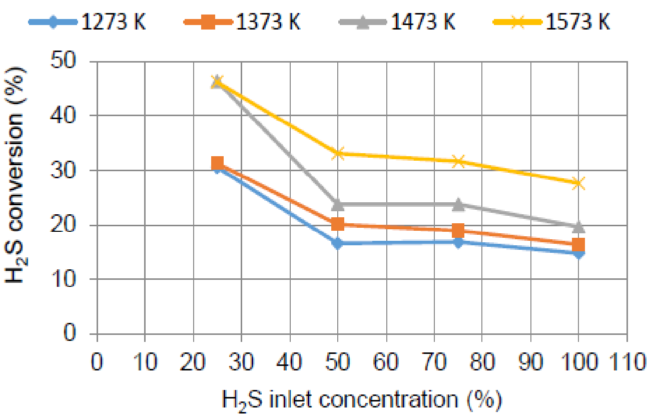


Figure 49: H_2S conversion with change in acid gas composition (H_2S diluted in N_2). [9]

As useful as the gathered information may be, it is not to be forgotten that very often CO_2 is present in acid gas streams out of tail refinery units together with H_2S and that the focus of the AG2S™ configuration is to exploit the production of not only hydrogen but also CO, in other words syngas

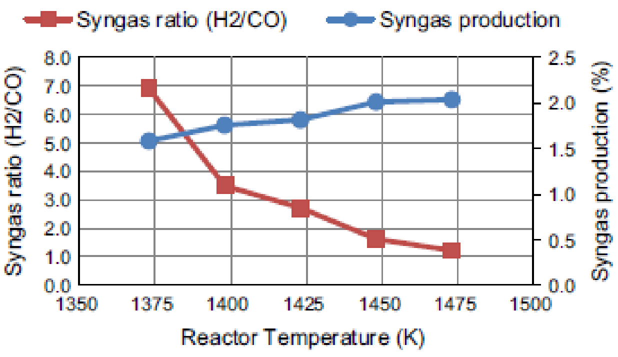


Figure 50: Effect of reactor temperature on syngas production (3% H_2S/2% CO_2 diluted in 95% N_2). [6]

A more specific investigation in this sense was done by El Melih et al. [6]. Experimental examination of the production of syngas from acid gas stream is not available in the open literature even if acid gas thermal decomposition is well suited for the treatment of lean acid gas that poses serious operational issues in Claus process plants. Presence of impurities, such as hydrocarbons in acid gas will also be an added value since this will favor higher yield of syngas.

Pyrolysis of acid gas mixture was examined in the plug flow reactor over a temperature range of 1323–1473 K at high residence times of 2 s. The objective was to examine the effect of temperature on the syngas (H_2 and CO) composition. These results are obtained from a mixture with 3% H_2S/2% CO_2/95% N_2, which represents a 60% H_2S / 40% CO_2 (H_2S/CO_2 ratio equal to 1.5) acid gas mixture since N_2 is inert. The ratio of H_2 to CO is observed to reduce while the syngas production increases with increase in reactor temperature. Production of CO was very minimal at temperatures below 1373 K. A reactor temperature less than 1373 K favors syngas yields with high H_2 content. If the requirement is to utilize only the produced H_2. It is nevertheless important to also note that lower reactor temperatures results in higher amounts of H_2S emission from low conversion of both H_2S and CO_2, even when sufficient residence time is available. The effect of inlet composition of acid gas (H_2S/CO_2 ratio) on the composition of syngas (H_2 and CO) was also examined at a reactor temperature of 1473 K and residence time of 2 s. Fig. 51 shows the composition of syngas evolved with change in the inlet composition of acid gas (H_2S and CO_2), diluted in N_2 gas at reactor temperature of 1473 K and residence time of 2 s. It also shows the amount of syngas produced with the variation in inlet composition of the acid gas. From Fig. 51, one can see that the ratio of H_2 to CO production is dependent on the initial acid gas composition (H_2S and CO_2). This ratio decreases with increase in the amounts of CO_2 in the inlet acid gas. This demonstrates that lean acid gas (with high

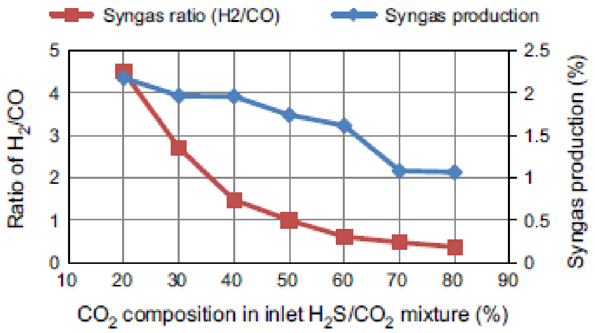


Figure 51: Effect of acid gas composition on syngas production at 1475 K (H_2S/CO_2 diluted in 95% N2). [6]

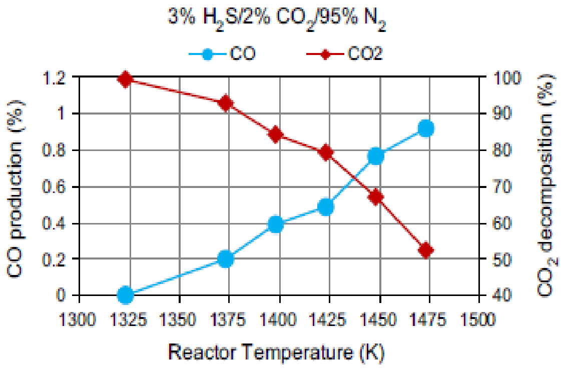


Figure 52: CO production and CO_2 decomposition with change in reactor temperatures (H_2S/CO_2 diluted in 95% N_2). [6]

CO_2 content) is preferable and well suited for syngas production with a H_2 to CO ratio of 0.3–1.ratio.

This can help ease the operational difficulties that lean acid gases pose in the operation of Claus process. Note that syngas production decreases with the increase in CO_2 amounts in the inlet acid gas stream due to high stability of CO_2.

Again Groisil et al. [8], focused their attention on the possible production of syngas from SRUs, of course examining the pyrolysis of H_2S and CO_2. The composition of syngas produced was examined over a temperature range of 1000-2000°C and residence time of 1.4 seconds. At temperatures below 1500°C, the conversion of H_2S was less than 60% for a given acid gas composition (see figure 51). Significant difference in H_2S conversion was much apparent at higher temperatures above 1800°C, but these temperatures promoted significant amounts of unwanted Sulfur compounds, such as SO_2 and COS. Moreover, the conversion of CO_2 was also insignificant at temperatures below 1500° C.

Therefore, temperature range of 1500-1800° C was most suitable for examining the composition of syngas produced. It is desirable to produce syngas with wide range of H_2 to CO ratios so that the produced syngas can be used in gas engines or ammonia and liquid fuels production. The results revealed that the ratio of H_2 to CO decreases with increase in temperature, and only high temperature reaction, above 1550°C and specific acid gas composition can allow the production of usable syngas in ammonia production and gas engine applications. Acid gas with high H_2S content, more than 60%, produced syngas that was very rich in H_2.

The results show that high temperatures are more suitable for acid gas pyrolysis. However, some undesirable compounds can also be produced during the pyrolysis. The evolution of these compounds with temperature should therefore be explored. Figures 53 and 54 show the evolution of COS and SO_2. The results show the production of COS and SO_2 is favored at temperatures above 1600°C. However, the formed S_2 has the tendency to oxidize at higher temperatures due to increased decomposition of CO_2 to release oxygen. Sulfur compound can not be used if consistent formation of SO_2 and COS happens, since the latter must be removed due to their environmental and health hazards. Therefore, the ultimate ideal goal would be to maximize conversion of H_2S, produce S_2 and syngas with a suitable H_2 to CO ratio that meets the specific industrial requirement, while minimizing the formation of other unwanted compounds.

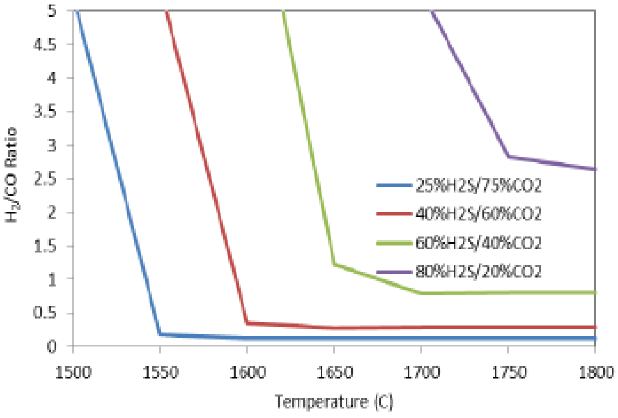


Figure 53: Ratio of H_2/CO for different composition of acid gas [8]

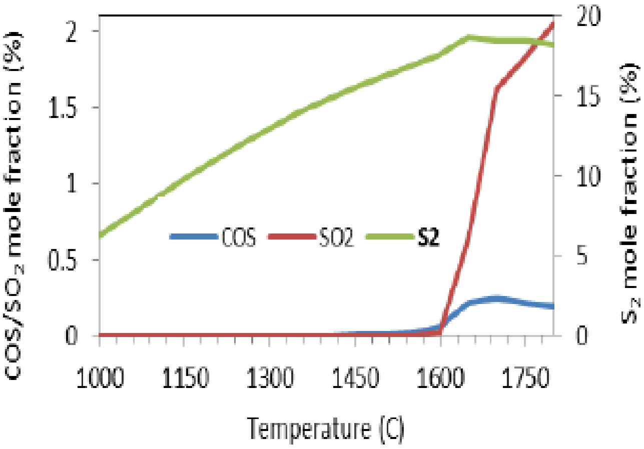


Figure 54: Mole Fractions of Sulfur and other Sulfur compounds for 60%H_2S-40% CO_2 acid gas [8]

It seems quite obvious that the governing parameters for the performance of the RTR in the AG2S™ process (in terms of selectivity towards syngas) seem to be the maximum temperature reached inside the combustion chamber and the H_2S/CO_2 inlet ratio. While implementation of the second one as independent variable for the objective function is straight-forward, the same does not apply for temperature, which itself is a dependent variable: its value will be inferred by means of changing the inlet oxidizer flow rate (in this case it is going to be pure O_2).

2. Definition of the Objective Function

The purpose of the AG2S™ process, and the principal aim of this work, is to create an attractive alternative to the well-known and widely adopted Claus configuration focusing on the production of syngas instead of the less valuable elemental Sulphur. Investigations carried out in the previous chapters reveal, nevertheless, that in order to efficiently switch from a process solution to another a change in the operative conditions of the reactor is needed, as the classic Claus furnace is not suited to obtain adequate quantities of CO and H_2, whichever their ratio may be.

2.1. Unconstrained Optimization

The yield of syngas has then to be the objective function through which it will possible to examine how the change in the independent variable is affecting the operation of the RTR. The mathematical problem can be expressed in this way:

$$\max_{((\frac{H_2S}{CO_2})_{IN}\,,\,\dot{m}\,O_{2,IN}} \frac{\dot{m}\,(H_2+CO)_{OUT}}{\dot{m}\,(H_2S{+}CO_2)_{IN}}$$

The previous expression is completely equivalent to the following one, suited to be implemented in the optimization algorithm (based on the **BzzMinimizationRobust** class of functions):

$$\min_{((\frac{H_2S}{CO_2})_{IN}\,,\,\dot{m}\,O_{2,IN}} -\frac{\dot{m}\,(H_2+CO)_{OUT}}{\dot{m}\,(H_2S{+}CO_2)_{IN}}$$

The complexity of the problem is now evident: while the objective function in somewhat explicit in the first independent variable (acid gases inlet ratio), this absolutely doesn't hold true for the second one. The latter dependency is highly non-linear and deeply affects the performance of the solver.

2.2. Constrained Optimization

More often than not, an industrial process presents constraints of various nature (economic, material, etc.) that limit the number of degrees of freedom taken into account while evaluating the best operative conditions among all the possible ones. In this work two different constraints are considered: the first one related to efficiency of the catalytic zone (dependent on the H_2S/SO_2 ratio exiting from the furnace, that has to be close to 2) that follows the RTR and the second one to the maximum temperature reachable inside the combustion chamber (for corrosion reasons it is advisable not to go beyond 1300°C).

The easiest way to impose constraints upon an objective function is to express them in the form of penalty functions; i.e. to sum the absolute value of the difference between the variable and its constraint value (of course this does not apply to inequalities) to the function that is being minimized. In the two cases examined, then, the objective functions will have an expression of the same kind:

$$\min_{((\frac{H_2S}{CO_2})_{IN}\,,\,\dot{m}\,O_{2,IN}} -\frac{\dot{m}\,(H_2+CO)_{OUT}}{\dot{m}\,(H_2S{+}CO_2)_{IN}} + |\frac{x_{H_2S}^{OUT}}{x_{SO_2}^{OUT}} - 2|$$

$$\min_{((\frac{H_2S}{CO_2})_{IN}\,,\,\dot{m}\,O_{2,IN}} -\frac{\dot{m}\,(H_2+CO)_{OUT}}{\dot{m}\,(H_2S{+}CO_2)_{IN}} + |T_{MAX} - 1573|$$

3. AG2S™ Unconstrained Optimization Results

In a completely aprioristic way, based exclusively on the literature data, one can expect a certain type of results stemming from an analysis of a process of this kind. More precisely, it has been found [7] that an increased reaction temperature favors syngas production at the expenses of H_2S, while H_2/CO ratio dramatically falls down. High-temperature pyrolysis also favors the formation of undesirable compounds like COS and CS_2.

The great stability of CO_2 represents an enemy of syngas production; so higher values of the H_2S/CO_2 inlet ratio are expected; anyway, hydrogen sulfide conversion obviously decreases as its initial concentration grows: the results will also show which factor carries the biggest weight for a successful operation of the process.

3.1. Shiraz: Optimized Process

The robust optimizer finds a maximum in syngas yield for an inlet mixture presenting a massive ratio between H_2S and CO_2 equal to 1.5 and an O_2 flow rate of 4157 kg/h (17% of oxygen in the inlet mixture), the latter able to generate a maximum temperature in the combustion chamber equal to 1360°C. Those result are in perfect agreement with the ones obtained by El Melih and reported in the previous chapter, obtained studying a mixture which presents the same acid gas ratio at the inlet. This large value (with respect to the initial one) favors hydrogen formation from H_2S: it is important to produce it in quantities at large as possible at the very beginning of the combustion chamber since H_2 eventually recombines with elementary Sulphur following the backwards reaction towards hydrogen sulfide.

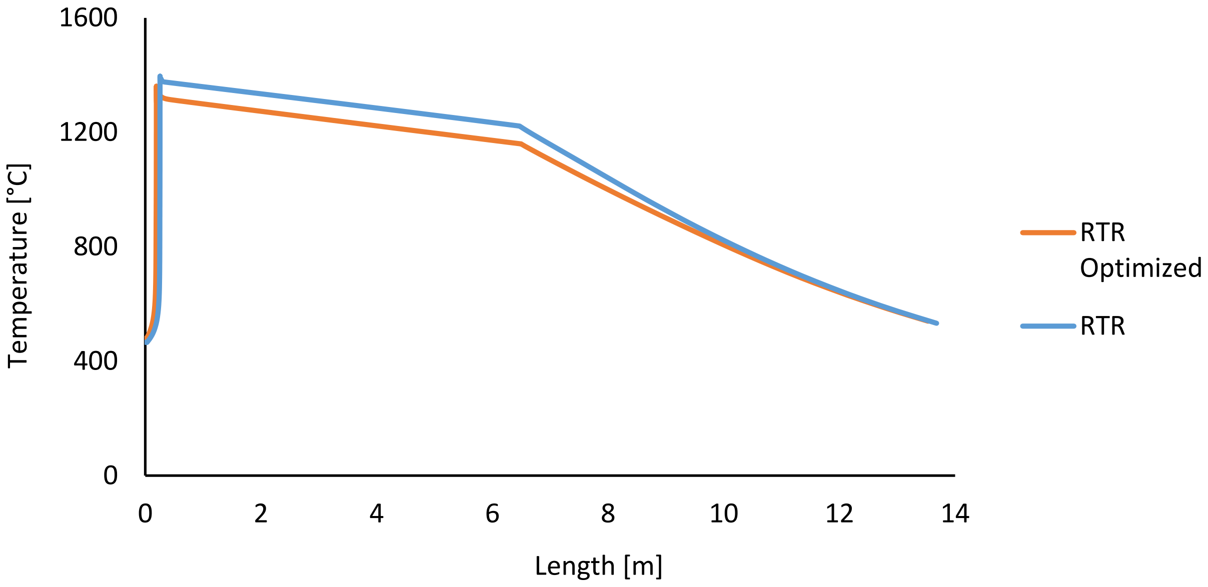


Figure 55: Shiraz RTR, optimized vs non-optimized temperature profile

Fig. 50 and 51 show respectively the syngas production (in terms of H_2 and CO molar fraction in the outlet stream) varying the amount of CO_2 at the inlet and the profiles for CO production and CO_2 conversion against temperature. It is clear that a maximum of 2% syngas is obtained at low to moderate carbon dioxide concentrations (20-30%): the optimal ratio found by the BzzMath solver corresponds to a molar fraction of CO_2 around 25%.

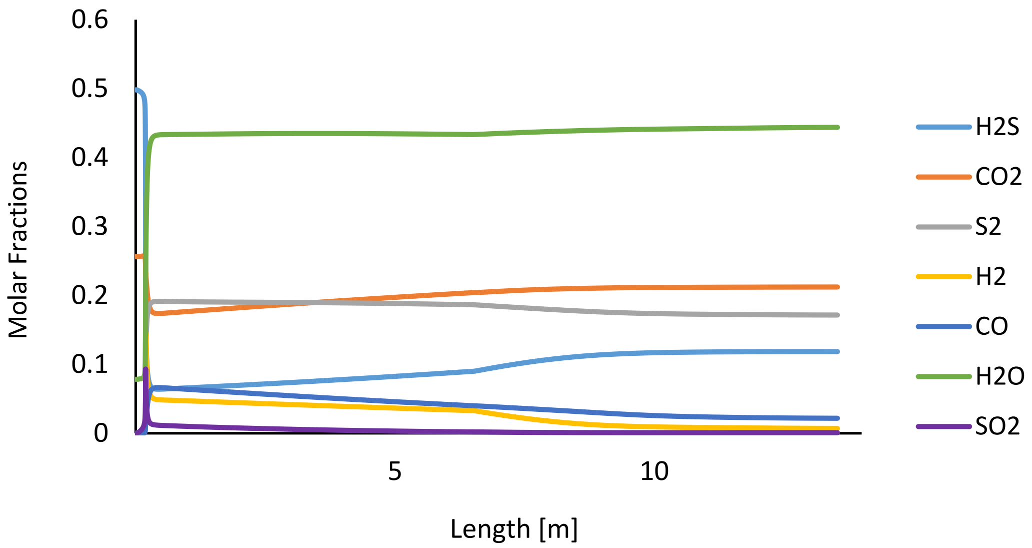


Figure 56: Shiraz RTR, optimized composition profile

The maximum temperature here obtained (1630 K) is not reported in figure 51 , but from the trend one can expect it will bring very low value of CO_2 conversion (16% in this case, in massive terms) and cause the production of CO in amounts larger of 1% of the total mixture (1.8% of the outlet mixture). The increment in the oxidizer flow rate is almost negligible with respect of its original value (less than 2%); the presence of H_2S in large quantities is enough to promote as much as possible the endothermic CO_2 decomposition via pyrolysis, generating radical species such as H, SO and S [13]. As a result said maximum temperature is slightly lower than the one reached in the non-optimized case.

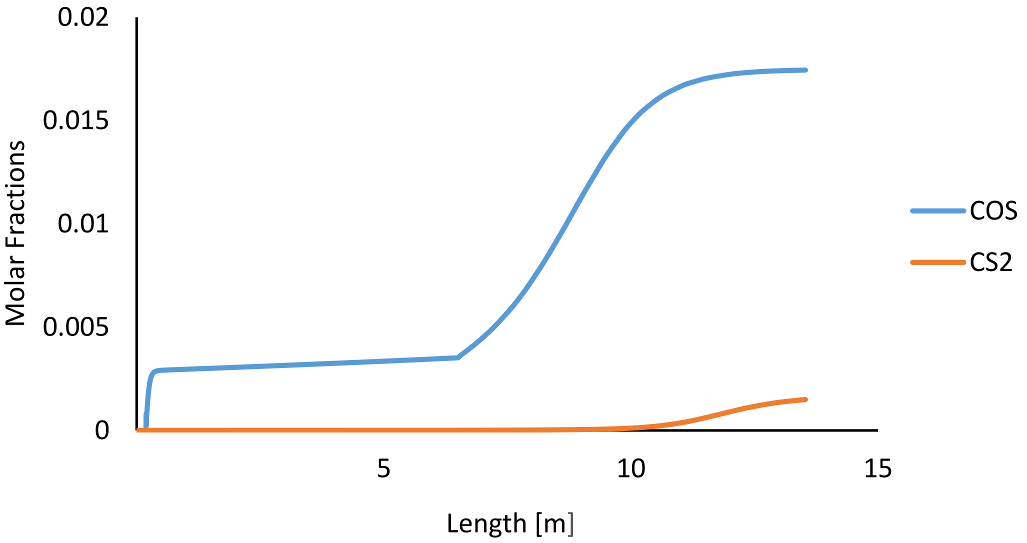


Figure 57: Shiraz RTR, COS and CS_2 profiles

SO_2 is produced in low concentrations during acid gas pyrolysis. With increasing concentration of H_2S in feed, SO_2 concentration was found to decrease due to the preferential consumption of O atoms by H_2 (produced in high amounts for high H_2S concentrations in feed) that reduces S2 oxidation. The H_2S content in acid gas can be increased to enhance the desired H_2 and CO production while minimizing undesired products (SO_2 and COS) through CO_2 separation from acid gas using techniques discussed in the literature. It is important to remark how SO_2 as a by-product is very undesirable, as it carries no commercial value and thus it is a waste that must be disposed (i.e. a cost). Economic considerations regarding the process and the optimized version will be expanded later.

COS and CS_2, instead, are produced in very small quantities in the combustion chamber but their formation benefits from the quenching of the outlet mixture that happens in the WHB, as they are one result of the numerous recombination reactions that happen in this phase.

The last tables help visualizing the change in the composition that can be obtained adopting the optimized conditions and the concentration of the outlet mixture going to the catalytic section, and report the increase in the objective function: the massive yield reaches 2.3% (up from 1.6%), i.e. a 4% molar one.

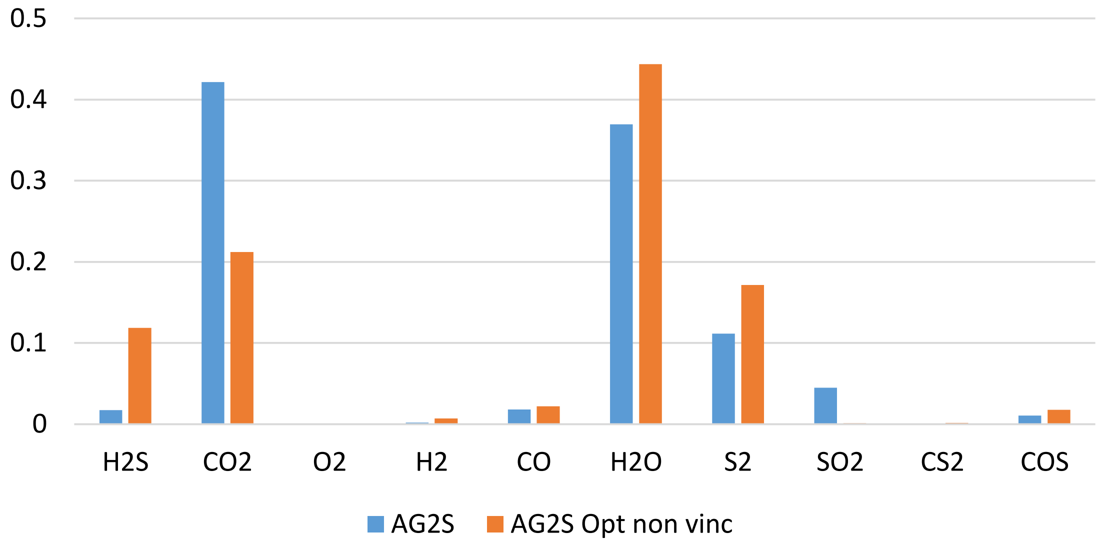


Figure 58: Shiraz RTR, optimized vs non-optimized outlet mixture composition

	AG2S™	**AG2S™ Optimized**
H_2S	1.719116	11.85488
CO_2	42.13685	21.22961
O_2	9.20323E-12	1.04E-12
H_2	0.199763	0.716635
CO	1.823151	2.19127
H_2O	36.95317	44.38982
S_2	11.13149	17.16347
SO_2	4.49193	0.098079
CS_2	2.48E-02	0.150007
COS	1.07	1.74454
T_{MAX}	1395 °C	1360°C
W O_2^{IN}	4077 kg/h	4157 kg/h
$(H_2S/ CO_2)_{IN}$	0.524	1.51
Syngas massive yield	1.6%	2.4%

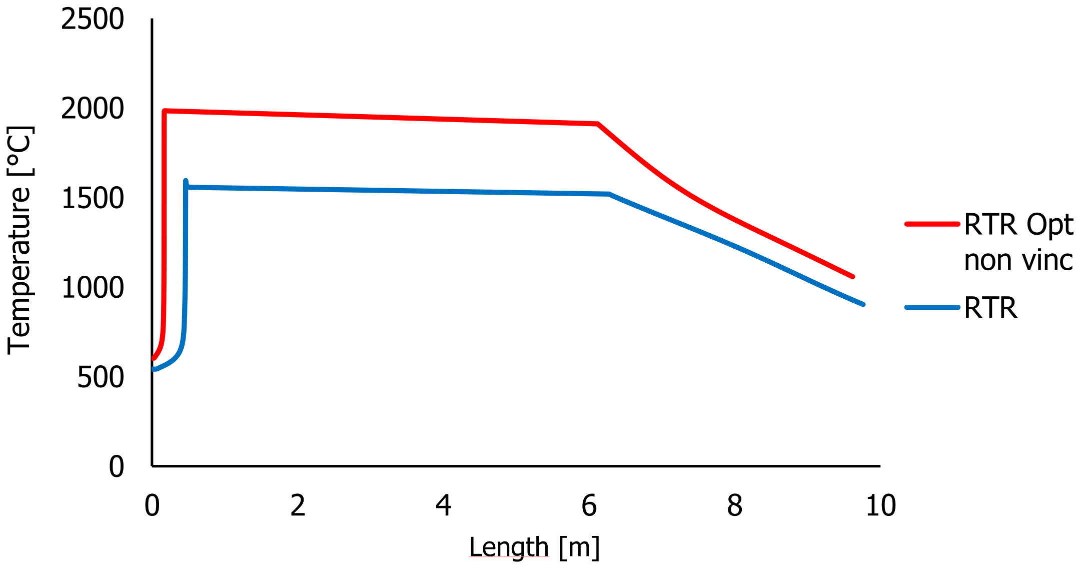
Table 17: Shiraz RTR, optimized vs non-optimized final comparison

3.2. Khangiran: Optimized Process

This time the robust optimizer converges towards a feed stream richer in CO_2 than H_2S (H_2S/ CO_2 ratio equal to 0.317) and consequently to higher oxygen flow rate directed inside the furnace, as carbon dioxide endothermic decomposition needs high temperatures to take place.

The results are then affected by the particular composition that is entering the RTR: composition-wise, much larger quantities of CO and SO_2 are obtained, since CO_2 provides excess oxygen that reacts preferentially with Sulphur. Flame temperature reaches up to 2000°C, a value very different from the non-optimized one and uncommon even in thermal reactors of this kind. Hydrogen sulfide conversion is almost 100% but selectivity is preferentially directed towards SO_2, as shown in Fig 58.

CS_2 and COS, together with SO_2, represent dangerous or waste compounds that must be disposed or converted into something else (usually this happens inside the catalytic zone). The formation of the first two, in this case, is affected by the very high temperature at which the flame is operated



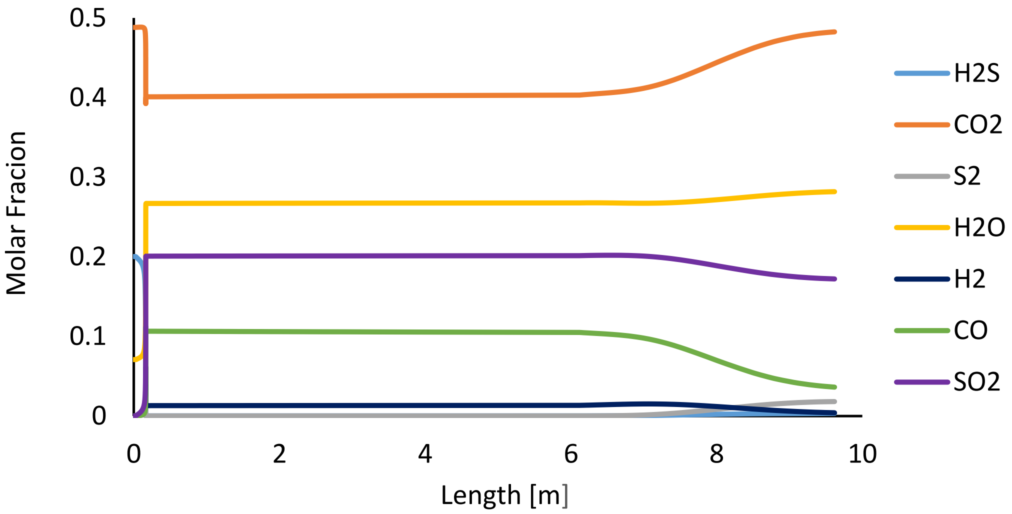


Figure 59: Khangiran RTR optimized vs non-optimized temperature profile;

Figure 60: Khangiran RTR, optimized composition profile

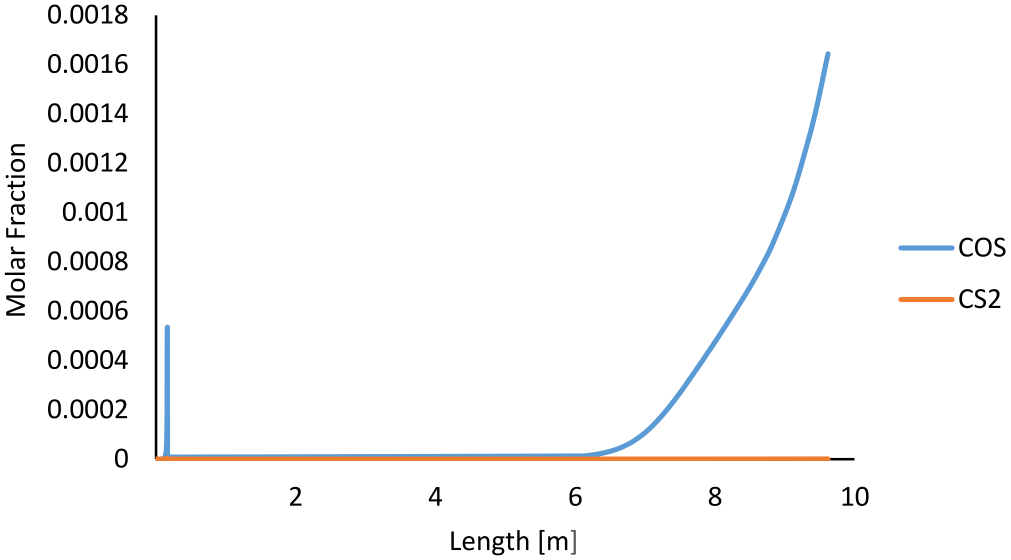


Figure 61: Khangiran RTR, COS and CS_2 profiles

and as a result their concentration at the outlet of the WHB is lower than the ones in the non-optimized process. Temperature and concentration profiles are reported below:

Again, CO_2 conversion happens only to a small extent, especially if compared to H_2S one. The latter of the two is necessary on order to avoid S_2 production and shift the selectivity of the overall reaction towards syngas. As a downside, the new inlet composition, richer in CO_2, provides a suitable environment for SO_2 formation: the compound is a waste and has to be disposed of.

The last graphs and tables report the differences in terms of overall performances of the non-optimized and optimized process. Remarkably, syngas massive yield increases to 3.8% from 2.95% (5.5% yield in molar terms).

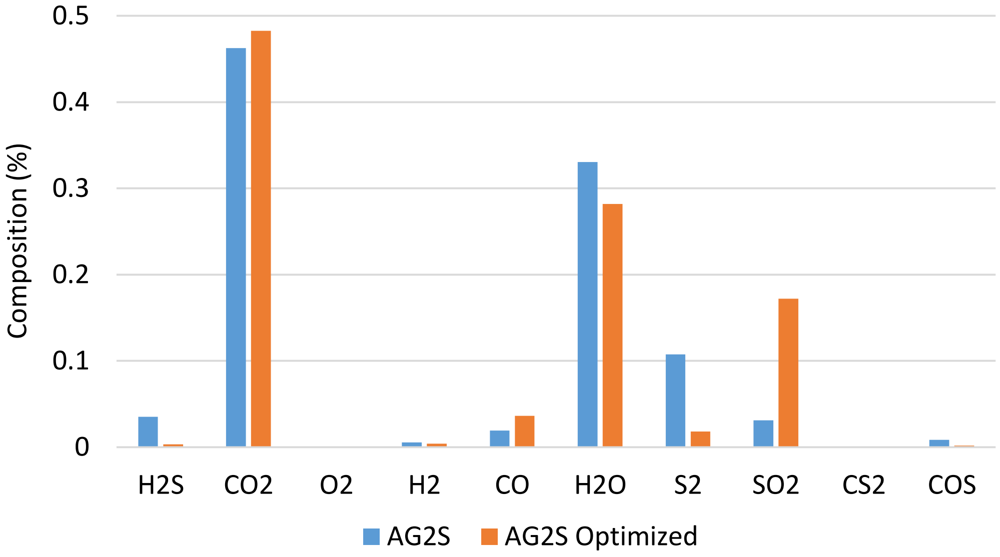


Figure 62: Khangiran RTR, optimized vs non-optimized outlet mixture composition

	AG2S™	**AG2S™ Optimized**
H_2S	3.508496	0.313943
CO_2	46.26892	48.26744
O_2	1.27E-11	9.1E-09
H_2	0.527051	0.396361
CO	1.900362	3.61661
H_2O	33.04482	28.18451
S_2	10.75337	1.805437
SO_2	3.112079	17.21644
CS_2	0.028766	4.52E-05
COS	0.856134	0.16431
T_{MAX}	1230 °C	1984°C
$M\ O_2^{IN}$	21906 kg/h	26417 kg/h
$(H_2S/\ CO_2)_{IN}$	0.416	0.318
Syngas massive yield	2.95%	3.8%

Table 18: Khangiran RTR, optimized vs non-optimized final comparison

3.3. General considerations

The two optimized process show some similarities and some significant differences:

- Inlet temperature should be around 450-550°C in order to allow sustainability of the flame and operation at higher temperatures than the one of the Claus furnace;

- O_2 concentration at the inlet of the RTR (just before the flame) should be 15-20% of the overall mixture; the dependence of the overall performances from this variable grows as CO_2 content in the acid gas increases, as its decomposition is endothermic;

- Syngas yield follows different behaviors at high (>1500°C) or low RTR temperatures: in the first case decreasing values of the H_2S/CO_2 ratio are preferred (with an optimum around 0.3), while in the second case the ratio should increase towards an optimum of 1.5; nevertheless, the yield assesses around 2-3%.

- The only heterogeneous (with respect to acid gases and oxygen) compound considered at the inlet of the RTR is water, with a molar fraction of 7-8%. Its presence enhances, at least in the flame, H_2 and SO_2 formation thanks to active radicals reactions, and inhibits COS and CS_2 formation through CO as it fights against CO_2 decomposition lowering the temperature in the combustion chamber.

4. AG2S™ Constrained Optimization Results

Constrained optimization analysis of the case studies presented before finds new motivations in the results of the unconstrained one. While the first of the two does not seem to present particular issues from an operational point of view, the second one, starting form a feed very rich in CO_2, tends to converge towards even greater inlet concentrations of carbon dioxide and thus even higher temperatures that make the configuration industrially unfeasible.

Further studies of this situation are interesting, since a key point of AG2S™ process is that CO_2 presence is not only accepted but attractive, so completely the opposite of what happens and has been largely analyzed for the Claus process.

The results for the process optimization subjected to a constrain on the H_2S/SO_2 ratio at the outlet of the WHB and to a constrain on the maximum temperature allowed in the combustion chamber are presented in the following paragraphs.

4.1. Shiraz: Optimized Process with Constraints

Shiraz acid gas and RTR react to the presence of constraints on the optimization procedure by showing very little variation from the unconstrained case. After all, the first result, presented in the previous paragraphs, was already acceptable on his own as far as the temperature profile and acid gas conversion are concerned.

Imposing the respect of a certain outlet ratio for H_2S and SO_2, then, has the obvious effect of lowering the H_2S/CO_2 inlet ratio, bringing consequently the temperature to higher values and enhancing hydrogen sulfide conversion. At the same time carbon dioxide decomposition yields, as a byproduct, SO_2, which was completely absent in the previous optimization work.

The constrain on the maximum temperature (set at 1300°C) does not contribute to change in a significant way the consideration expressed for the unconstrained optimization: in fact, a maximum temperature of 1360°C was already obtained, making it necessary only to slightly lower the H_2S/CO_2 inlet ratio. As a result, in both cases syngas yield decreases (this was expected) reaching a value around 1.7%, which resembles the original one, related to the non-optimized process.

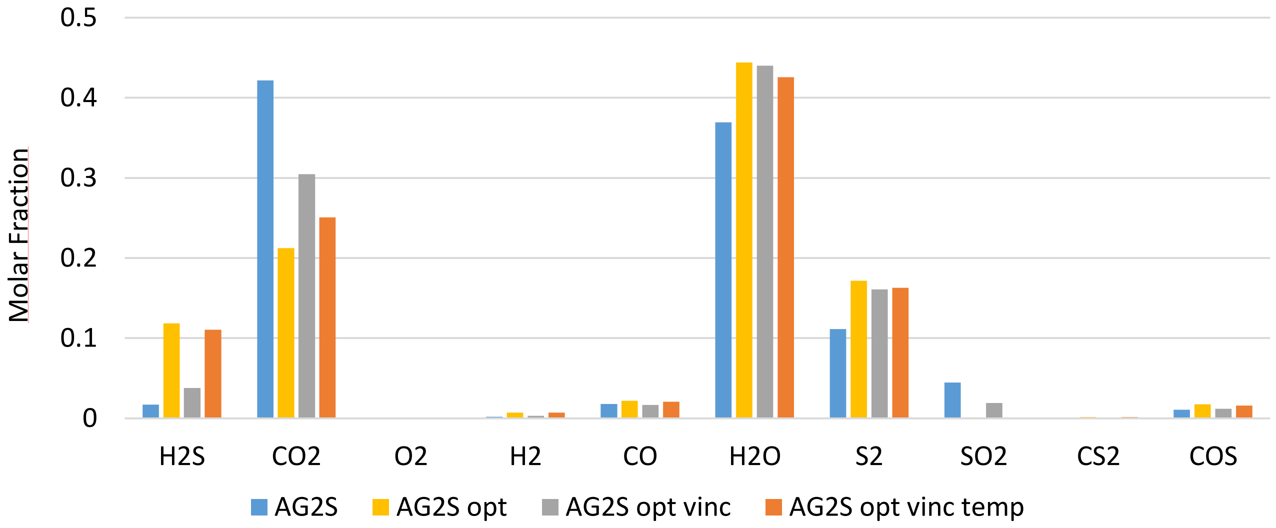


Figure 63: Shiraz RTR, outlet mixture composition for the non-optimized, optimized and optimized with constraints cases

Fig 63 and Fig 64 help visualizing the differences in the composition and in the temperature profile for every operating point of this RTR, either non-optimized, optimized and optimized with well-defined constraints.

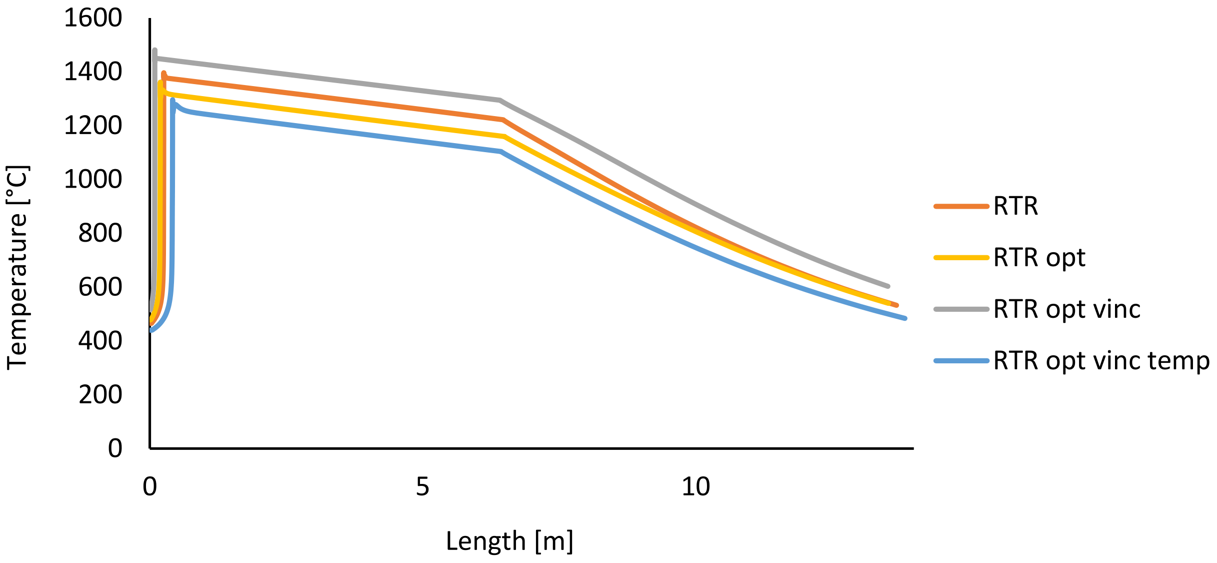


Figure 64: Shiraz RTR, temperature profile for the non-optimized, optimized and optimized with constraints cases

Table 19 reports the composition at the outlet section of the WHB for each situation, which summarizes the considerations reported above.

ù

	AG2S™	**AG2S™ Optimized (H2S/ SO2 constr)**	**AG2S™ Optimized (Temp constr)**
H_2S	1.719116	3.802456	11.07135
CO_2	42.13685	30.44957	25.07592
O_2	9.20323E-12	3.89E-12	0
H_2	0.199763	0.304202	0.697828
CO	1.823151	1.650598	2.074408
H_2O	36.95317	44.01345	42.55715
S_2	11.13149	16.07213	16.27548
SO_2	4.49193	1.900459	0.075859
CS_2	2.48E-02	0.032652	0.147396
COS	1.07	1.20332	1.605534
T_{MAX}	1395 °C	1481°C	1296°C
M O_2^{IN}	4077 kg/h	4700 kg/h	3842 kg/h
$(H_2S/ CO_2)_{IN}$	0.524	0.935	1.24
Syngas massive yield	1.6%	1.67%	1.69%

Table 19: Shiraz RTR, non-optimized, optimized and optimized with constraints final comparison

4.2. Khangiran: Optimized Process with Constraints

The first constrain, on the H_2S/SO_2 ratio, forces the optimizer to point towards higher relative ratios of the acid gas feed of the RTR. This contributes at reaching the goal by lowering both H_2S conversion and SO_2 production. The decrease in CO_2 initial concentration has the effect of lowering the oxygen flow rate: doing so, the temperature dramatically decrease, since the very large reacting mass is very sensitive to any change in this second independent variable, upon which depends its ignition in the combustion chamber.

The optimal combination of the two independent variables leads to a syngas yield lower than the one related to the non-optimized configuration (2.61% vs 2.91%). As the maximum temperature obtained (1245°C) is lower than the one allowed by our second constrain, there is still room to improve the yield by running another optimization analysis.

As expected, the results show a slight decrease in the H_2S/CO_2 ratio and a slight increase in the oxidizer flow rate. In this way the set value of 1300°C is reached while promoting a better H_2S conversion (which is beneficially not complete) and SO_2 production. The new operating point are responsible for a yield which goes up to 2.96% and is at least comparable to the original one.

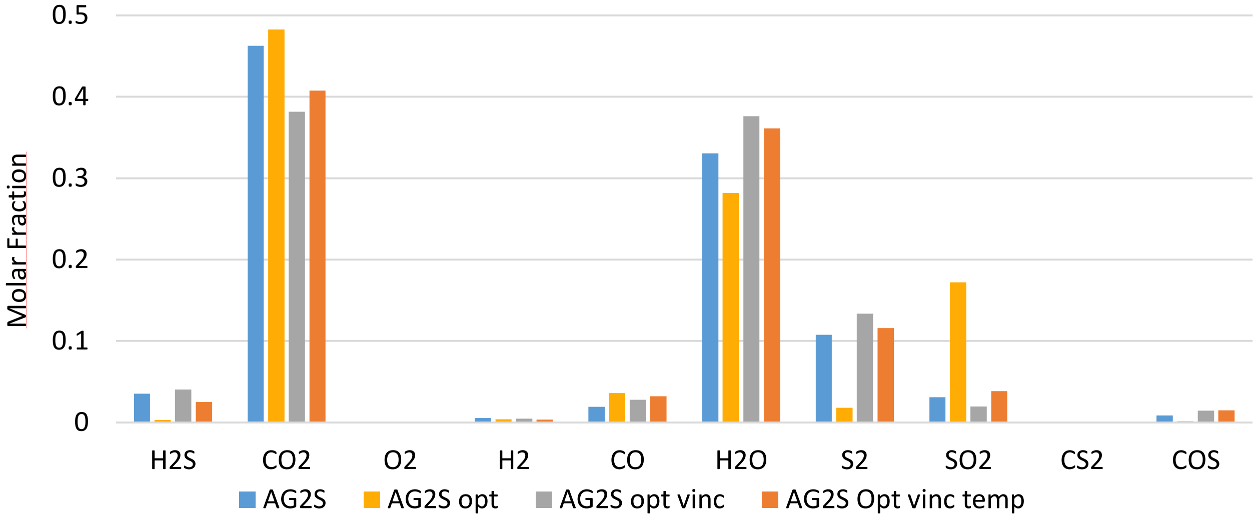


Figure 65: Khangiran RTR, outlet mixture composition for the non-optimized, optimized and optimized with constraints cases

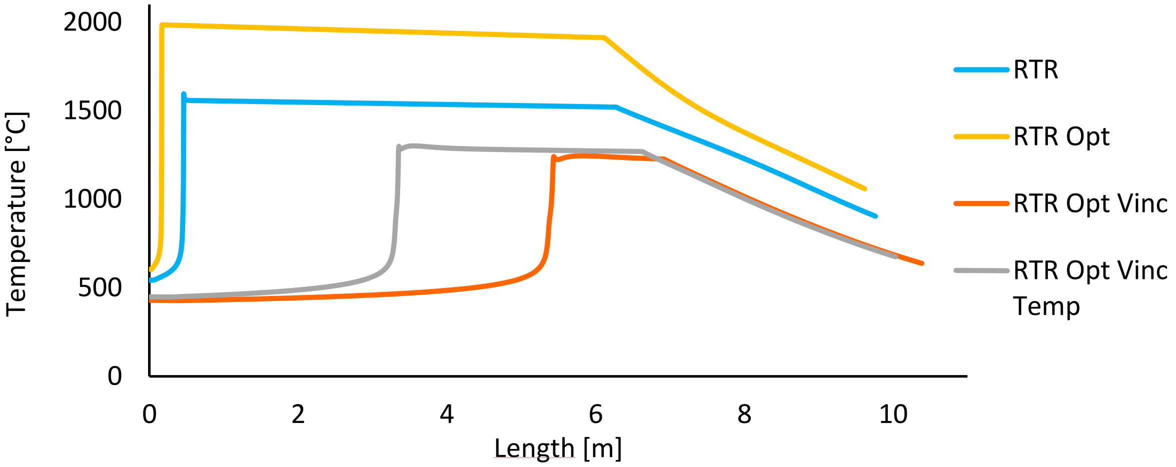


Figure 66: Khangiram RTR, temperature profile for the non-optimized, optimized and optimized with constraints cases

Fig 65 and Fig 66 help visualizing the differences in the composition and in the temperature profile for every operating point of this RTR, either non-optimized, optimized and optimized with well-defined constraints.

Table 20 reports the composition at the outlet section of the WHB for each situation, which summarizes the considerations reported above.

	AG2S™	AG2S™ Optimized (H2S/ SO2 constr)	AG2S™ Optimized (Temp constr)
H_2S	3.508496	4.021618	2.485872
CO_2	46.26892	38.16579	40.73939
O_2	1.27E-11	1.19E-11	2.66E-11
H_2	0.527051	0.477579	0.360808
CO	1.900362	2.797692	3.22476
H_2O	33.04482	37.60099	36.12055
S_2	10.75337	13.37368	11.5777
SO_2	3.112079	1.949337	3.82442
CS_2	0.028766	0.044163	0.024156
COS	0.856134	1.438778	1.490601
T_{MAX}	1230 °C	1245°C	1300°C
$M\ O_2^{IN}$	21906 kg/h	15145 kg/h	15934 kg/h
$(H_2S/\ CO_2)_{IN}$	0.416	0.629	0.533
Syngas massive yield	2.95%	2.61%	2.96%

Table 20: Khangiran RTR, non-optimized, optimized and optimized with constraints final comparison

4.3. General considerations

Both case studies show that imposing a constraint upon the objective function leads the process to perform worse than in the unconstrained optimization version of the same, on a level which is comparable to the non-optimized one. This work was nonetheless necessary as few facts below can point out:

- The constraint on H_2S/SO_2 outlet ratio is the most strict but also very much necessary if one intends to operate efficiently the catalytic zone of the process. Here, in fact, Claus reaction takes place, converting Sulphur dioxide to Sulphur and elimination of the hazardous compounds CS_2 and COS is possible.

- The constraint on the maximum temperature may be the most interesting in term of revamping of old Claus SRUs to RTRs, as the former are designed to sustain a certain kind of stresses (not only in term of materials resistance but also in terms of supply of fuel for the flame) which is preferable not to modify mainly from an economical point of view; furthermore results show that for the second case studies this objective function is the one that offers the best compromise in terms of syngas yield and feasibility of the process.

5. AG2S™ Economic Optimization

The motivation behind the AG2S™ solution is to find an alternative to the Claus process able to face the continuously decreasing price of Sulphur, which is affecting the appeal of the well-known method for refinery tail gas treatment. It is then natural the one of the principal fields of evaluation of the goodness of the novel configuration is the economic one.

An analysis of the pay-back time for a possible SRU revamping has been done by Marco Fontana during his thesis work, so the focus in this last chapter will be to apply the newly developed robust optimization algorithm to an objective function that represent the revenue derived from running at an industrial level the AG2S™ process.

5.1. Definition of the new objective function

The simplest and quickest way to evaluate the economic quality of a process or any activity is the computation of the profit, usually defined as revenue minus cost. The problem to be solved by the optimizer is then:

$$\min_{\left(\left(\frac{H_2S}{CO_2}\right)_{IN},\, \dot{m}\, O_{2,IN}\right)} -(Revenue - Cost)$$

In this simplified situation, the cost term is made up by:

- Price of steam generated to heat the process feed in the pre-heater;
- Price of running the air separation unit that generates the pure O_2 that serves as oxidizer;
- Price of running the units necessary to dispose of the SO_2 generated by the RTR.

The revenue term, instead, considers the following factors:

- Market price of syngas;
- Market price of Sulphur;
- Value of the steam generated in the WHB by the reaction quenching.

Table 21 summarizes the prices considered for every terms cited above.

	Price
Sulphur	0.0115 €/kg
Pure oxygen	0.042 €/kg
Syngas	0.19 €/kg
HP Steam	20 €/GJ
SO_2 **disposal**	0.43 €/kg

Table 21: prices for several components involved in the AG2S™ thermal section

5.2. Economic Optimization Results

The structure of the objective function leads to results, in terms of independent variables, which are comparable in both case studies. This is appreciable and finds its reasons in the very structure of the objective function, that manages to assign a different weight to every term by which it is composed.

It happens then that the first concern is minimizing SO_2 quantity at the outlet, since it is a waste and in no way can represent a useful by-product of the process. The second direction the optimizer pursuits is maximizing syngas yield, as, among all components coming out of the RTR, is the one that carries the highest intrinsic value. A similar logic dictates the selection of an operating point suited for Sulphur production.

It can be noted that the presence of prices for SO_2 disposal and pure O_2 generation push in the same direction: increase in the initial H_2S/CO_2 ratio and decrease in the oxidizer flow rate are the working condition of choice if the presence of CO_2 is limited.

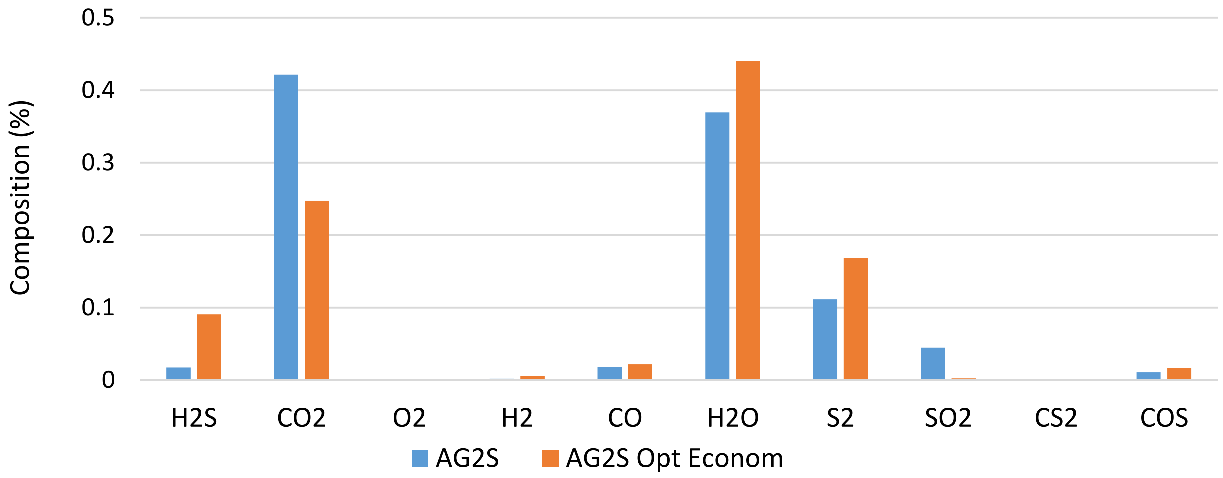


Figure 67: Shiraz RTR, economic optimized vs non-optimized outlet mixture composition

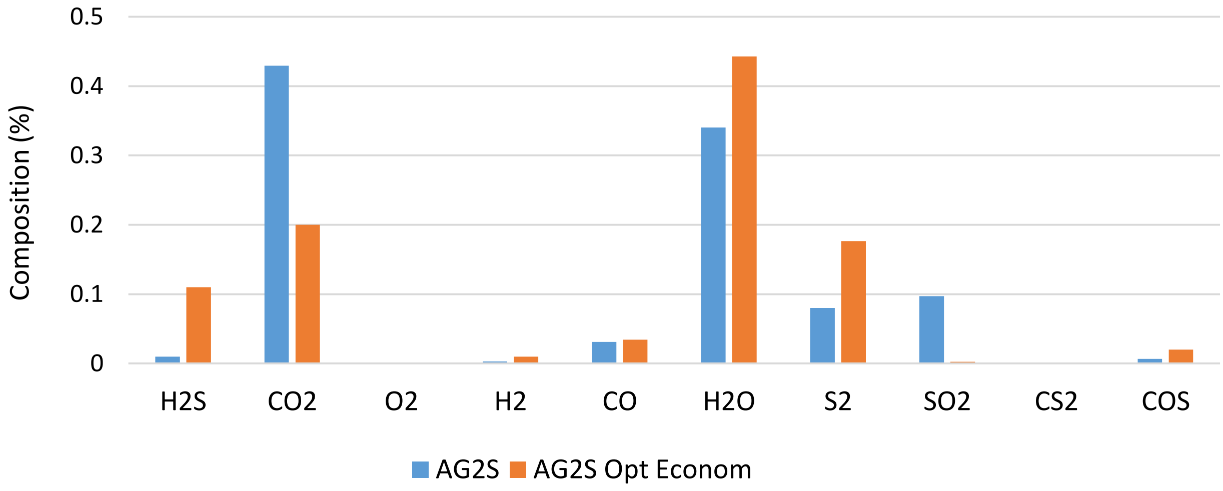


Figure 68: Khangiran RTR, economic optimized vs non-optimized outlet mixture composition

Fig 67 and Fig 68 report the composition at the outlet of the WHB for the two RTR run at the operating point determined by the robust optimizer following an analysis from an economic point of view. Noticeably, the trend for every compound is exactly the same, indicating that the process has a well-defined optimum that applies to every situation.

The optimal inlet composition, furthermore, turns out to be the basically the same confirming the assumption made before: 25% CO_2, 45% H_2S and 16% O_2, among all the other components.

Fig 69 and Fig 70 show the temperature profiles for the two RTR. Even without imposing a constrain on the maximum temperature, its value reaches up to 1350°C in both cases. A conclusion having general character can then be made: for the AG2S™ process, an optimal working point for mixtures having a H_2S/SO_2 ratio around 1.3-1.5 is represented by the temperature specified above.

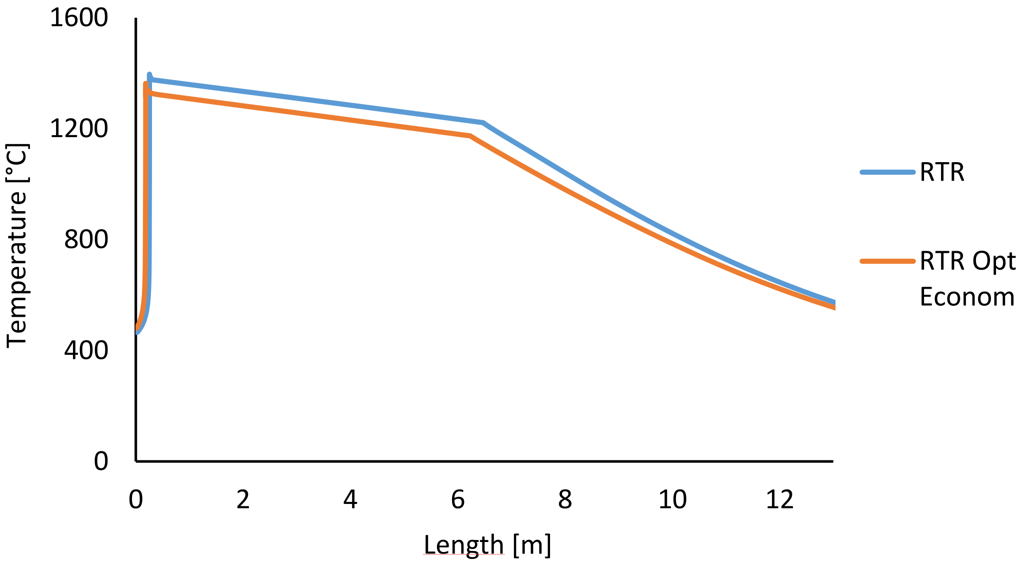


Figure 69: Shiraz RTR, economic optimized vs non-temperature profile

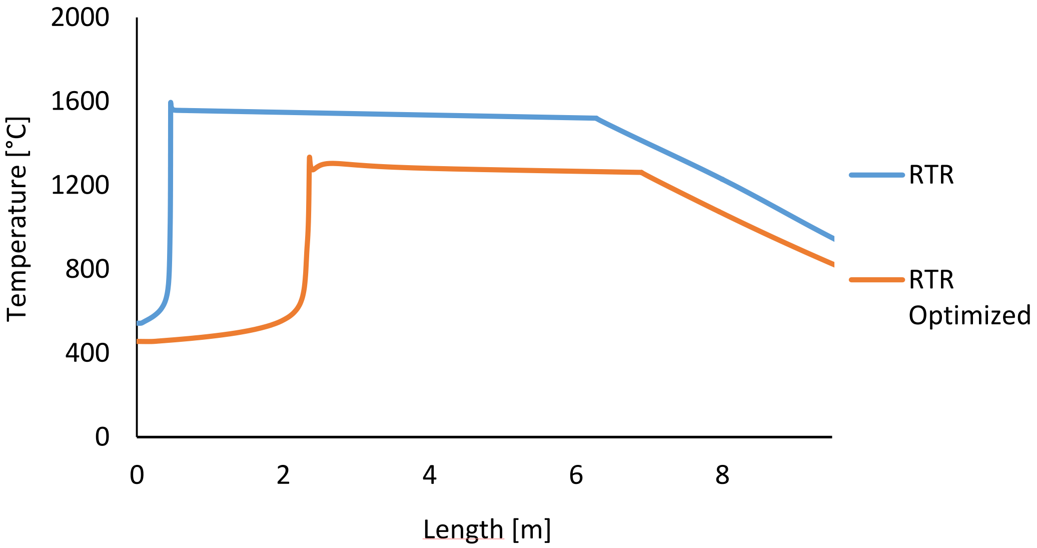


Figure 70: Khangiran RTR, economic optimized vs non-temperature profile

Table 22 and Table 23 summarize the results. It is interesting to point out that syngas yield increases with respect to the not-optimized situation, reaching values that are comparable to the ones relative to the unconstrained optimization. This means that an economical optimization is the one really meaningful for the AG2S™ process, perfectly in line with the motivation that lies behind its existence and continuous development.

	AG2S™	**AG2S™ Economic opt**
H_2S	1.719116	9.057788
CO_2	42.13685	24.75674
O_2	9.20323E-12	1.41E-12
H_2	0.199763	0.572444
CO	1.823151	2.174981
H_2O	36.95317	44.07045
S_2	11.13149	16.83168
SO_2	4.49193	0.22127
CS_2	2.48E-02	0.114573
COS	1.07	1.690564
T_{MAX}	1395 °C	1363°C
$M\ O_2^{IN}$	4077 kg/h	4107 kg/h
$(H_2S/\ CO_2)_{IN}$	0.524	1.24
Syngas massive yield	1.6%	2.13%

Table 22: Shiraz RTR, economic optimized vs non-optimized final comparison

	AG2S™	**AG2S™ Optimized**
H_2S	3.508496	0.109881
CO_2	46.26892	0.199968
O_2	1.27E-11	3.51E-14
H_2	0.527051	0.010027
CO	1.900362	0.034458
H_2O	33.04482	0.442793
S_2	10.75337	0.176614
SO_2	3.112079	0.002503
CS_2	0.028766	0.000606
COS	0.856134	0.020042
T_{MAX}	1230 °C	1334°C
$M\ O_2^{IN}$	21906 kg/h	17985 kg/h
$(H_2S/\ CO_2)_{IN}$	0.416	1.5
Syngas massive yield	2.95%	3.53%

Table 23: Khangiran RTR, economic optimized vs non-optimized final comparison

CONCLUSIONS AND FUTURE DEVELOPMENTS

The work that has been presented was aimed to realize an efficient connection between different simulation environments in order to create a robust optimization algorithm (from the BzzMath Library, written in C++ language) able to combine quick computations for standard units (in Aspen Hysys), reactors having complex kinetics (made possible by OpenSMOKE++).

That algorithm has been developed, written and tested. The testing part exploits the AG2S™ process, a novel solution for refinery acid gases treatment recently developed at Politecnico di Milano, through application of the configuration to real Claus SRUs. It has been found that an economic analysis is the one that yields the better results, just as an economic reason is the one behind the AG2S™ process (shifting the selectivity of Sulphur recovery from elementary Sulphur to syngas).

The great flexibility of the algorithm paves the way to a whole new set of similar or deeper exams: as a first further step, a process run with air instead of pure O_2 can be considered, as well as one having other heterogeneous compounds in the feed than water: hydrocarbons (aromatic or not) and even ammonia, which forces to split the inlet steam and feed it in different point of the combustion chamber.

A more complex development could regard the extension of the optimization problem to the whole AG2S™ plant, made up not only by the thermal and the catalytic zone but also by the upstream TGT absorption columns and the downstream unit dedicated to syngas purification and treatment (to bring its H_2/CO ratio towards more commercially attractive values).

At the basis of all of this progresses lies an efficient estimation of the AG2S™ reaction (redox H_2S and CO_2 pyrolysis) kinetic mechanism. A new and complete scheme has been recently developed but the work in this field continues towards more and more precise estimation of the parameters, that can give great reliability to the simulations, which turn out to be of fundamental importance when dealing with compounds like H_2S, that present a high degree of hazard from an experimental point of view.

It is important to remark that the C++ code needed for analysis of this kind has general value and must undergo only small changes to fit the function wrapped by the Matlab Compiler into a shared library, and so does said Matlab function every time the objective function needs to be changed. This positive traits make this method very attractive and suited to be used to efficiently solve a wide field of different problems.

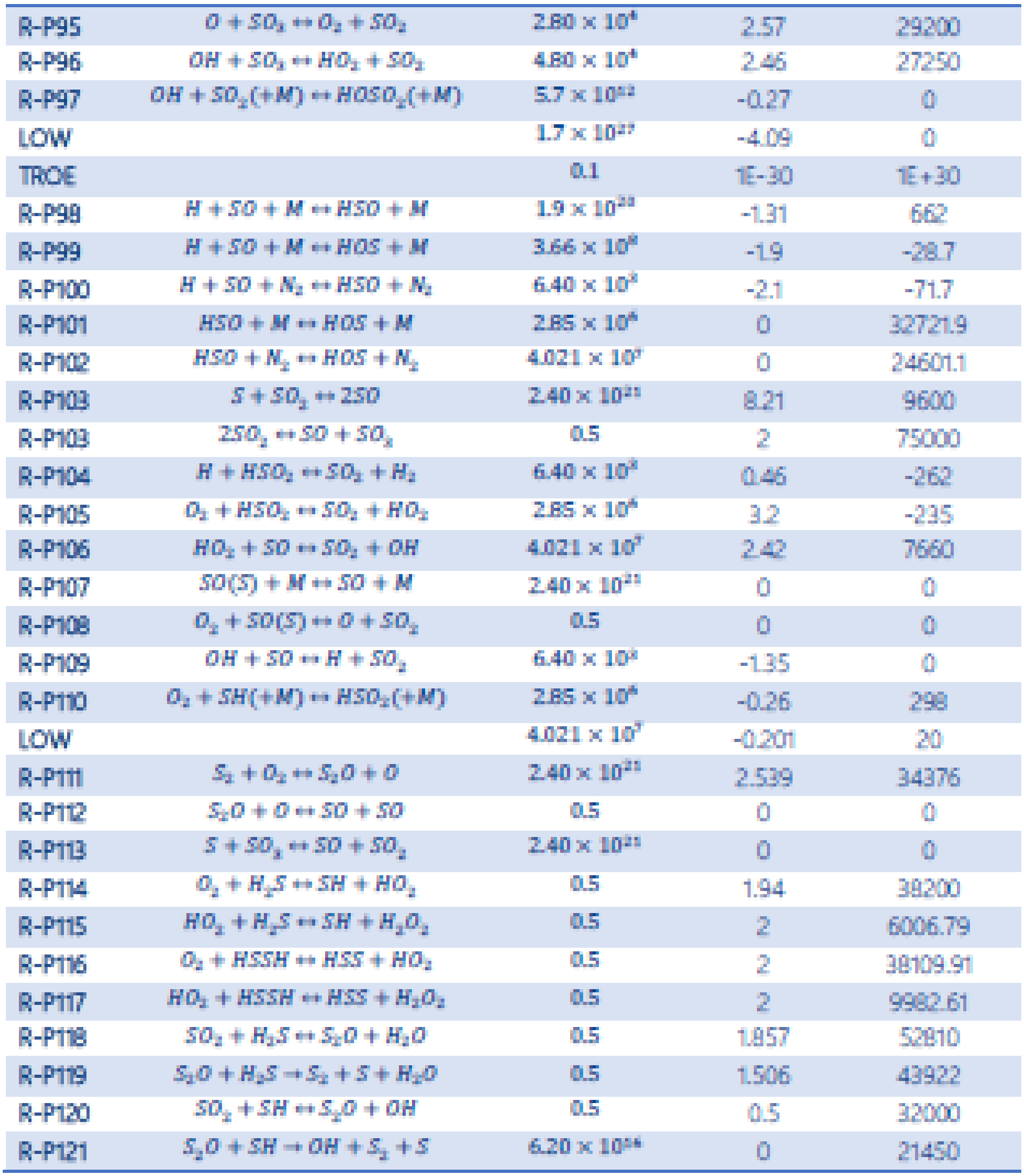
BIBLIOGRAPHY

1. Collodi G, S.D., *Gassificazione. Enciclopedia degli idrocarburi Treccani*.
2. ; N. J. Nabikandi, Fatemi S. (2015); *Kinetic modelling of a commercial Sulfur recovery unit based on Claus straight through process: Comparison with equilibrium model*; Journal of Industrial and Engineering Chemistry.
3. H. Ghahraloud, M. Farsi, Rahimpour (2017); *Modeling and Optimization of an Industrial Claus Process: Thermal and Catalytic Section*; Journal of the Taiwan Institute of Chemical Engineers.
4. Rezazadeh R., Rezvantalab S. (2013); *Investigation of Inlet Gas Streams Effect on the Modified Claus Reaction Furnace*; Advances in Chemical Engineering and Science.
5. Groisil M., Salisu I., Al Shoaibi A.S., Gupta A.K.; *Acid Gas Simulation for Recovering Syngas and Sulphur*; Proceedings of the ASME 2015 Power & Energy Conference.
6. Elmelih A., Salisu I., Al Shoaibi A.S., Gupta A.K. (2016); *Experimental examination of syngas recovery from acid gases*; Applied Energy. 164. 64-68. 10.1016/j.apenergy.2015.11.025.
7. Salisu I., Abijheet R. (2016); *Kinetic Simulation of Acid Gas (*H_2S *and* CO_2*) Destruction for Simultaneous Syngas and Sulfur Recovery*; Ind. Eng. Chem. Res., 55 (24), pp 6743–6752.
8. Groisil M., Salisu I., Al Shoaibi A.S., Gupta A.K. (2015); *Numerical Examination of Acid Gas for Syngas and Sulfur Recovery*; Energy Procedia 75, 3066 – 3070.
9. Elmelih A., Salisu I., Al Shoaibi A.S., Gupta A.K. (2016); *Reactor Parameters Effects on Hydrogen Production from Hydrogen Sulfide*; 14th International Energy Conversion Engineering Conference.
10. Bassani, A., *Conversione di gas acidi in gas di sintesi. Applicazione al processo di gassificazione del carbone e studio di fattibilità su scala industriale*. 2014.
11. Andrea Bassani, Carlo Pirola, Enrico Maggio, Alberto Pettinau, Caterina Frau, Giulia Bozzano, Sauro Pierucci, Eliseo Ranzi, Flavio Manenti; *Acid Gas to Syngas (*AG2S™ *) technology applied to solid fuel gasification: Cutting H_2S and* CO_2 *emissions by improving syngas production*; Applied Energy (2016);
12. Manenti, G., Molinari, L., Manenti, F., *Syngas from H_2S and CO_2: an Alternative, Pioneering Synthesis Route?*;, Hydrocarbon Processing, 6, 2016.
13. Salisu I.; Ramees K.R., Abijheet R. (2017); *Effects of H_2O in the Feed of Sulfur Recovery Unit on Sulfur Production and Aromatics Emission from Claus Furnace*; Ind. Eng. Chem. Res., 56 (41), pp 11713–11725.
14. H. Kazempour, F. Pourfayaz*, M. Mehrpooya (2017); *Modeling and multi-optimization of thermal section of Claus process based on kinetic model*; Journal of Natural Gas Science and Engineering.
15. M. Fontana, "*Approccio multi-scala per la riconversione dei gas acidi a gas di sintesi*"; 2015.
16. F. Cecchetto, "AG2S™ *Technology: Model-Based Design And Economic Assessment*"; 2016.
17. A. El Ziani, "AG2S™ *Technology: Development Of A Detailed Kinetic Mechanism And Study Of Sulfur System Towards* AG2S™ *Process Simulation"*; 2017
18. Manenti F., Buzzi-Ferraris G., *Nonlinear Systems and Optimizationfor the Chemical Engineer – Solving Numerical Problems*; Wiley (2015)
19. *Air Pollution Control Technology Fact Sheet*; EPA-452/F-03-034

APPENDIX A: KINETIC MODEL FOR AG2S™ REACTION [17]

ID N°	Reaction	A (cm³/mol/s)	n	E (cal/mol)
R-P27	$H_2S + O \leftrightarrow HSO + H$	1.39×10^{8}	1.1	5099
R-P28	$H_2S + O \leftrightarrow SH + OH$	1.80×10^{6}	2.6	2532
R-P29	$H_2S + OH \leftrightarrow SH + H_2O$	8.7×10^{13}	−0.7	0
R-P30	$H_2S + OH \leftrightarrow SH + H_2O$	4.07×10^{7}	1.800	0
R-P31	$O + SH \leftrightarrow OH + S$	1.8×10^{12}	0	0
R-P32	$O + SH \leftrightarrow OH + S$	4.32×10^{8}	2.100	3582.70
R-P33	$O + SH \leftrightarrow SO + H$	4.30×10^{11}	0.7240	1027
R-P34	$OH + SH \leftrightarrow H_2O + S$	1×10^{13}	0	0
R-P35	$HO_2 + SH \leftrightarrow OH + HSO$	1×10^{12}	0	0
R-P36	$O_2 + SH \leftrightarrow HO_2 + S$	4.72×10^{6}	2.000	36913
R-P37	$O_2 + SH \leftrightarrow OH + SO$	7.5×10^{4}	2.100	16384
R-P38	$O_2 + SH \leftrightarrow H + SO_2$	6.5×10^{11}	0	15000
R-P39	$O_2 + SH \leftrightarrow O + HSO$	1.9×10^{13}	0	17925
R-P40	$OH + S \leftrightarrow H + SO$	1.46×10^{13}	0.200	1361.4
R-P41	$O_2 + S \leftrightarrow O + SO$	5.4×10^{5}	2.110	1450
R-P42	$S_2 + O \leftrightarrow S + SO$	1×10^{13}	0	0
R-P43	$O + HSS \leftrightarrow S_2 + OH$	7.50×10^{7}	1.750	2900
R-P44	$OH + HSS \leftrightarrow S_2 + H_2O$	2.7×10^{12}	0	0
R-P45	$O + HSSH \leftrightarrow OH + HSS$	7.50×10^{7}	1.750	2900
R-P46	$OH + HSSH \leftrightarrow H_2O + HSS$	2.7×10^{12}	0	0
R-P47	$H + SO_2 \leftrightarrow O + HOSO$	2.5×10^{5}	2.920	50300
R-P48	$O + SO(+M) \leftrightarrow SO_2(+M)$	3.20×10^{13}	0	0
LOW		1.2×10^{21}	−1.54	0
TROE		0.5	1e − 30	1e30
R-P49	$OH + SO_2 \leftrightarrow O + HOSO$	3.9×10^{8}	1.89	76000
R-P50	$SO + M = O + S + M$	4×10^{14}	0	107000
R-P51	$OH + SO(+M) \leftrightarrow HOSO(+M)$	1.6×10^{12}	0.5	400
LOW		9.5×10^{27}	−3.48	970
R-P52	$O_2 + SO \leftrightarrow O + SO_2$	7.60×10^{3}	2.37	2970
R-P53	$H + HSO \leftrightarrow HSOH$	2.5×10^{20}	−3.14	920
R-P54	$H + HSO \leftrightarrow OH + SH$	4.9×10^{19}	−1.86	1560
R-P55	$H + HSO \leftrightarrow H_2O + S$	1.6×10^{9}	1.37	340
R-P56	$H + HSO \leftrightarrow H_2 + SO(S)$	1×10^{13}	0	0
R-P57	$H + HSO \leftrightarrow H_2SO$	1.8×10^{17}	−2.47	50
R-P58	$H_2SO \leftrightarrow O + H_2S$	4.9×10^{28}	−6.6	71700
R-P59	$H + HSO \leftrightarrow H_2 + SO$	1×10^{13}	0	0
R-P60	$O + HSO + M \leftrightarrow HSO_2 + M$	1.10×10^{29}	−1.73	-50
R-P61	$O + HSO \leftrightarrow H + SO_2$	4.5×10^{14}	−0.4	0
R-P62	$O + HSO + M \leftrightarrow HOSO + M$	6.9×10^{19}	−1.6	1590
R-P63	$O + HSO \leftrightarrow O + HOS$	4.8×10^{8}	1.02	5340

R-P64	$O + HSO \leftrightarrow OH + SO$	1.4×10^{13}	0.15	300
R-P65	$OH + HSO \leftrightarrow H_2O + SO(S)$	1×10^{13}	0	0
R-P66	$OH + HSO \leftrightarrow HOSHO$	5.2×10^{28}	-5.440	3170
R-P67	$OH + HSO \leftrightarrow H + HOSO$	5.3×10^{7}	1.570	3750
R-P68	$OH + HSO \leftrightarrow H_2O + SO$	1.7×10^{8}	1.03	470
R-P69	$O_2 + HSO \leftrightarrow HSO_2 + O$	8.4×10^{-7}	5.1	11312
R-P70	$HSOH \leftrightarrow OH + SH$	2.8×10^{39}	−8.75	75200
R-P71	$HSOH \leftrightarrow H_2O + S$	5.8×10^{29}	-5.6	54500
R-P72	$HSOH \leftrightarrow H_2S + O$	9.8×10^{26}	-3.4	86500
R-P73	$HOSO(+M) \leftrightarrow HSO_2(+M)$	1.0×10^{9}	1.03	50000
LOW		1.7×10^{35}	-5.640	55400
TROE		0.4	1E-30	1E+30
R-P74	$HOSO + M \leftrightarrow O + HOS + M$	2.5×10^{30}	-4.8	119000
R-P75	$OH + HOSO \leftrightarrow H_2O + SO_2$	6.0×10^{12}	0	0
R-P76	$OH + HSO_2 \leftrightarrow H_2O + SO_2$	1.0×10^{13}	0	0
R-P77	$HOSO_2 \leftrightarrow O + HOSO$	5.4×10^{18}	-2.34	106300
R-P78	$HOSO_2 \leftrightarrow H + SO_3$	1.40×10^{18}	-2.91	54900
R-P79	$H + HOSO_2 \leftrightarrow H_2O + SO_2$	1.0×10^{12}	0	0
R-P80	$O + HOSO_2 \leftrightarrow OH + SO_3$	5.0×10^{12}	0	0
R-P81	$OH + HOSO_2 \leftrightarrow H_2O + SO_3$	1.0×10^{12}	0	0
R-P82	$O_2 + HOSO_2 \leftrightarrow HO_2 + SO_3$	7.8×10^{11}	0	656
R-P82	$HOSHO \leftrightarrow H + HOSO$	6.40×10^{30}	-5.389	73800
R-P83	$HOSHO \leftrightarrow H_2O + SO$	1.2×10^{24}	-3.59	59500
R-P84	$H + HOSHO \leftrightarrow H_2 + HOSO$	1.0×10^{12}	0	0
R-P85	$O + HOSHO \leftrightarrow OH + HOSO$	5.0×10^{12}	0	0
R-P86	$HO + HOSHO \leftrightarrow H_2O + HOSO$	1×10^{12}	0	0
R-P87	$H + SO_2(+M) \leftrightarrow HSO_2(+M)$	5.30×10^{8}	1.59	2470
LOW		1.40×10^{31}	-5.190	4510
TROE		0.39	0.167	2191
R-P88	$H + SO_2(+M) \leftrightarrow HOSO(+M)$	2.4×10^{8}	1.630	7340
LOW		1.8×10^{47}	-6.140	11070
TROE		0.2830	272	3995
R-P89	$O_2 + HOSO \leftrightarrow HO_2 + SO_2$	9.6×10^{1}	2.355	10130
R-P90	$H + HOSO \leftrightarrow H_2 + SO_2$	1.80×10^{7}	1.720	1286
R-P91	$H + HOSO \leftrightarrow H_2O + SO(S)$	2.4×10^{14}	0	0
R-P92	$O + SO_2(+M) \leftrightarrow SO_3(+M)$	3.7×10^{11}	0	1689
LOW		2.40×10^{27}	-3.6	5186
TROE		0.442	316	7442
R-P93	$O + SO_2(+N_2) \leftrightarrow SO_3(+N_2)$	3.7×10^{11}	0	1689
LOW		2.90×10^{27}	-3.58	5206
TROE		0.430	1371	7442
R-P94	$H + SO_3 \leftrightarrow OH + SO_2$	8.4×10^{9}	1.220	3320



R-P95	$O + SO_3 \leftrightarrow O_2 + SO_2$	2.80×10^{4}	2.57	29200
R-P96	$OH + SO_3 \leftrightarrow HO_2 + SO_3$	4.80×10^{4}	2.46	27250
R-P97	$OH + SO_2(+M) \leftrightarrow HOSO_2(+M)$	5.7×10^{12}	-0.27	0
LOW		1.7×10^{27}	-4.09	0
TROE		0.1	1E-30	1E+30
R-P98	$H + SO + M \leftrightarrow HSO + M$	1.9×10^{23}	-1.31	662
R-P99	$H + SO + M \leftrightarrow HOS + M$	3.66×10^{8}	-1.9	-28.7
R-P100	$H + SO + N_2 \leftrightarrow HSO + N_2$	6.40×10^{8}	-2.1	-71.7
R-P101	$HSO + M \leftrightarrow HOS + M$	2.85×10^{6}	0	32721.9
R-P102	$HSO + N_2 \leftrightarrow HOS + N_2$	4.021×10^{7}	0	24601.1
R-P103	$S + SO_2 \leftrightarrow 2SO$	2.40×10^{21}	8.21	9600
R-P103	$2SO_2 \leftrightarrow SO + SO_3$	0.5	2	75000
R-P104	$H + HSO_2 \leftrightarrow SO_2 + H_2$	6.40×10^{8}	0.46	-262
R-P105	$O_2 + HSO_2 \leftrightarrow SO_2 + HO_2$	2.85×10^{6}	3.2	-235
R-P106	$HO_2 + SO \leftrightarrow SO_2 + OH$	4.021×10^{7}	2.42	7660
R-P107	$SO(S) + M \leftrightarrow SO + M$	2.40×10^{21}	0	0
R-P108	$O_2 + SO(S) \leftrightarrow O + SO_3$	0.5	0	0
R-P109	$OH + SO \leftrightarrow H + SO_2$	6.40×10^{8}	-1.35	0
R-P110	$O_2 + SH(+M) \leftrightarrow HSO_2(+M)$	2.85×10^{6}	-0.26	298
LOW		4.021×10^{7}	-0.201	20
R-P111	$S_2 + O_2 \leftrightarrow S_2O + O$	2.40×10^{21}	2.539	34376
R-P112	$S_2O + O \leftrightarrow SO + SO$	0.5	0	0
R-P113	$S + SO_3 \leftrightarrow SO + SO_2$	2.40×10^{21}	0	0
R-P114	$O_2 + H_2S \leftrightarrow SH + HO_2$	0.5	1.94	38200
R-P115	$HO_2 + H_2S \leftrightarrow SH + H_2O_2$	0.5	2	6006.79
R-P116	$O_2 + HSSH \leftrightarrow HSS + HO_2$	0.5	2	38109.91
R-P117	$HO_2 + HSSH \leftrightarrow HSS + H_2O_2$	0.5	2	9982.61
R-P118	$SO_2 + H_2S \leftrightarrow S_2O + H_2O$	0.5	1.857	52810
R-P119	$S_2O + H_2S \rightarrow S_2 + S + H_2O$	0.5	1.506	43922
R-P120	$SO_2 + SH \leftrightarrow S_2O + OH$	0.5	0.5	32000
R-P121	$S_2O + SH \rightarrow OH + S_2 + S$	6.20×10^{14}	0	21450