ULB UNIVERSITÉ LIBRE DE BRUXELLES

Bruface
Brussels
Faculty of
Engineering

VUB VRIJE UNIVERSITEIT BRUSSEL

Techno-Economic Analysis of the implementation of a Dryer for Medium-Scale Biomass Combustion Plants

Salvador Griñán Egea

Master thesis submitted under the supervision of
Prof. Dr. Julien Blondeau

the co-supervision of
Ir. Thibault Coppieters't Wallant

in order to be awarded the Degree of
Master of Science in Electromechanical Engineering
major in Energy

Academic year
2019 – 2020

The author(s) gives (give) permission to make this master dissertation available for consultation and to copy parts of this master dissertation for personal use. In all cases of other use, the copyright terms have to be respected, in particular with regard to the obligation to state explicitly the source when quoting results from this master dissertation.

08/06/2020

This master's thesis came about (in part) during the period in which higher education was subjected to a lockdown and protective measures to prevent the spread of the COVID-19 virus. The process of formatting, data collection, the research method and/or other scientific work the thesis involved could therefore not always be carried out in the usual manner. The reader should bear this context in mind when reading this Master's thesis, and also in the event that some conclusions are taken on board

---

Title: Techno-Economic Analysis of the implementation of a Dryer for Medium-Scale Biomass Combustion Plants

Author: Salvador Griñán Egea

Master of Science in Electromechanical Engineering – major in Energy

Academic year: 2019 – 2020

Abstract

The consumption of biomass as a fuel for heat and power generation is increasing its contribution to global final energy consumption. Replacing coal by biomass avoids the emission of a huge amount of carbon (in the form of CO_2) stored in the earth for hundreds of thousands of years. However, the properties of biomass, in particular its water content, have a significant impact on efficiency. Different techniques such as the pre-drying of biomass or the installation of a flue gas condenser, makes it possible to increase the efficiency of the boilers considerably. In this Master Thesis, the techno-economic feasibility of implementing a dryer in a medium-scale DH system is studied. The drying agent studied is the boiler flue gas itself which flows through a belt dryer. No external energy sources are considered. The drying effect is reflected in a decrease in fuel consumption. The results show that, due to the high estimated investment costs and the associated fixed cost defined $(10\ \%_{CAPEX}/year)$ the project is not viable even at high running hours. A realistic heat demand curve has also been analysed. Due to the summer and spring periods, the boiler operates under nominal conditions for about 25 % of the time and for less than 7000 h. Therefore, the project will not be feasible under the premises contemplated. Fixed cost, plant size and fuel price have an important impact on its feasibility.

Keywords: Biomass, Dryer, District Heat network, Techno-economic feasibility

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CO_2	carbon dioxide
H_2	Hydrogen
H_2O	Water
MW_e	Mega-Watt of electricity
O_2	Oxygen
ad	As Delivered basis
BFB	Bubbling Fluidised bed
BM	Bare module
CEPCI	Chemical Engineering Plant Cost Index
CFB	Circulating Fluidised bed
CHP	Combined Heat Power
daf	Dry ash free basis
db	Dry Basis
DH	Dristict Heat
ELV's	Emission Limit Values
ESP	Electrostatic Precipitator
EU	European Union
FB	Fluidised bed
FG	Flue Gas
FGD	Flue Gas Desulphurization
FOB	Free on board
FSP	fiber saturation poin
H&C	Heating and cooling
HHV	High Heat Value
HRU	Heat recovery unit
LCP	Large Combustion Plants
LCPD	Large Combustion Plan Directive
LF	Load Factor
LHV	Low Heat Value
MCPD	Medium Combustion Plan Directive
MSD	multy-stage drying
MSW	Municipal Solid Wastes
NPV	Net Present Value
O&M	Operation and Management
RDF	Refuse-derived fuel
RH	relative humidity
SCR	Selective Catalytic Reduction
SNCR	Selective non Catalytic Reduction
SSD	single-stage drying
wb	Wet Basis

Part I

Memory

Chapter 1

Introduction

In this first chapter, the main ideas on which the project is based will be developed. Firstly, the project will be contextualized, i.e., the different aspects that make the study of the viability of the installation of biomass dryers interesting will be developed. The following section develops the definition of biomass and the characteristics of it different types. An especial mention to the biomass moisture content takes place. The last part of this chapter will describe the different types of dryers and boilers that are usually used in the industry, specifically for biomass.

1.1 Context of the project

1.1.1 From large scale of Fossil Fuel Power Plants to medium scale CHP and District Heat plants

Over the last few decades, the contribution of solid fuels to the Gross inland consumption in Europe has dropped considerably. From 1990 to 2018, the share of solid fossil fuels has decreased to about half, being a 30 % since the beginning of the century. On the other side, the share of renewable energy and biofuel in the Gross inland consumption was 4.5 %, while nowadays it represents 14.5 %. These data and those from other sources such as the Nuclear heat, which now has a consumption similar to that of 30 years ago, can be seen in figure 1.1. This information is obtained from Eurostat [1].

Final energy consumtion corresponds to the Gross Energy Inland excluding deliveries to the energy transformation sector and the energy industries themselves. Heating and cooling (H&C) (being heating the major part) represents nearly half of the final energy consumption in the EU (data of 2017), without considering the electricity used for H&C[2]. This previously cited report also states that the contribution of Renewables in H&C, has increased from around 10 % in 2004 to nearly 20 % in 2017. Fossil fuels still account for about 80 % of the contribution but, due to the policies of descarbonization, this contribution in favour of renewable energies will continue decreasing.

In terms of electricity, in Europe, renewables represent 30.5 % of the gross electricity production and a 5.6 % is produced by bioenergy (in ktoe basis). Since 1990 the carbon footprint of the energy produced in Europe (EU28) has dropped by 43.5 %. This decline is due to the replacement of the use of solid fuel and petroleum products by gas or renewables [3]. Most bioelectricity in Europe (60 % in 2017) is generated by Combined Heat Power (CHP) plants, but there are cases in locations where heat is not needed, so it is important to recognise the role of biopower-only installations.

Nowadays, in Large Combustion Plants (LCP) that are electricity-only plants only $40 - 60$ % of the fuel energy can be converted into electricity [4], and the remaining energy is lost as low temperature waste heat into the air or water or both. This waste heat can be usefull in cases where a heat source in needed, such as an industry process or district heat networks. However, this is not always possible due to aspects suchs as the distance between the demand site and the plant, or when there ir not enough heat demand to make it economically feasible.

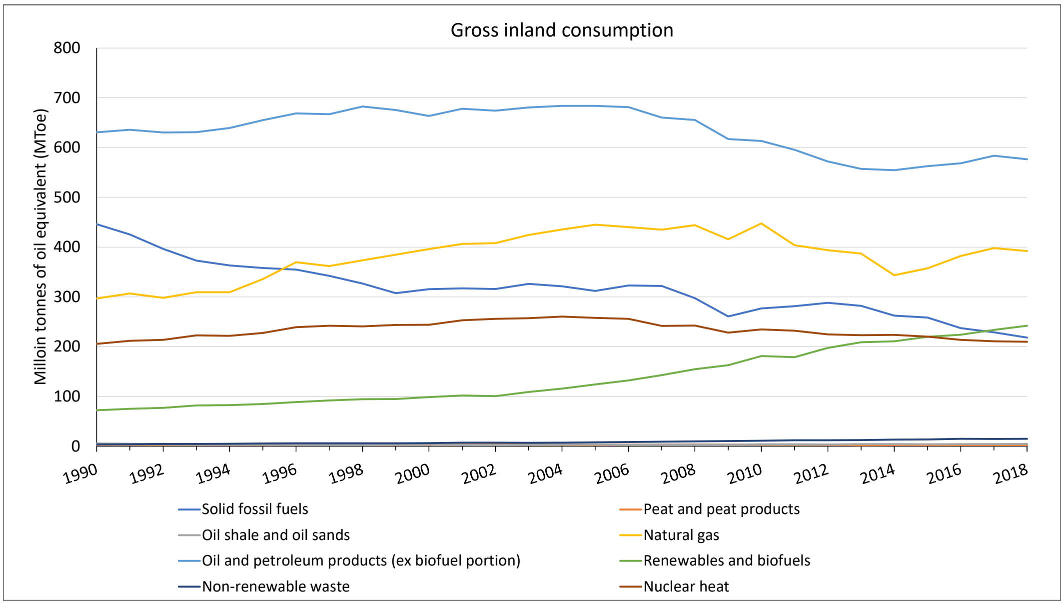


Figure 1.1 – Gross energy inland consumption in EU-28. Source: Eurostat[1]

On the other hand, the global efficiency (electricity and heat) in CHP plants can reach up to 90 % in some applications. There are several types of engines with different well-known configurations with a wide range of heat to power ratios, that allow them to adapt to many different types of demand curves. In the case of Dristict Heat (DH) system with only heat production, the efficiency in the transformation from fuel energy to thermal energy can reach up to 90 %. The drawback of this type of heat plants is the low quality of the transformation, where exergy efficiency is below 18 % [4].

Current energy policies are geared towards carrying out renewable energy projects in an attempt to reduce the carbon footprint and make progress in the fight against climate change. The development of these policies leads to the descentralisation of the electricity production since the energy density of this renewable energy is lower than the traditional sources and the increase of auto-consumption installations. This introduction of new installed power leads also to a shift on the merit order and the decrease of demand (due to auto-comsuption) in the electricity market. This causes a decrease in the load factor of this conventional plants that will operate more intermittently.

The issues above mentioned are the main aspects that lead to the conclusion that LCP will probably become non profitable. However, because of the penetration of these renewable energy technologies, which have an intermittent production (e.g. wind, solar), some plants may be required in the future as back-up to the renewable portfolio and to maintain security of supply [4].

Regarding Dristict Heat, it is referred as the heat that is distributed to final consumers through a network. It can be produced from Combined Heat and Power (CHP) or heat only plants. Most of the DH plants still rely on fossil fuels (table 1.1), but this consumption has decreased in the last few decades. Instead, the share of renewables is increasing, mainly biomass for Dristict Heat, which has been multiplied by more than 3 from 2000 [2]. The operation of CHP for DH is profitable only if sufficient heat is demanded throughout the year. These plants usually work for baseline operation and back-up boiler are used to supply the peak heat demand [5]. However, in small and medium scale of DH systems, this configuration is not usually profitable and the heat demand is covered only by a boiler.

In Belgium, two large scale coal-fired power plants were retrofitted and nowadays, the biomass fuel used are pellets: *Les Awirs* with a capacity of 80 MW_e , and *Rodenhuize*, with a capacity of 200 MW_e and the world's lowest emissions [6].

Heat from:	Total Gross Heat Production	Growth Rate (2016-2017)	**CHP**	**Heat Only**
All fuels	57.628	0.3%	40.271	17.358
Solid Fossil Fuels	13.378	-4%	10.633	2.745
Natural Gas	21.699	0%	14.289	7.409
Oil and Petroleum Products	2.342	-9%	1.688	654
Non-Renewable Waste	3.108	-4%	2.464	644
Manufactured Gases	908	-7%	802	106
Peat and Peat Products	722	-6%	531	191
Oil Shale and Oil Sands	46	-10%	46	0
Nuclear Heat	108	5%	108	0
Electricity	46	12%	11	35
All Renewables	15.271	4%	9.698	5.573
Solid Biomass	10.952	4%	6.593	4.359
Liquid Biofuels	100	-13%	59	40
Biogas	734	8%	624	110
Renewable Municipal Waste	2.905	4%	2.332	573
Geothermal	258	9%	0	258
Solar Thermal	43	25%	0	43
Ambient Heat (Heat Pumps)	280	-7%	90	190

Table 1.1 – Derived Heat Production by Fuel in EU28 in 2017 (ktoe). Source: [2]

As stated above, H&C represents about 50 % of the final energy consumption in EU. 84 % of this share is still generated from fossil fuels while only 16 % is produced from renewable energies. In order to fulfill the EU's climate and energy goals, H&C demand must sharply reduce the consumption of fossil fuels in favour of renewable energies.[7].

1.1.2 Sustainable energy

Currently, in the EU28, most of the biomass energy comes from forest biomass, followed by agricultural biomass and, finally, waste [8]. The use of biomass as an energy source is one of the several contributions to the movement towards sustainable energy. Several aspects should be considered to justify the use of biomass instead of fossil fuels.

Carbon released as CO_2 in the combustion of fossil fuels has been stored underground for hundreds of thousands of years. This implies an increase in the concentration of CO_2 in the atmosphere and its implications in terms of the greenhouse effect and climate change. However, the carbon emitted by biomass combustion was stored on a much smaller time scale, ranging from a few months to several years. This difference is one of the aspects that defines biomass as sustainable: instead of releasing more carbon dioxide to the atmosphere, the CO_2 is absorbed by plants and released again when it is burned, whih leads to a theoretical "net-zero" emissions. A representation can be observed in figure 1.2. As a drawback, the biomass combustion can emit other pollutants in higher quantities than fossil fuels.

The CO_2-neutral or "net-zero" emissions is an aspect to be read with caution. If the biomass burned is not replaced in a short period of time, then this term is not true anymore and the biomass combustion may be even worse than burning fossil fuels. Not only is CO_2 released to the atmosphere, but the organica mass responsible for absorbing this CO_2 is also reduced.

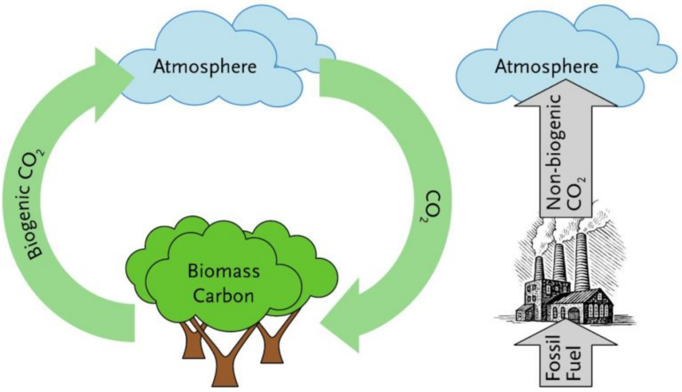


Figure 1.2 – CO_2 cycle. Source: [9]

In 2015 the EU28 land area had approximately 42 % of forest (approxiamtely 182 million hectares) and it is currently increasing, being 5.2 % higher than in 1990. This changes in forest area varies substantially between EU28 Member States. For instance, Portugal was the only country whose forest area decreased slightly, mainly due to fires, while in other countries such as Bulgaria or Spain, the area has increase more than 10 %. Forest fuel and forest management in general are key to prevent forest fires [8].

Despite what can be presumed about the forest management and the possible deforestation, it has been proven to be effective against fire and the growth of forest area since its exploitation could be profitable for the managers. Some figures about this aspect can be seen in figure 1.3.

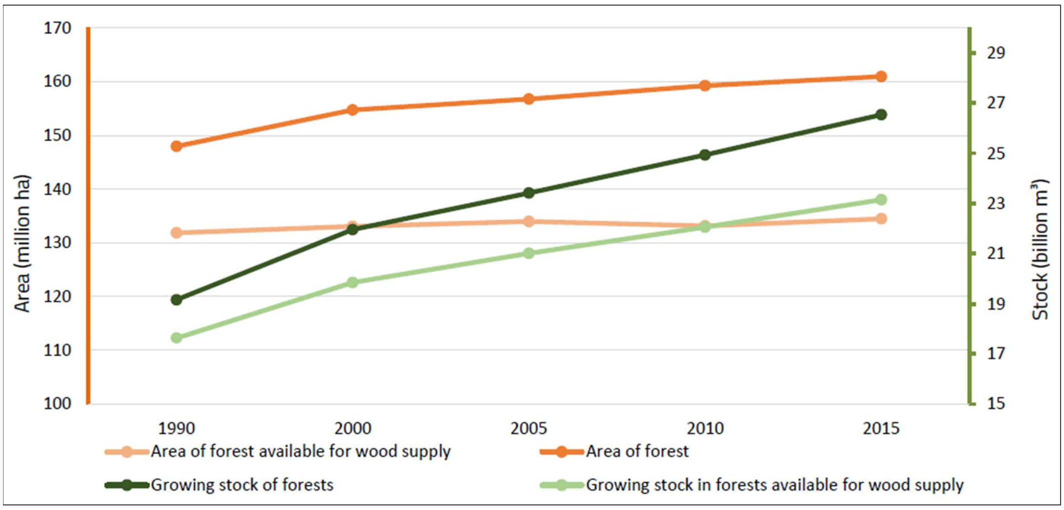


Figure 1.3 – Evolution of total area (left axis) and available stock (right axis) of forest and forest availabe for wood supply in EU28 (million hectares and billion m^3) Source: [8]

Current land use in the EU for the production of dedicated energy crops is still marginal, but this type of energy is a primising form of bioenergy: it has very low input requirement and short carbon cycles [8]. Nevertheless, this change in the explotation of agricultural lands can lead to a competition to other uses such as food products.

Compared to other renewables energies and apart from the above mentioned about CO_2-neutral, biomass has important advantages over other renewable energies: it is always available for electricity or heat production in contrast with other sources such as wind or solar energy. Biomass has also higher energy density and it can be dispatched in many ways: liquid fuel for transport, pellets for boilers, bio-gas, etc. These characteristics are what give biomass the ability to replace oil-based fuels and coal in many applications.

As a summary, the introduction of biofuels as remplacement for fossil fuels results in a large drop in carbon dioxide (CO_2) emissions, taking into account also emissions due to transportation of the biomass. Considering the bio-energy sustainable or non-sustainable depends on aspects such as the management of the biomass and its remplacement.

1.2 Biomass. Main Concepts

The definition of biomass can be very broad, but on the energy side, it is defined as any organic matter from plants or animals and is thus a renewable energy source. The biomass can be solid, liquid or gaseous form including or not fuel processing steps [8]. Biomass is a solar energy storage achieved through photosyntesis, in which CO_2 and water are converted into organic matter and O_2 through an input of energy. This carbon dioxide (CO_2) is released again into the atmosphere during combustion.

1.2.1 Classification and composition

Biomass can be classified in many ways. Based on the valorisation of biomass and its origin, one can distinguish between primary, secondary and tertiary biomass:

- **Primary biomass**. It is that which comes directly from forestry and agriculture. This group includes energy crops and the wood used directly as an energy source. This type of biomass generally does not make a large contribution to the total biomass used for energy production due to its greater added value for other sectors such as furniture in case of wood from forestry.

- **Secondary biomass**. This group is made up of by-products of the first industrial conversion of primary biomass such as bark, sanwdust or wood chips.

- **Tertiary biomass**. It is the waste biomass. This group includes the biomass that comes from diferent sources such as wood from demolition, Municipal Solid Wastes (MSW), sewage sludge.

Another general classification can be seen in figure 1.4. In this diagram, the biomass is classified according to the four sources before mentioned and then, the classification in primary, secondary or tertiary. Energy crops deserve especial mention as its sole purpose is to be a source of energy in contrast with forestry or agricultural biomass, whose main contributions to energy generation are from by-products or secondary and tertiary biomass.

Focusing now in woody biomass, its chemical analysis shows that it is composed of the following components [10]:

- **Cellulose**. Represents $30-40$ % of the mass of dry woody biomass. This molecule is formed as a result of natural polymerization of glucose molecules in a large number, around $5000-10000$ units per molecule of cellulose. It is resposible for the strength of the woody biomass since these microfilms of cellulose twist around each other to

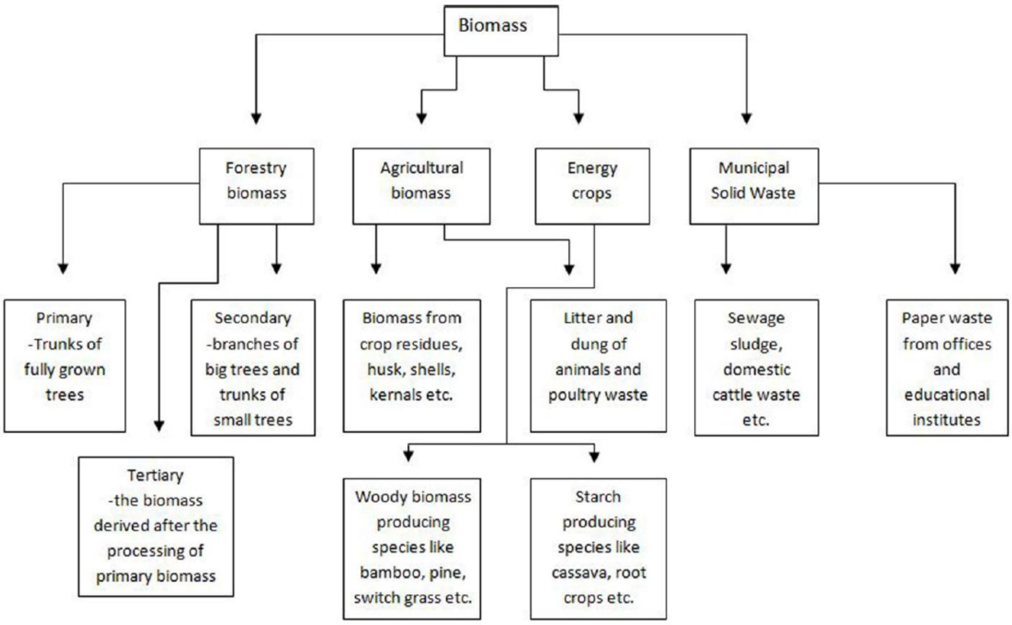


Figure 1.4 – Biomass classification. Source: [10]

form a tubular structure. The water (humidity content) is contained in this structure. When brokendown ($240°C - 360°C$), most part of the cellulose is converted into volatile compounds.

- **Hemicellulose**. Around $25 - 30$ % of the biomass in dry basis is comprised by this component. In contrast to cellulose, it contains only around 150 units of monomers and possesses a relative low weight. The chain of this heterogeneuos polymer consists of a branched structure. In the decomposition process ($200°C - 260°C$), there are more volatile components formed compared to tars and chars. Also, this volatile yield component is greater than in cellulose formation.

- **Lignin**. Its share in the composition depends on the type of biomass, varing from 16 % in hard biomass material to 33 % in the case of softer biomass materials. This component breaks down at a temperature range of $280°C - 500°C$ and forms phenols and a greater percentage of char. When pyrolysis, the liquid obtained is formed by 20 % water and 15 % residual chars.

- **Inorganic minerals.** They are found in a very low percentage and they form an amount of ash. These minerals are usually potassium, sodium phosphorus, calcium and magenisium.

- **Organic components.** They are found in traces in woody biomass materials and includes sugars, fats, proteins, waxes, etc... that help the plants get immunity against insect attacks and as used as energy reserves.

Lignocellulose is the matter composed by lignin, cellulose and hemicellulose. It is the most abundant available raw material for the production of biofuel. The proportion of each compotent varies depending on the types of plants. Some examples can be seen in table 1.2.

Feedstock	Rice straw	Wheat straw	Switch grass	Hard wood (hybrid poplar)	Soft wood (pine)	Waste paper
Cellulose [$\%_{db}$]	35	38.2	45	44.7	44.6	76
Hemicellulose [$\%_{db}$]	25	21.2	31	18.6	21.9	13
Lignin [$\%_{db}$]	12	23.4	12	26.4	27.7	11
Other [$\%_{db}$]	28	17.2	12	10.3	5.8	0

Table 1.2 – Organic components of some lignocellulosic biomass. Source: [11]

Through the proximate and ultimate analysis, a biomass matter can be defined in terms of composition. Proximate analysis is the characterisation of fuel as a function of the by-products during combustion process. In this way, three main components can be distinguished: moisture content, volatiles and fixed carbon, and ashes. Volatiles and fixed carbon, i.e. the valuable components for the energy production, form what is known as fuel Dry ash free basis (daf). The sum of these components and ashes is named fuel dry or Dry Basis (db) and the sum of all the components is the fuel As Delivered basis (ad) or Wet Basis (wb). A graphic explanation can be seen in figure 1.5.

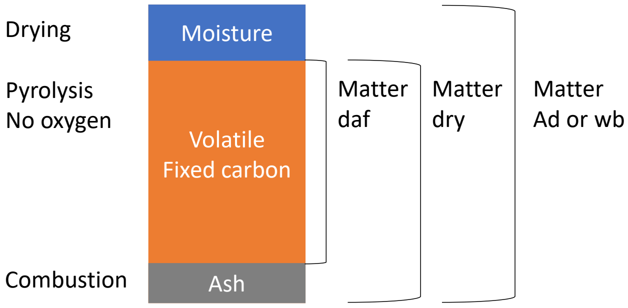


Figure 1.5 – Proximate analysis

Ultimate analysis is the determination of the weight percent of the diferent elements that form the fuel, mainly carbon, hydrogen, sulfur, nitrogen and oxygen and it is measured using specialized laboratory equipment [12]. The composition is given in Dry Basis to remove the hydrogen and oxygen that come from water.

Some examples of these two analysis described above can be seen in table 1.3. As can be obserbed, in view of the proximate analysis, the compositions of Dry ash free basis fuel for biomass are very similar, while the ash content varies greatly depending on the type of biomass. Compared to coal, biomass contains much more volatiles and, in general, less ash content.

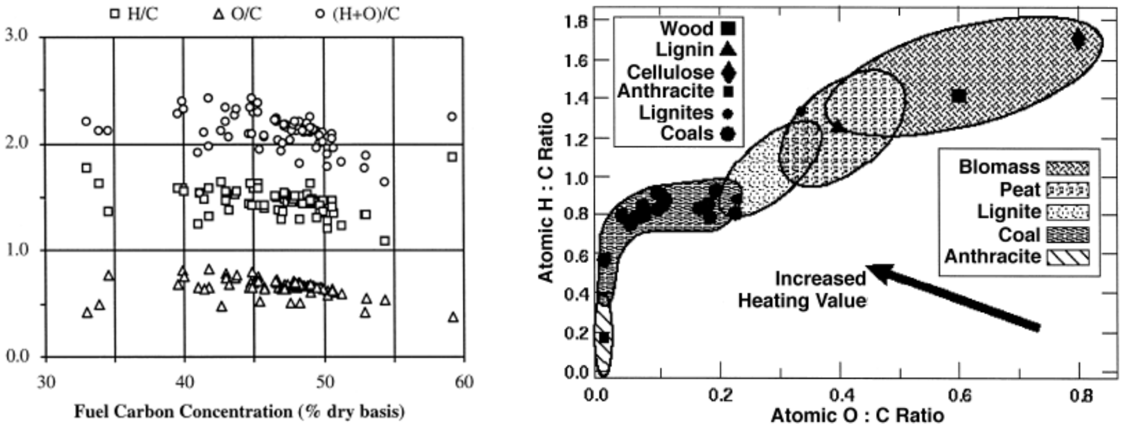
Concerning ultimate analysis, there is a huge difference in the weight of carbon between biomass and coal, which leads to a higher ratio of hydrogen-carbon and oxigen-carbon and a lower Low Heat Value (LHV). It is also significant the difference in sulphur weight. This is an important advantage of biomass in terms of emissions compared to coal.

Acording to [14] and its results obtainend in figure 1.6a from different types of biomass as well as the values in table 1.3, it can be concluded that the H/C and O/C ratios for biomass stay almost costant regardless of carbon concentration in fuel. In figure 1.6b, the difference in composition between coal and biomass above mentioned can be seen.

In view of the above and neglecting the weight content of Nitrogen and Shulfur, a first

Proximate analysis	Wood	Grass / Plant	Husk/ Shell / Pit	Straw	Organic residue	Refused derived fuel	Coal
Volatiles [$\%_{daf}$]	81.8	82.6	76.9	81.3	85.0	88.7	39.8
Fixed carbon [$\%_{daf}$]	18.2	17.4	23.1	18.7	15.0	11.3	60.2
Ash [$\%_{daf}$]	2.7	7.0	5.8	7.5	10.5	17.0	11.1
Ultimate analysis							
C [$\%_{daf}$]	50.8	49.1	50.3	48.6	49.3	54.8	79.0
H [$\%_{daf}$]	6.06	5.98	6.24	5.96	6.52	8.12	5.06
O [$\%_{daf}$]	42.5	43.5	42.4	43.4	41.8	34.0	13.7
N [$\%_{daf}$]	0.51	1.24	1.18	0.9	2.36	0.94	1.48
S [$\%_{daf}$]	0.08	0.17	0.15	0.16	0.63	0.40	1.66

Table 1.3 – Proximate and ultimate analysis of some types of biomass and coal [13]



(a) Molar ratios of H and O in biomass

(b) Van Krevelen diagram

Figure 1.6 – Ratios of composition for biomass (a) and representation of different biomass and coals (b). [14]

aproximation of the formula for biomass can be $CH_{1.44}O_{0.66}$ if no more information is provided. This formulation is widely used and some examples of its use can be seen in references [15–17].

From the study of the composition of the biomass, either by ultimate or proximate analyses, it is possible to define the value of High Heat Value (HHV). A simple formula based on ultimate analysis can be seen in equation (1.1). A more advanced formula is the Boie correlation and can be seen in equation (1.2). It can be noted that the oxygen is withdrawn with the HHV. Different empirical correlations based on proximate and ultimate analyses of biomass can be seen in [18]. Normal values for the biomass compositions above mentioned are around 20 MJ/Kg_{daf}.

$$HHV_{daf} = 45.7w_C - 2.7[MJ/kg_{daf}] \tag{1.1}$$

$$HHV_{daf} = 35.17w_C + 116.2w_H - 11.1w_O + 10.47w_S + 6.28w_N - 0.439[MJ/kg_{daf}] \tag{1.2}$$

HHV refers to the energy of the exhaust gas (after biomass combustion) considering the condensation of water. In fuels such as biomass, this water does not only come from hydrogen in the fuel, but also from moisture, since there is normally high moisture content. This is an important fact to bear in mind when considering the exhaust gas as a heat source in a dryer or in a Flue Gas Condenser. As it can be seen in equations before mentioned, the energy units are given per mass of fuel daf. In this case, the water condensation considered is only that which comes from the hydrogen present in the fuel.

In biomass boilers, the condensation of the flue gas is not normally considered so in practice, the reference value used is usually the Low Heat Value (LHV), where condensation is not considered. For a general composition of the fuel CH_yO_x, LHV_{daf} can be defined by equation (1.3), where L_v is the latent heat value of the water at the pressure of the boiler and its value is around 2.5 MJ/kg. Normal values are around $15-20$ MJ/kg_{daf}.

As mentioned, biomass is usually the substitute for coal in combustion plants. Common values of LHV for coal is around $25-30$ MJ/kg_{daf}. This issue is an important handicap in the retrofitting from coal to biomass.

$$LHV_{daf} = HHV_{daf} - Lv \cdot \frac{y}{2}\frac{18}{12+y+16x}[MJ/kg_{daf}] \tag{1.3}$$

It is important to distinguish between the water formed due to the combustion by the oxygen and hidrogen molecules released, and the water contained in the fuel. The latter is the water that must be removed from the fuel by evaporation before combustion. This means that part of the chemical energy contained in the biomass (LHV_{daf}) will be used to evaporate this water. Equation (1.4) defines the LHV considering this consequence and the dillution effect of the moisture and ashes. The impact of ash is purely dillutive as it is considered to be inert. MC_{daf} and Ash_{daf} are the weight content of moisture and ash, respectively, in dry ash free basis, obtained from proximate analysis.

$$LHV_{ad} = \frac{LHV_{daf} - L_v \cdot MC_{daf}}{1 + Ash_{daf} + MC_{daf}}[MJ/kg_{ad}] \tag{1.4}$$

1.2.2 Moisture content

The moisture in biomass can remain in two forms: free (or external) and inherent (or intrinsic or equilibrium) [19]:

- **Free moisture**. It is that which is above the equilibrium moisture content. It generally resides outside the cell wall. It depends on the harvest and storage condition [20].
- **Inherent moisture**. It is the moist absorbed within the cell walls.

When the walls are completely saturated, the biomass is said to have reached the fiber saturation poin (FSP) or equilibrium moisture. It is defined as the moisture content at which the wood is neither gains nor loses moisture [21]. The value of this FSP is strongly dependent on the relative humidity and weakly on the air temperature [19] when the temperature is close to ambient temperature. This variation can be seen in figure 1.7, where the FSP of wood is plotted as a function of RH for several temperatures. The correlations used are those proposed in [21]. According to the authors, these values may be applied to wood of any species for most practical purpose. The correlation plotted is referred to a isotherm sorption.

Another way to classify the moisture is according to the vapor pressure of the moist [22]:

- **Bound moisture**. This term refers to the water retained that exerts a vapour pressure less than that of free water at the same temperature. Such water may be retained in small capillaries adsorbed on surfaces, or as a solution in cell walls.

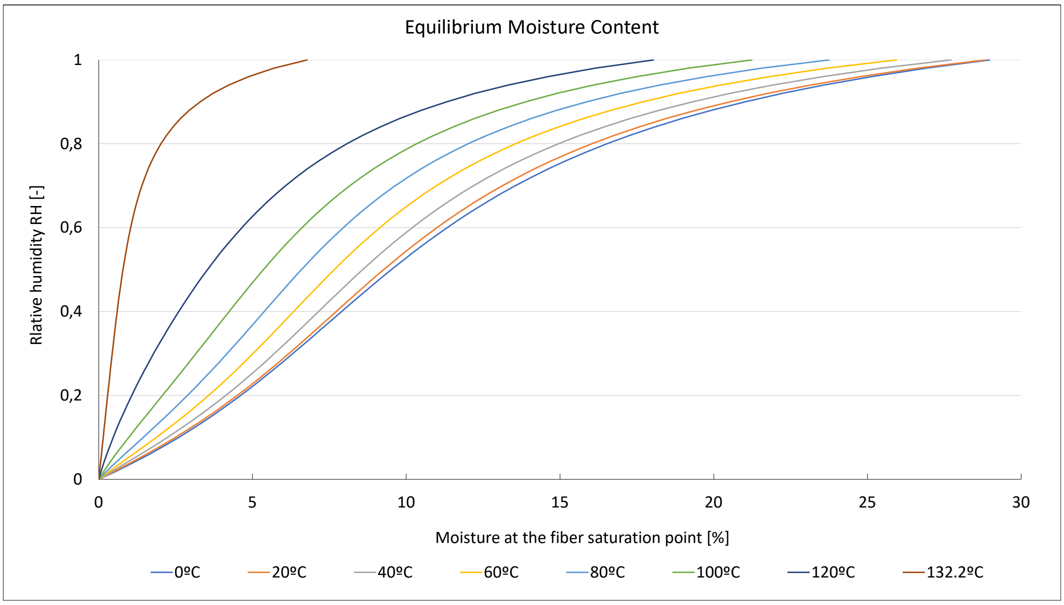


Figure 1.7 – FSP of wood as a function of relative humidity for some temperatures

- **Unbound moisture**. This is the amount of free moisture which is in excess of the equilibrium moisture content corresponding to the saturation humidity. It is the water present in the surface of the biomass.

In summary, at a certain temperature and relative humidity conditions, a generic biomass has an equilibrium moisture corresponding with the saturation of the fiber. If this ambient conditions change, biomass will reach the new corresponding equilibrium by absorbing or releasing water from or to the ambient air. If the moisture content of the biomass increase over the equilibrium moisture by adding more external water to the biomass, it will reach the maximum moisture bound. In this condition, all the water content below this value has a pressure vapor below the normal pressure for the liquid at the external/ambient conditions. Above this value, moist is present over the biomass surface in liquid form. Theses steps can be seen in figure 1.8.

Biomass moisture content can be obtained by the weight difference before and after drying a biomass sample. There are different test protocols for some types of biomass. In ASTM standards, it can be found $D-871-82^{40}$ for wood, $D-134894$ for cellylose, $D-1762-84$ for wood charcoal and $E-949-88$ for Refuse-derived fuel (RDF). [20]

As can be seen in the previous section, the moisture content plays an important roll during biomass combustion. The presence of moisture in the fuel leads to a decrease in the LHV. Accoding to [23], some industrial boilers require a LHV to be above 15 MJ/kg_{wb}. Auto-thermal and self-supporting combustion are also affected, being the limit of moisture content around 65 $\%_{wb}$ according to [24, 25]. Some examples of biomass matters, their moisture content and heat values can be seen in table 1.4.

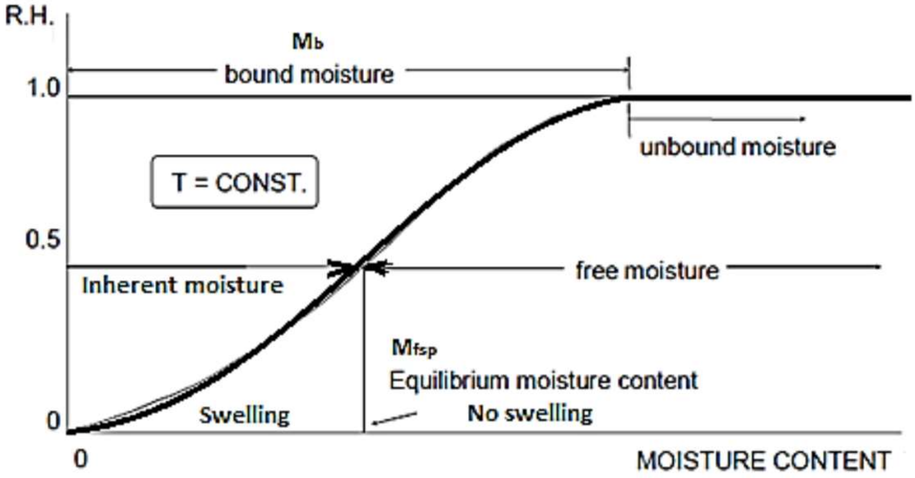


Figure 1.8 – Various type of moisture content [20]

	Moisture content $[wt\%_{wb}]$	HHV $[MJ/kg_{db}]$	$LHV[MJ/kg_{db}]$
Wood pellets	10	19.8	16.56
Wood chips - hardwood - pre-dried	30	19.8	12.24
Wood chips - hardwood	50	19.8	7.92
Wood chips - softwood - pre-dried	30	19.8	12.24
Wood chips - softwood	50	19.8	7.92
Grass - high-pressure bales	18	18.36	13.68
Bark	50	20.16	8.28
Trticale (cereals) - high-pressure bales	15	18.72	14.4
Sawdust	50	19.8	7.92
Straw (winter Wheat) - high-pressure bales	15	18.72	14.4

Table 1.4 – Moisture content and heat values of some biomass matters [23]

1.3 Biomass Boilers. Main Concepts

Boilers designed for biomass combustion are not so different from coal boilers. Both fuels are carbonaceous, can be crushed or pulverised, they are both storable and can be found almost everywhere. However, there are important differences such as the organic and inorganic composition. As previously stated: biomass has a lower LHV but more volatiles (up to 90% compared with $5-40\%$ of coal). Despite these differences, retrofitting to biomass is possible in coal-fired power plants, however some operational issues must be considered:

- **Milling stage:** Biomass matter is more fibrous and less brittle than coal, which results in larger particle size (pulverized-fuel boilers). This issue reduces the capacity of the boiler since the time required for the total combustion of larger particles is longer.

- **Furnace:** The capacity of the plant is reduced also due to the energy density variation. In case of retrofitting, the volumen of the boiler is already defined. As biomass has less energy density (lower LHV), a higher combustion rate would be required to meet the capacity of the carbon-fed plant.

- **Furnace and heat exchanger:**. Due to the higher volatile composition, there is an increase in fly ashes that reach the exchanger at high temperature. Moreover, ash released during biomass combustion has a lower initial melting point [4]. This causes slagging, fouling and bed agglomeration.

In case of woody biomass, another aspect that should be considered is that the dust from wood is highly explosive due to volatile content. This requires a important control in stages such as unloading or milling, where the presence of metals and the friction with mills and electrostatic electricity can produce sparks.

1.3.1 Combustion process

The combustion process of the biomass that enters the boiler until the flue gas reaches the exchanger includes four stages: drying, pyrolysis, gasification and combustion itself. Drying and pyrolysis/gasification are stages proper of a solid fuel combustion process. The relative importance of these steps will vary, depending on the combustion technology implemented, the fuel properties and the combustion process conditions [23].

- **Drying.** Moisture will evaporate at low temperatures, below 100 $°C$. the process of evaporation requires energy and this comes from the combustion of the biomass itself. As a result, the temperature in the combustion chamber will be lower as the fuel humidity is higher. This effect also leads to the slow down of the combustion process. Above a certain value of moisture, around $60-65\ \%_{wb}$, the combustion process can not be maintained. Above these values, much energy is required for evaporation and the temperature drops below the minimum required to sustain combustion.

- **Pyrolysis.** It consists of the thermal degradation (devolatilisation) in absence of an externally supplied oxidising agent. The products are mainly tar and carbonaceous charcoal, and low molecular weight gases, besides CO and CO_2 in case of high oxygen content in the fuel. Pyrolysis is affected by different fuel properties, temperature, pressure, heating rate, reaction time and properties of the products formed. Temperatures reached are around $400-500°C$

  Instead of continuing to burn, these products can be used in a variety of ways. The char can be upgraded to activated carbon for metallurgical industry, domestic cooking fuel, etc. Gas can be used for heat production or power generation, or synthesised to produce methanol or ammonia. Part of the liquids can be upgraded to high-grade hydrocarbon liquid fuels.

- **Gasification.** It is the thermal degradation, this time in the presence of an external oxidising agent. While pyrolysis is usually optimised regarding a maximum char or tar yield, gasification is optimised with respect to maximum gas yield. Temperatures of $800-1100°C$ are reached. This gas contains mainly CO, CO_2, H_2O, H_2, CH_4 and other hydrocarbons. In processes with a different intention than combustion, the oxidising agent can be different from the air, such as pure oxygen or steam, that produces a higher HHV gas than with air.

- **Combustion.** It can be defined as a complete oxidation of the fuel. The hot gases from the combustion may be used for direct heating purposes, indirectly heating using water or the production of steam for power generation. Combustion is always preceeded by pyrolysis/gasification in a solid-fuel combustion.

1.3.2 Types and classification

There are mainly two ways to classify boilers: in function of the size and the legislation applied, or according to the technology. Depending on the size and the legislation applied, it is possible to distinguish three stages:

- **Small scale** ($< 1\ MW_{th}$)**:**. There is not a severe legislation in terms of emissions but some european directives focus more on ecodesign and energy labelling. There are no secondary measures[1] applied.
- **Medium scale** ($1 - 50\ MW_{th}$)**:** In section 1.1.2, it was mentioned the expected increase in medium scale biomass-fired combustion plants due to the introduction of renewable energies and decentralization of electricty production. As a result, a rencent Directive (Medium Combustion Plan Directive (MCPD)) was developed in 2015.
- **Large scale** ($> 50\ MW_{th}$)**:** The Large Combustion Plan Directive (LCPD) was uploaded in 2001 and the Emission Limit Values (ELV's) are becoming more and more restrictive. For this reason, secondary measures are always applied. $DeNOx$, mainly Selective Catalytic Reduction (SCR) and Selective non Catalytic Reduction (SNCR), $DeSOx$, such as FGD, and *Dedusting* systems, such as Electrostatic Precipitator (ESP) and Fabric filters, are needed.

In figure 1.9, different principle technologies of boilers can be seen [26]. These elements will be summarized below according to [23]:

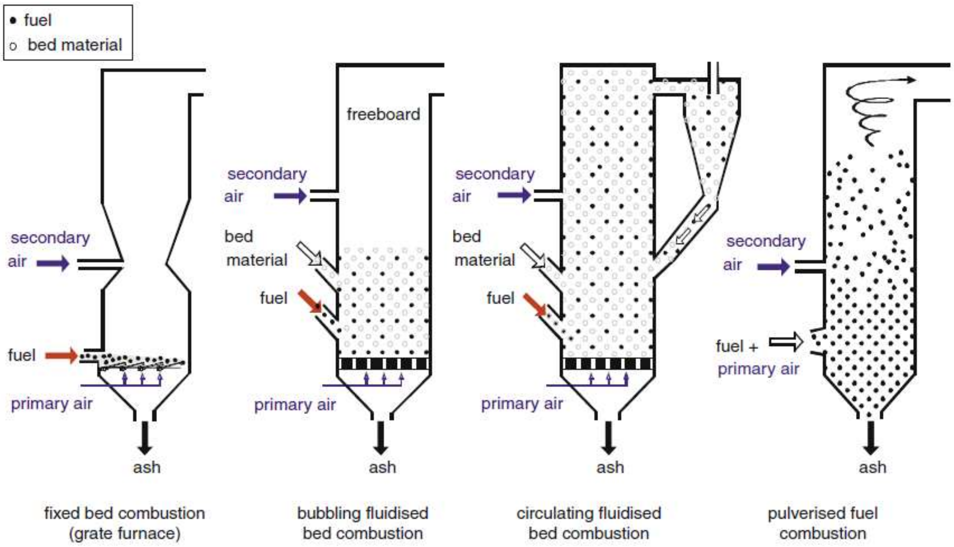


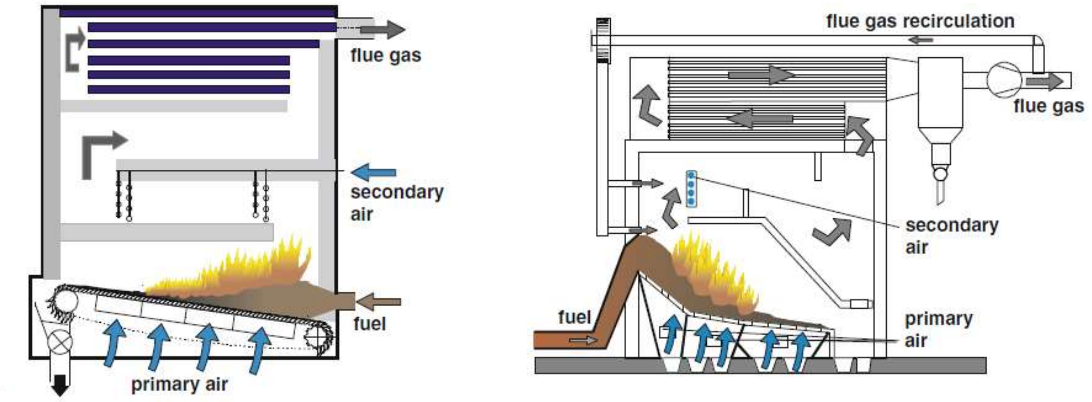
Figure 1.9 – principle combustion technology for biomas [26]

- **Fixed bed/ grate furnace.** The capacity range for this type of boilers varies from 100 kW up to several hundred MW_{th} [26]. The basic principle of operation is the passage of a primary air upwards through a perforated grid where the solid fuel is deposited. Over this furnace, drying, gasification and charcoal combustion take place.

---

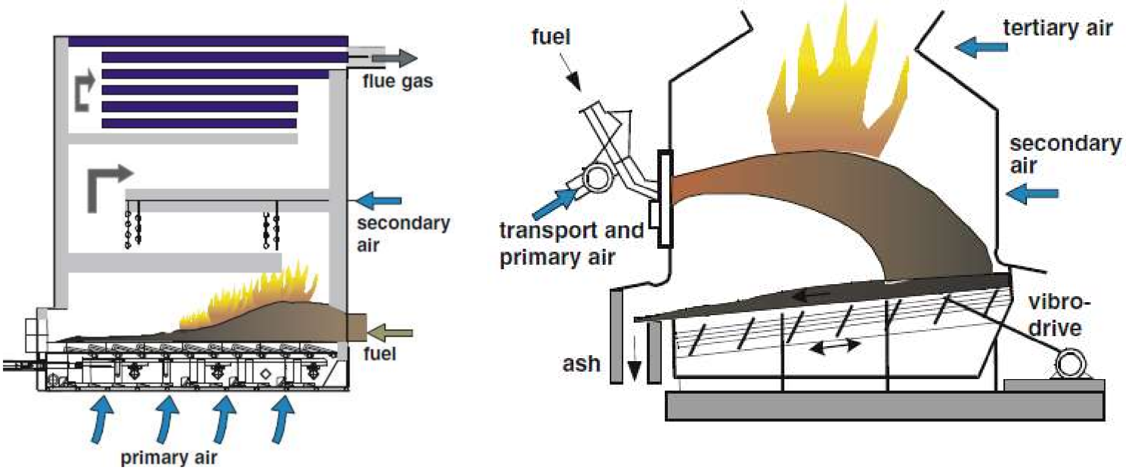
[1] secondary measures is the system installed downstream the boiler in order to reduce pollutant emission in contrast to primary measures, which consists on the control of combustion parameters in order to avoid the pollutant formation

Adding a secondary air out of the bed caused gases produced during gasification to burn. This technology is appropiate for biomass fuels with a high moisture content. It allows a smooth operation of grate furnaces at partial loads of up to a minimum of about 25 % of the nominal load. There are different types of grates and fuel feeding systems with specific advantages and drawbacks depending on fuel properties. A representation of some of them can be seen in figure 1.10.



(a) Traveling grate

(b) Inclined moving grate



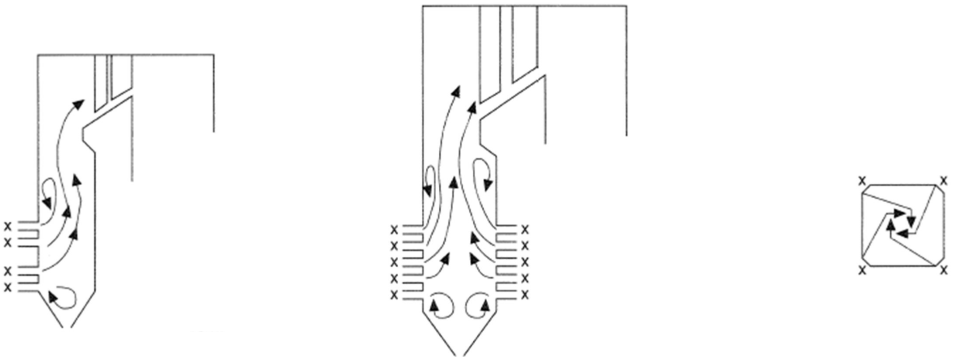
(c) Horizontal moving grate

(d) vigrating grate and spreader strokers

Figure 1.10 – Different grates furnace technologies. [26]

- **Pulverized/dust fuel boilers.** They are usually for large scale solid fuel combustion ($100-2000\ MW_{th}$). Suitable for fuels available as small particles (average diameter smaller than 2 mm). A mixture of fuel and primary air combustion is injected into the combustion chamber, where the fuel is burned in suspension. Secondary air is injected at the burners level and there may be even a third level of air staging and over fire air. Fuel moisture content should not exceed 20 $\%_{wb}$. Due to fire hazard of dust wood, the fuel feeding needs to be controlled very carefully. In this case, charcoal and gasification take place at the same time because of the small particle size.

  The usual arrangement of the burners is horizontal, although it is also possible to find burners in a vertical position. Depending on the fuel used and the design conditions, three different horizontal arrangements of burners can be distinguished: front wall, opposed walls and tangential. These assemblies can be seen in figure 1.11.



(a) Front wall (b) Opposed walls (c) Tangential

Figure 1.11 – Burners configurations. [27]

- **Fluidised bed combustion.** Biomass fuel is burned in a self-mixing suspension of gas solid-bed material into which combustion air enters from the bottom. The bed material represents $90 - 98$ % of the mixture. Depending on the fluidization velocity, bubbling fluidized bed and circulating fluidized bed combustion can be distinguished. Due to the good mixture achieved, there is flexibility in the fuel mixtures, yet limited by the fuel particle size. In addition, partial load operation is limited due to the need of bed fluidisation.

  There are two main types of Fluidised bed (FB): Bubbling Fluidised bed (BFB) and Circulating Fluidised bed (CFB). BFB, interesting in plants $< 20\ MW_{th}$ and with more flexibility in the particle size and moisture content of biomass fuels, but with difficulties at partial load operation. CFB has higher turbulence due to higher fluidising velocity ($5 - 10\ m/s$ compared with $1 - 2.5\ m/s$ of BFB). This leads to a better heat transfer and a very homogeneous temperature distribution in the bed. However, this type of boiler has a larger size, which usually means higher investment cost. The particles size required ($< 40\ mm$ in diameter). These boilers also have problems at partial load operation.

1.4 Biomass Dryer. Main Concepts

1.4.1 Drying process

The drying process of biomass can be splited into 3 stages: warming-up, constant rate period and falling rate period [20, 28–34]:

When the solid fuel enters in a dryer in the case of a continous dryer, or the drying process begins in case of batch dryers, this fuel is heated up to the wet bulb temperature to produce a driving force for water to leave the wet material. This stage is usually the shortest.

Once the moist reaches the wet bulb temperature, the constant drying rate period begins. This period is usually fairly quickly. During this stage, the water removed is that free moisture present on the fuel surface. Vaporization takes places from this surface and some shrinkage might occur as the moisture surface is drawn back toward the solid surface. In this stage of drying the rate-controlling step is the diffusion of the water vapor across the drying agent[2]-moisture interface and the rate at which the surface for diffusion is removed. Toward the end of this period, moisture has to be transpoted from the inside of the solid to the surface by capillary forces and the drying rate may still be constant. The drying rate remains constant while water is supplied by mass transfer from within the solid to the surface at a sufficient rapid rate. This mass transfer keeps the biomass surface saturated. This phase lasts until the moisture content has reached the critical moisture content, X_{cr}.

The critical moisture content is thus identified as the average moisture content of the solid at the moment when the first increment of dry area appears on the surface of the solid. This value depend on the ease of the moisture movement through the solid, and hence, upon the pore structure of the solid, sample thickness and drying rate. As a result of reaching this critical moisture content, the biomass surface is not fully saturated. At this point the surface film of the moisture has been so reduced by evaporation, that further drying causes dry spots to appear upon the surface.

When first dry spots appear, the falling rate period begins. This period can be divided in two phases. The first one is the unsaturated surface drying phase. It begins when dry spots appear in the surface, i.e. after the critical moisture content. Despite dry spots, the drying rate is computed with respect to the overall solid surface area. Hence the drying rate falls even though the rate per unit wet solid surface area remains constant. This phase lasts until the surface film of the liquid is entirely evaporated.

The second phase of the falling rate period lasts until the moisture content reaches the equilibrium, which depends on relative humidity and temperature and not on the drying rate. At this point, the vapor pressure over the solid is equal to the partial pressure of the vapor in the drying agent. At this stage, the drying rate is controlled by diffusion of moisture from the inside to the surface and then mass transfer from the surface. Heat must be transfered by conduction to the remaining water further inside the solid; in effect, the evaporating surface recedes into the material as drying proceeds. Because the dried solid near the surface is generally a poor conductor of heat, the rate of heat transfer decline progressively. Figure 1.12 is a representation of the complete drying process.

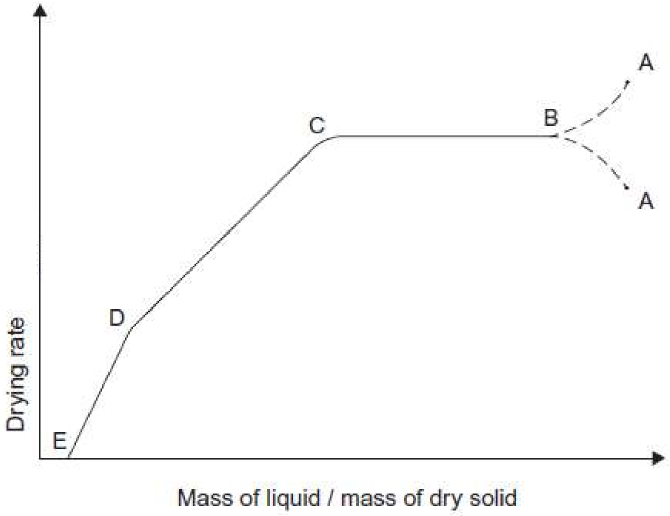


Figure 1.12 – Drying rate curve as a function of the moisture content in Dry Basis [34]

In figure 1.12, Segment AB corresponds with warming-up or the phase when the

---

[2] The drying agents usually considered are air, flue gas or waste gas as a result of an industrial process, and superheat water vapour

biomass reach the wet bulb temperature. Segment BC is the constant drying rate period. Point C represents the critical moisture content. The two falling rate periods correspond with segments CD and DE. In D, the surface of the biomass is considered completely dry while point E is the equilibrium moisture.

1.4.2 Types and classification

Dryers can be classified into two big groups based on the interaction between the drying agent and the dried matter: direct and indirect dryers. In the former, the heating medium is in contact with the product to be dried and they are the most commonly used in the industry. In indirect drying, the heating medium does not come into contact with the product being dried. Instead, wet material is dried by contact with a heated surface; heat transfer to the wet material occurs mainly by conduction from this surface. One important consequence of indirect drying is that it is possible to recover the latent heat of evaporated water as the water vapor is not diluted by the drying agent.

Regarding the direct dryers, the most common ones will be described below [28, 29]:

- **Rotary Dryers.** It is the most common type in chemical engineering and for biomass drying. In this type of dryer, hot gases are in contact with the biomass material inside a rotating drum. The drum or cylindrical shell is normally slightly tilted to the horizontal. The rotation of the drum, with the aid of flights, lifts the solids in the dryer so they trumble through the hot gas, promoting better heat and mass transfer.Their robust and simple construction combines flexibility with reliability, enabling this type of dryer to operate under the most arduous conditions [31]. Co-current and counter-current flow are both used equally. The use of one or the other depends on the properties of the dried matter.

  Co-current flow is used for heatsensitive materials, even for high inlet gas temperature, due to the rapid cooling of the gas during initial evaporation of the surface moisture. With this configuration, the hottest gases come in contact with the wettest material. For other materials, counter-current flow are more suitable to take advantage of the high thermal efficiency that can be achieved in this way. This configuration produces the lowest moisture leaving the dryer, but the matter can reach higher temperatures, which increases the fire risk.

  The temperature of the drying agent at the inlet of the drum varies in the range $230 - 1,092\ °C$ while the outlet temperature varies from $71\ °C$ up to $110\ °C$, but most dryers have outlet temperatures above $104\ °C$ to prevent condensation of acids and resins. Retention time varies from less that an minute for small particles up to $10 - 30\ min$ for larger materials. A diagram of the most widely-used rotary dryeris can be seen in figure 1.13.

- **Flash Dryers.** Also called pneumatic dryers. The drying phenomenon is based on the mixture of the solids with a high velocity hot air stream (or other drying agent, such as waste heat in form of flue gas). This causes a very rapid drying process. These types of dryers can be used for drying only small size biomass particles. Generally, the drying medium is superheat steam. It is introduced from the bottom and biomass particles are fed from a side port and trasported up by the drying agent to the cyclone, where the dried particles are separated and the driying medium is re-ciculated [35].

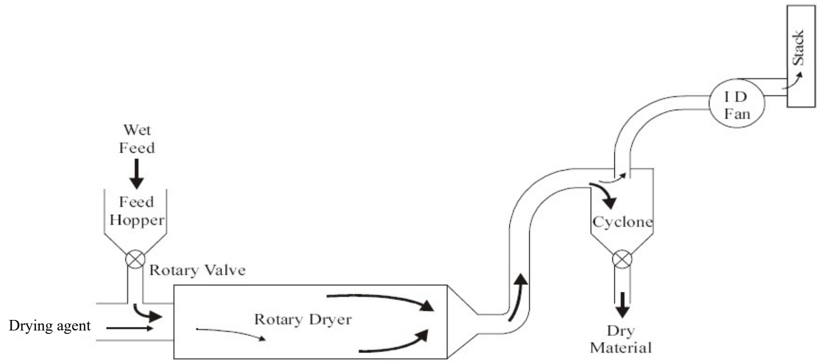


Figure 1.13 – Diagram of a Single-Pass Rotary Dryer. Source: [28] with modifications

Because of the short drying time, the equipment is more compact compared with rotary dryeres but the electricity consumption is higher because of the faster air flows and due to the biomass particle which must be reduced by a shredder or grinder in order to be suspended in the stream. Gas temperatures tend to be slightly lower than for rotary dryers. These dryers have a lower fire risk due to the shorter retention time and the lower operating temperatures. A representation of this type of dryers can be seen in figure 1.14.

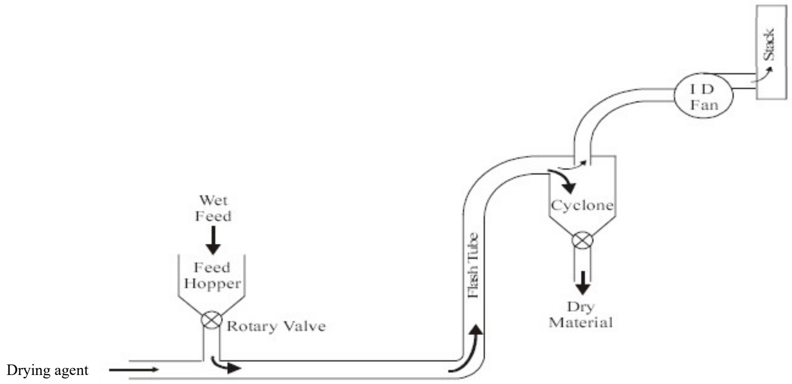


Figure 1.14 – Flash Dryer configuration. Source: [28] with modifications

- **Fluidised bed Dryers.** The fluidized bed is formed by passing a gas stream from the bottom of a bed of solid particulates. At low gas velocities, the bed is static and packed. At a certain gas velocity, the bed is fluidized, which means that the gas stream totally supports the weight of the whole bed. This velocity is called minimum fluidization velocity, u_{mf}. Pressure drop across the bed in the dryer increases as the velocity of the gas increases until it reaches this fluidization state, when pressure reamins constant. At a certain velocity, the bed become a bubbling fuidization bed and the pressure drop decreases.

  The fluidization operating veloctity has a range between $2-4\ u_{mf}$. This minimum velocity is obtained from experiments and can be estimated using various correlations. Wetter matter requires a higher minimum fluidization velocity due to the cohessive force exerted by wet surfaces. Temperatures required for the drying medium are in the range $150-200\ °C$ [31].

- **Conveyor Belt Dryer.** These types of dryer are conceptually very simple. Biomass matter is carred through the dryer on a conveyor and the drying agent (air, waste heat in form of flue gas or superheat steam) is forced through the bed product. It is usually described has a conveyor in a box with hot air.

Conveyor dryers are one of the most versatile dryers available. They can handle such a wide range of products in terms of composition, shape, and size. Although it is simple in concept, an improper understanding of the heat and mass transfer processes in the conveyor dryer will surely lead to poor product handling, wasted energy, and nonuniform product quality.

This technology is best suited for drying material in form of particles with diameters between $1 - 50\ mm$. Its simplicity and versatility allows it to be used to dry matter from a moisture content of 95 %, such as fruits, down to less than 10 % (on wb). Another aspect that makes this type of dryer interesting is the posibility to operate at relatively low drying agent stream temperatures i.e. to take advantage of low grade and waste heat from an industrial production or a thermal power plant. These operation temperatures range from 30 to $200 - 250\ °C$. The lower temperature required of the drying agent compared with other technologies, leads to a lower fire hazard and lower emissions to the air [31].

Moreover, the matter to be dried can be stacked on the conveyor without excessive sticking or clumping. This issue results in a smaller dryer for a given retention time. However, drying time is usually longer compared with other technologies for the same capacity. Typical drying times are $5 - 240\ min$ depending on different aspects, such as, the temperature of the drying agent, its water vapor content and velocity, the height of the bed, etc.

In figure 1.15, a representation of a belt dryer can be observed. In this case, the drying agent flows in cross-current in regard to the movement of the matter to be dried. The change in the direction of this drying agent from being forced upwards through the first section of the dryer and then being forced downwards through the second section of the dryer, leads to a more homogeneous final moisture of the dried matter.

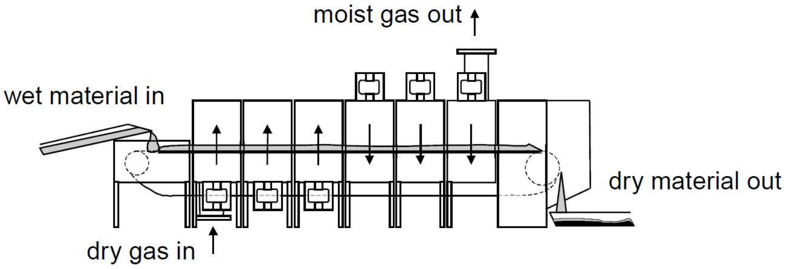


Figure 1.15 – Diagram of a belt dryer. Source: [36]

All the aspects above mentioned and others are summarized in table 1.5. This data is obtained from different analysis done by [20, 31, 35].

	Rotary	**Flash**	**Belt**	**Fluidised-bed**
Evaporation rate (t/h)	3-23	4.8-17	0.4-40	5.0-40
Drying Temp (ºC)	200-600	150-280	30-200	150-200
Capacity (t/h)	3-45	4.4-16	-	-
Fuel moisture at inlet (%)	45-65	45-65	45-72	50-60
Moisture discharge (%)	10-45	10-45	15-25	10-15
Feed Moisture at outlet (%)	-	12	25	-
Pressure drop (kPa)	2.5-3.7	7.5	0.5	-
Particle size (mm)	25-125	0.5-50	0.5-100	0.1-2
Thermal requirement (GJ/t-ev)	3.0-4.0	2.7-2.8	1.26-2.5	2.2
Power consumption	High	Low	Low	Medium
Capital cos (for 1 kg/h of water evaporated)	Medium	High	Medium	Medium
Maintenance cost	High	Medium	Medium	Medium
Energy efficiency	Medium	Medium	High	High
Retention times	From 1 min up to 10-30 min	10-30 sec	-	-
Applications	Sludge, bark/wood chips, residues, sawdust, bagasse. The most common for wood	Wide range of materials. Wet or sticky materials can be recycled to imporve materia handling. Woodchips	Second most common for wood. Sludge, bark, chips, sawdust, bagasse, agricultural products.	-

Table 1.5 – Summary of main characteristics of differenct dryers. [20, 31, 35]

1.4.3 Advantages and Drawbacks

Drying biomass is a process that involves a huge amount of energy consumption and the improvement of its thermal properties may not always be profitable despite the increase in combustion efficiency. This improved efficiency can be impaired by excessive drying. Some advantages and drawbacks are described below.

Concerning thermal aspects, as mentioned in section 1.2.2, for autothermal and self-supporting combustion, the biomass matter must contain a moisture content lower than approximately 65 $\%_{wb}$ [14, 23, 25]. At this moist level, the LHV is much lower compared with the totally dried biomass. Hence, it is possible to take advantage of different types of biomass with high moisture content after been dried. In terms of combustion efficiency, according to [23], drying a fuel from 50 $\%_{wb}$ to 30 $\%_{wb}$ leads to a potential efficiency increment of $8 - 10$ %.

There are some limitations in the water removed from the fuel. Drying below about 10 $\%_{wb}$ may lead to an increase in the CO, what is a tare in the improvement of the efficiency, and the total particle emissions [37]. The less water content also implies an increase in the adiabatic temperature of combustion. This could lead to the formation of NO_x and problems with the fusion of ash from fuel and slagging risk, if its fusion temperature is reached.

A lower amount of water in the biomass fuel means a less total flow of biomass in the boiler, where it is burned. In case of retrofitting, this effect along with the increase in efficiency, could lead to an increase in the capacity of the plant, but this issue is limited by the resistance of the materials from which the boiler is made. In case of a new plant, these benefits lead to smaller boilers, which means less investment costs.

Drying also supposes some operational issues. It is common in indutry to control the excess air in the boiler and the oxygen content as a function of boiler load and moisture of the fuel in order to maintain an acceptable efficiency. The wetter the biomass fuel, the more amount of oxygen needed in order to obtain a complete combustion. This is important for two reasons. On one side, in case of using the flue gas as a drying agent, the increase in O_2 concentration in the drying streams entails an increase in fire hazard during drying. On the other hand, the decrease in excess air mass flow leads to smaller dimensions of the post-treatment device and boiler, i.e. less investment and operational costs.

The profitability of drying biomass is affected by many aspects. The objetive of increasing the heat transfer between drying agent and matter to be dried, can lead either to an increase in the size of the dryers, which implies an increase the investment costs, or the velocity of the drying medium, which implies an increase the operational costs (fans, electricity). These aspects are crucial in the selection of the adequate dryer, besides other parameters.

The economic aspects are also affected by important issues such as running time and fixed cost. Annual revenues for the fuel saved or the extra energy produced as a consequence of drying strongly, depends on the number of hours that the plant is operating. Large investment costs may means that there is no return on investment at the end of lifespan and that even fixed and operating cost even exceed the fuel saved or the extra energy produced.

Chapter 2

Background

In this project, the viability of drying biomass before it is burned for energy purposes is study in a defined scenario. This chapter is dedicated to a literature review and analysis of studies with similar objetives in different scenarios. Differences such as size of the plant, energy purpose (CHP, DH, only power generation) or asumptions considered has been analysed.

2.1 Literature review

In this section, main parameters such as drying agent, its temperature as well as the initial and final moisture content of fuel and others, from different authors and works will be compared and conclusions related with this project will be summarised.

De Fusco et al. [38] studies the economic feasibility of an air belt dryer to enhance the operation of a biomass steam boiler. It is focused on the minimum size for which drying the biomass matter is feasible. In addition to the dryer, a submodel for the steam boiler has been developed in order to study the additional steam produced due to the drying effect.

The biomass matter considered are wood chips with 60 $\%_{wb}$ initial moisture content. Different final moisture content from 10 $\%_{wb}$ up to 40 $\%_{wb}$ and the additional steam produced in each case is studied. The temperature of the air heated is set up at 150 $^{\circ}C$ and decreases down to $30-70\ ^{\circ}C$ at the outlet of the dryer. It is considered that the air is fully saturated at the outlet of the dryer. This drying medium is sent to a Heat recovery unit (HRU) before going to the exhaust. Recirculation of the air is possible in this model, but not considered and set up at 0 %. A diagram of the drying model in seen in figure 2.1.

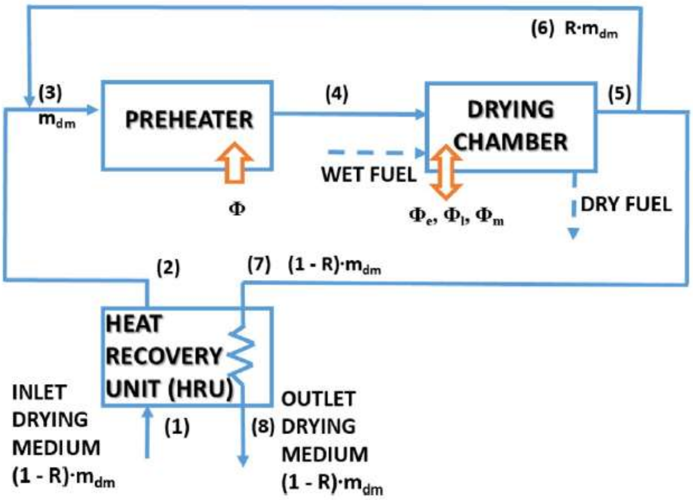


Figure 2.1 – Diagram of the single stage drying system by De Fusco et al.[38]

The study concludes that for a final moisture content of 25 $\%wb$ and 8000 h/y of running time, the size of the plant must be higher than about 1.8 t_{daf}/h. No lower water content is considered because the possible emission increase and fire risk could affect the NPV. Other cases are studied varing the O&M, the cost of steam and the interest rate.

This article is an extension of the Master Thesis by De Fusco [20], where the feasibility of different drying systems with different drying medium (preheated air, flue gas or steam) for small and medium size biomass combustion plants are studied. In the case of using flue gas as a drying medium, its temperature is set at 130 $°C$ and, since the heat exchanger needed to heat up the air or the steam is avoided, this case is the one with the lowest investment cost.

Li et al. [31] evaluate the drying process using waste heat from industry processes. This waste heat ($100MW$) is generated by a process industry plant and consists of either flue gas (40 %) at $250 - 450$ $°C$ or hot water (60 % of the waste heat) at 90 $°C$. Two drying mediums are studied: superheat steam at $140 - 180$ $°C$ and $1 - 2$ $bars$ generated by the flue gas and hot water source or direclty flue gas at different temperatures in the range $250 - 450$ $°C$. the biomass fuel feeds a 40 MW power plant. The type of dryer proposed for the study is a belt conveyor dryer. A diagram of this drying process is seen in figure 2.2.

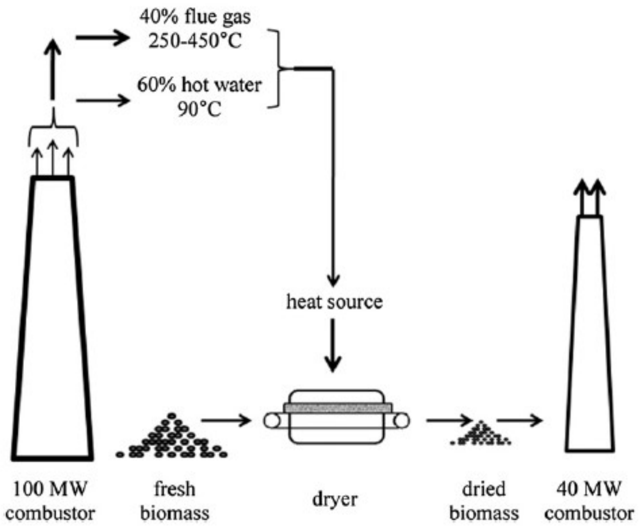


Figure 2.2 – Scheamatic of biomass drying process proposed by Li et al.[31]

The biomass matter considered a white pine wood chips with a LHV equal to 16.66 MJ/kg_{db}. Three final moisture content are estudied, 10, 20 and 30 $\%_{wb}$ for two initial moisture contents, 50 and 60 $\%_{wb}$. In the case of using steam as the drying agent, the idea that there is enough waste energy to generate the amount of steam at the temperature studied, is assumed. The estimated running time for economic aspects is 8400 h/y. Recirculation is studied for steam as drying medium, and it has considerable benefits at high recycle ratio in terms of heat required from flue gas.

The results of this article show a higher investment cost when superheat steam is considered as the drying medium (around €3 $million$) than using flue gas (around €2.5 $million$) due to the need of a stainless steel construction. Overall, for both drying mediums, $3 - 4$ years of operation is expected to give a return on the investment at a fuel price of 14 €$/MWh$. However, the results are very sensitive to the biomass fuel-selling price. It is concluded that it a minimum cost of 8 €$/MWh$ is needed to achieve a return on the investment after 10 years. Present value decreases slowly as the final fuel moisture content increase until it reaches levels of around 25 $\%_{wb}$. A NPV of around €3.6$million$ after 10 years for values between 10 and 30 $\%_{wb}$ of final moisture content, is expected.

Myllymaa et al. [39] evaluates the effect of drying kinetics on drying costs. The considered drying agent is air and aspects such as fuel bed height, air temperature and air velocity and their optimal values are studied as well as the initial moisture content effect. Air velocities vary from 0.3 to 0.7 m/s and 0.9 m/s in one case. Bed heights studied are 300, 400 and 500mm and the temperatures of the drying agents are between 30 and 110 $°C$. A purely economic variable is also studied: the rate of interest. As a size parameter, a mass flow rate of 2.6352 t_{db}/h is considered.

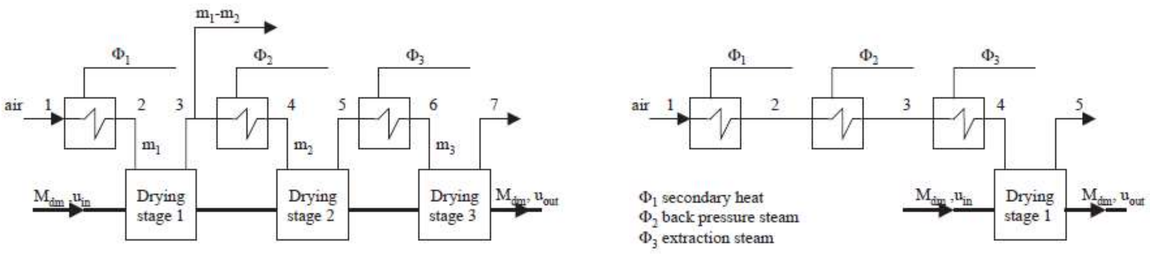
Several types of biomass have been considered: wood chips and bark, and a mix of soot sludge and moist sawdust. The soot sludge is extremely moist waste from chemical industry which needs to be destroyed and drying it is an opportunity to exploit it. Initial moisture content of wood chips and bark are between 50 and 56.5 $\%_{wb}$. In case of the mix of soot sludge and moist sawdust, 84 $\%_{wb}$ is defined as a result of mixing them in equal parts. Final moisture content of wood chips and bark are 17 and 37.5 $\%_{wb}$ and 41 $\%_{wb}$ for the mix of sludge and sawdust. The running time is 8000 h/y. The price of electricity and heat in the base case studied are setled at 40 €$/MWh$ and 5 €$/MWh$ respectively for every air temperatures reached. A sensitivity analysis of these values is done. Regarding heat price, it mainly depends on the means by which it is produced. Profitability is very sensitive to this value. Moreover the article presents a study about splitting the dryer in blocks with different air temperatures and velocities, however this is out of the scope of this analysis.

The results obtained conclude that, for wood chips and bark, the higher bed height and the higher air temperature yield, the lower the drying costs, regardless of the change of moisture content, when the price of the heat was kept constant for every air temperature. However, in the case of soot sludge and sawdust mixture, rising the bed height from 200 mm to 300 mm slightly increased drying costs. This indicates that there is an optimal bed height, which depends on the material and its moisture content. Regarding air velocity, its increase could not enhance the drying phenomenom and it can even rise the drying cost.

The senstivity analysis shows that the drying costs are not influenced by the rate of interest nor the electricity price, while the heat price affects the optimal design parameters. These results revealed that the lowest drying costs are achieved by the lowest air temperatures when the heat price changes depending on the air temperature. However, if the prices for lower and higher air temperatures are at the same magnitude, the higher air temperatures should be used.

Holmberg et al. in [30] and [40] compare the drying cost [30] and the energy efficiency (on the basis of energy and exergy analysis)[30] of two alternative drying systems: multi-stage drying, and single-stage drying with multi-stage heating. The existence of a HRU is considered in [40]. The articles study drying of biomass using air as drying medium in a pulp and paper mill; the air is heated up by energy from this industry: Secondary heat (or waste heat) in form of hot water at $50-90$ $°C$, back pressure steam at $3-4$ bar and extraction steam at $10-12$ bar. The order of magnitud and the size of the plant is defined by the mass flow rate of biomass to be dried: 1 kg_{db}/s (3.6 t_{db}/h).

Initial moisture content in both studies is 60 $\%_{wb}$ and final moisture contents 16.6, 28.5 and 37.5 $\%_{wb}$ are analysed in [30], while 23 $\%_{wb}$ is considered for the energy efficiency analysis [40]. The energy available from the industrial process was considered constant, 8 MW and in several cases the share of each energy available per each heat source (secondary heat, back pressure and steam extraction) was varied. the variation of amount of each energy source affect considerably due to the limitation in the heat transfer by the temperatures of each heat source. Regarding the air, a water vapor content of 0.064 kg/kg_{da} was set up considered at the entrance of the dryer. The recirculation factor is only considered in [40]. A scheme of the drying system proposed in [30] is seen in figure 2.3.



(a) Multi stage drying (b) Single stage drying and multi stage heating

Figure 2.3 – Scheamatic of biomass drying process proposed by Holmberg et al.[30]

The results in the drying cost analysis show that, if the amortisation time is short, single-stage drying is usually a more economic way to carry out the drying. If the amortsation time is long, more attention is paid to running costs and multi-stage drying is generally more profitable. In both systems, the heat consumption is similar regardless of the drying temperatures. Results show that the irreversibility rate depends to a considerable extent on the temperature of the heat source and also the drying system. A high temperature difference between the heat source and air also increase the irreversibility rate. The drying temperatures should be as low as possible, and the heat transfer should occurs over small temperature differences.

To summarize, the main figures from the studies analysed are seen in table 2.1.

	De Fusco et al.	**Li et al.**	**Myllymaa et al.**	**Holmberg et al. [30] and [40]**
Size of the plant feeded	$1-6\ t_{daf}/h$ $<\ 30\ MW$	40 MW	2.6352 t_{db}/h	3.6 t_{db}/s
Drying medium	Preheated air	Flue Gas or Superheat steam	Air	Air
Heat source	Heat source 130 $kJ/kg_{air_{db}}$	FG: directly from an industry process Steam: From hot water 90ºC and FG	Heat source 5 €/MWh (Base Case) for all air temperatures	Preheat by three heat sources: hot water (secondary heat), back pressure steam and extraction steam. Total 8 MW
Vapor content in the drying medium initial	RH=50%	-	$0.002-0.0051\ kgH2O/kg_{da}$	0.0065 $kgH2O/kg_{da}$
Vapor content in the drying medium final	RH=100%	-	-	-
Velocity drying medium (m/s)	0.65	-	0.3, 0.5, 0.7	0.6
Temperature of drying agent inlet (ºC)	150	FG: $250-450$ Steam: $140-180$	30-110	70, 120, 165
Temperature of drying agent outlet (ºC)	30-70	-	-	-
Recirculation	0.3	No	No	Yes
Moisture content inlet ($\%_{wb}$)	60	60,50	56,50 for wood chips and bark, 84 mix	60
Moisture content outlet ($\%_{wb}$)	10, 15, 20, 30, 40	10, 20, 30	16.6, 37.5 for woodchips	16.6, 28.5, 37.5
Running time (h)	8000 (base case), 6000, 5000, variable	8400	8000 (base case)	-

Table 2.1 – Summary of main parameters in literature review. [30, 31, 38–40]

2.2 Drying calcuations

In this section, some important aspects about the drying equations used in literature will be discussed. Belt dryer is considered in every case above mentioned, but different assumptions have been made to calculate the results.

Holmberg et al. [40] and De Fusco et al. present two main assumptions that define completly the energy and mass balance. Firstly, with regard to the drying medium, air is considered fully saturated at the outlet of the dryer. This means that the air is not able to absorb any more water and a decrease in its temperature leads to condensation. Saturation of the air depends on the temperature of the air at the outlet. This assumption is based on that the design of the bed height and the velocity of the drying medium are optimized to obtain this saturation condition.

The second assumption made by these authors is the linear correlation between temperature and moisture of the fuel based on drying experiments represented in [41], by Holmberg et al., and [42] where it is concluded that the constant drying rate period is relatively short in wood drying with moist air and the temperature of the material rise constantly as the moisture content decreases.

The size of the dryer in both studies, defined by the conveyor surface, has been calculated considering the mass flow rate of the drying medium and dividing it by its velocity through the bed and its density. Mass flow rate of the air will be determined by the mass flow of fuel and its initial and final moisture, among others.

Results of **Myllymaa et al.** for residence times in the dryer of biomass fuel are obtained experimentally in a batch-type fixed bed dryer by the measured drying curves/data published earlier by other authors [43–45]. The residence time depends on the initial and final moisture of biomass, air temperature, bed height, etc... In this work, no assumptions have been made about the calculation as in the studies analysed above. Instead, the main values and results have been obtained experimentally.

There is no explicit mention regarding the mass and energy balance in the dryer, except for the heat consumed in the dryer, which is the heat transferred to the drying agent. In this article, the estimation of the mass of the drying agent is calculated as a function of the velocity of the air through the bed, as a variable, the dryer surface, which is fixed by the relation defined in equation (2.1), and the density of the air, which in turn depends on the temperature, another variable.

Unlike the above-mentioned studies, the conveyor surface and therefore the size of the dryer is defined by equation (2.1), where $\dot{m}_{db}$ is the mass flow of material, τ is the residence time, ρ_{db}, bulk density of dry materia and Z the height of the bed.

$$A_{dryer} = \frac{\dot{m}_{db} \cdot \tau}{\rho_{db} \cdot Z} \tag{2.1}$$

Li et al., in the case of using the flue gas as a drying agent, point out that it is common to assume that within the interior of the dryer, the drying stream follows an adiabatic process, which means that the enthalpy at the entrance of the dryer is equal to that at the outlet of the dryer. The enthalpy of the flue gas is defined by the specific heat of the drying medium and its temperature, and the specific heat and latent heat of the water content. Notice that the specific heat of the drying medium is that of air, instead of that of the flue gas since in both mediums this value is very similar.

Taking into account all the above mentioned, the drying proccess is caused by a pure variation in the exergy since the vapour content in the flue gas will increase while the temperature decreases. The flue gas flow rate is calculated in function of the evaporation rate, which is defined by the initial and final moisture content and the mass fuel rate, and the initial and final humidity of the flue gas. No information about humidity values in the flue gas has been found. Same considerations are made in case of superheat steam, limited by saturated steam conditions at a given temperature.

The size of the dryer is defined by the mass fuel rate, its initial moisture content and the residence time of the fuel estimated for each final moisture content considered. The conveyor surface is obtained considering a unit area loading, W_{load}, of 30 kg/m^2 as seen in equation 2.2. M_{wood} is the fuel mass, MC, initial moisture and τ_{wood}, the residence time. As Myllymaa et al., the residence time, moisture contents and other principal values are obtained from experimental results, which correspond to the values for pinewood obtained from [46, 47].

$$A_{eff} = \frac{M_{wood} \cdot (1 + MC) \cdot \tau_{wood}}{W_{load}} \tag{2.2}$$

2.3 Economic evaluation comparison

As in the previous sections, the considerations of each work related to the economic aspects will be analysed. Depending on the purpose of each study or the economical comparison, different considerations about the capital cost and O&M costs are found.

De Fusco et al., as stated above, estimates the revennue as a result of the increase in the steam production as a function of the moisture removed. This means that the fuel mass in dry basis is kept constant and the increase in the steam production is due to the improved efficiency. The evaluation of the benefits from drying, has been analysed with the use of the Net Present Value (NPV). In this case, O&M cost are considered a fraction of the investment cost. In this case, it varies from 8 to 15 % in order to study its effect in the viability.

For NPV calculation, equation 2.3 is used. $R_{steam-incr.y}$ is the annual revenue, $C_{O\&M,dryer}$ is the dryer operation and management cost and $C_{INV,dryer}$ is the capital cost of the dryer. Capital cost is amortised over k years of operation at a nominal interest rate of $i\%$ per year.

$$NPV = \sum_{t=0}^{k} \frac{R_{steam-incr.y} - C_{O\&M,dryer}}{(1+i)^t} - C_{INV,dryer} \tag{2.3}$$

In this work, investment cost are estimated using two models and the average value between them is the one considered. Both models have as main parameter the conveyor surface of the dryer. The first model, the cost funtion obtained from Brammer et al. [36] whose data is obtained from manufacturers (equation 2.4). A_d is the dryer area. Theses figures are dated in 2002.

$$C_{INV,dryer,1}\,[€] = 7820 \cdot (2.79 \cdot A_d + 52.2)^{0.863} \tag{2.4}$$

The second model (equation 2.5) is obtained from equipment suppliers and from Brennan [48]. This model is also used by Holmberg et al. and Li et al. the result obtained is multiplied by a Lang factor, G of 1.6. This multiplication factor allows the inclusion of other costs such as electricity, instrumentation, etc.

$$C_{INV,dryer,2}\,[€] = G \cdot \sum C_{equipments} \tag{2.5}$$

The estimation cost for each equipment is of the form $k \cdot Y^n$, where k is a proportionality factor, Y is the capacity parameter and n is the size reduction factor. Table 2.2 contains these figures. These figures are from 2002.

Equipment	Relationship	Capacity parameter Y	Additional parameter
Coveyor	$2700Y$	Cross-sectional area	
Air-water heat exchanger	$9\Delta t Y^{0.9}$	Air mass flow	Δt is temperature increase in heat exchanger
Air-steam heat exchanger	$18\Delta t Y^{0.9}$	Air mass flow	Δt is temperature increase in heat exchanger
Air duct	$3770Y^{0.5}$	Air mass flow	
Fan	$0.9\Delta p Y^{0.7}$	Air mass flow	Δp is pressure drop of drying stage
Covering	$1200Y^{0.5}$	Cross- sectional area	

Table 2.2 – Equipment cost model used by Holmberg et al., De Fusco et al. and Li et al.

The revenue due to the extra steam production is calculated as shown in the equation 2.6. C_{steam} is the steam economic value ($c€/MJ$), ΔG_{steam} is the variation in the steam production due to the use of dried fuel (kg/s), h is the enthalpy content (kJ/kg) and $\tau_{operation}$ is the operational time of the plant ($hours$), which is considered 8000 h in this study.

$$R_{steam-incr,y} = C_{steam} \cdot \Delta G_{steam} \cdot \Delta h_{steam} \cdot 3600 \cdot \tau_{operation} \tag{2.6}$$

As above mentioned, **Holmberg et al., in [30]** evaluates the drying costs in two cases, multi-stage drying and single-stage drying with multi-stage heating. In each case, the drying cost (C_D) is calculated as shown by the equation 2.7.

$$Cost_D = Cost_{DC} + Cost_{IDC} + Cost_{RUN} \tag{2.7}$$

where $Cost_{DC}$ respresents direct capital cost and it is calculated in the same way as equation 2.5 and the same values with the same figures as in table 2.2. $Cost_{IDC}$ indirect capital cost, that is usually added as a percentage of direct capital cost and $Cost_{RUN}$, running cost.The latter is estimated as the sum of the operating cost due to the energy consumption in each heating stage, electricity cost consumed by fans in each dryer, and maintenance and other costs associated with the operation of the dryer. In equation 2.8, Φ is the heat consumption (W), P the electricity consumption (W), b_h the price of the heat, b_e, the price of electricity, $Cost_m$, maintenance cost and $Cost_x$ represent other costs asociated with the operation of the dryer. The price of the heat depends on the heat source.

$$Cost_{RUN} = \Phi\tau b_h + P\tau b_e + Cost_m + Cost_x \tag{2.8}$$

The comparison is done by the difference between drying costs, as shown in equation 2.9, where SSD means single-stage drying and MSD, multi-stage drying. If the difference is negative, single-stage drying is a more economic way to carry out the drying. If the difference is positive, then multi-stage drying is a more economic drying system.

$$Difference = Costs(SSD) - Costs(MSD) \tag{2.9}$$

The economic analysis done by **Myllymaa et al. [39]** consists on comparing the costs of a given case with the case with the highest costs. The drying costs of each case is defined by equation 2.10, where P is adjusted annual payment/cost (€/a), A_{inv} annual investment cost (€/a), A_{oper} annual operational costs (€/a), $I_{tot,dryer}$ total investment cost of the dryer (€) and AF is the annuity payment factor.

$$P = A_{inv} + A_{oper} = AF \cdot I_{tot,dryer} + A_{oper} \tag{2.10}$$

The total investment costs of the dryer is calculated using the correlation defined in equation 2.11. This equation is based on the investment cost obtained from current dryer manufacturer and it is accurate when $0\ m^2 \leq A_{dryer} \leq 480\ m^2$.

$$I_{tot,dryer} = A_{dryer} \cdot \left[-3905 \cdot \ln(\frac{A_{dryer}}{480}) + 5838 \right] \tag{2.11}$$

The annual operational cost (A_{oper}) is defined as the sum of electricity and heat costs. The former is defined by its price and the velocity of the air, the dryer surface, pressure drop and efficiency of fan. It is important to mention again that this work studies the variation of drying kinetics on drying cost. The variation in bed height, air velocity, air temperature and initial/final moisture content is studied. Thus, the pressure drop depends on the velocity and bed height, the dryer surface depends on the residence time and bed height and air velocity is a variable studied.

The heat cost is defined by the cost and heat consumption of the dryer, i.e. the heat transferred to air (drying medium) from the heat source. Regular calculation of the heat by multiplying the mass flow of air by the enthalpy drop is used. The mass flow of the drying medium is calculated by the multiplication of its velocity, the dryer surface and the density. The enthalpy of the air is calculated using the equation for the humid air, considering the latent heat value of the water andspecific heat capacities are considered constant.

The profitability studied by **Li et al.** is evaluated in terms of payback time and the Net Present Value is used. The calculation of the NPV by these authors is shown in equation 2.12, where C_t is the cash benefit in t years, $Cost_{main}$ is maintenance cost, $Cost_{capita}$ is the investment cost and i is the interest rate. This study, as well as De Fusco et al. and Holmberg et al., estimates the investment cost with the model proposed by [48] and these parameters can be seen in table 2.2. Maintenance cost is defined as a percentage of the capital cost. In this case, maintence cost is fixed at 5%.

$$NPV = \sum_{t=0}^{k} \frac{C_t - Cost_{main}}{(1+i)^t} - Cost_{capital} \tag{2.12}$$

The cash benefit, C_t is defined in equation 2.13, where C_{save} is the saved fuel per MWh, C_f is the price of energy stored in the flue gas (€$0.5/GWh$ here) and τ_{op} is the total number of operatin hours in year 't' (in this case, 8400 h). $Q \cdot C_f$ is defined as the total flue gas cost and Q is the heating rate for water evaporation (kJ/s).

$$C_t = (C_{save} - Q \cdot C_f) \cdot \tau_{op} \tag{2.13}$$

The saved fuel and thus the saved energy in the boiler (C_{save}) can be converted into a positive cash flow as seen in equation 2.14, where C_{fuel} is the price of fuel and $W_{evap} \cdot H_{latent}$ represents the total energy required to evaporate the desired amount of water in 1 h.

$$C_{save} = W_{evap} \cdot H_{latent} \cdot C_{fuel} \tag{2.14}$$

Chapter 3

Thermodynamic Model

This section describes the two thermodinamical models considered in the study: the biomass boiler and the dryer. The former was developed by Thibault Coppieters and used in studies such as [49], where a techno-economic design of Flue Gas Condenser for medium-scale biomass combustion plants is developed. The latter model has been developed in this study.

This project studies the feasibility of drying the biomass fuel that feeds a boiler using the flue gas from this boiler as the drying medium. The main ideas on which the models and their connection are based consist of, first, considering a biomass fuel and its moisture content, say, 50 $\%_{wb}$, enters the dryer. A certain amount of drying agent that is defined as a share of the total flue gas produced by the boiler, say, 70 % of the flue gas produced, dries the fuel to a certain moisture content. This less moist fuel enters the boiler, where it is burned. The mass flow of the fuel will depend on its humidity at the inlet of the boiler, the energy demand and the boiler efficiency. The amount of flue gas produced depends on several factors that will be described in sections below and that are mainly the moisture of fuel in the boiler and its load, defined by the demand and the nominal power rate.

As mentioned above, the moisture content of the fuel at the entrance of the boiler depends on the amount of flue gas produced and its share usage to dry the fuel, which in turn depends again on the former and on the load. This creates an interation loop where the initial moisture content, the share of flue gas (not the amount) and the demand and load are the main inputs. The main outputs are the fuel mass flow, its moisture content at the inlet of the boiler, the amounts of flue gas produced by the boiler and used to dry and the surface required to dry, among others.

Figure 3.1 represents a scheme of how the connection between both models works. The conveyor surface is obtained as a result of defining the flue gas velocity through the bed conveyor. Nominal rate defines the size of the boiler and the plant whereas the share of flue gas dedicated to drying defines the size of the dryer as a function of the size of the boiler.

The language used to program and obtain the results of the boiler and dryer is Python.

Efficiency appears directly proportional with how dry the fuel is in the boiler, but below a certain humidity level,$\approx 10-15\ \%_{ad}$, problems such as high flame temperature and CO production appear. However, economic conflicts may arise because the larger the dryer, the higher the investment cost. The fuel saved, or the energy overproduction considered in other studies may not be sufficient to compensate this cost increase and the economically optimal moisture content may not be equal to the optimal moisture content in terms of thermal efficiency.

Finally, the aforementioned condisers a constant and known the composition of the fuel and ambient conditions in terms of temperature and absolute humidity.

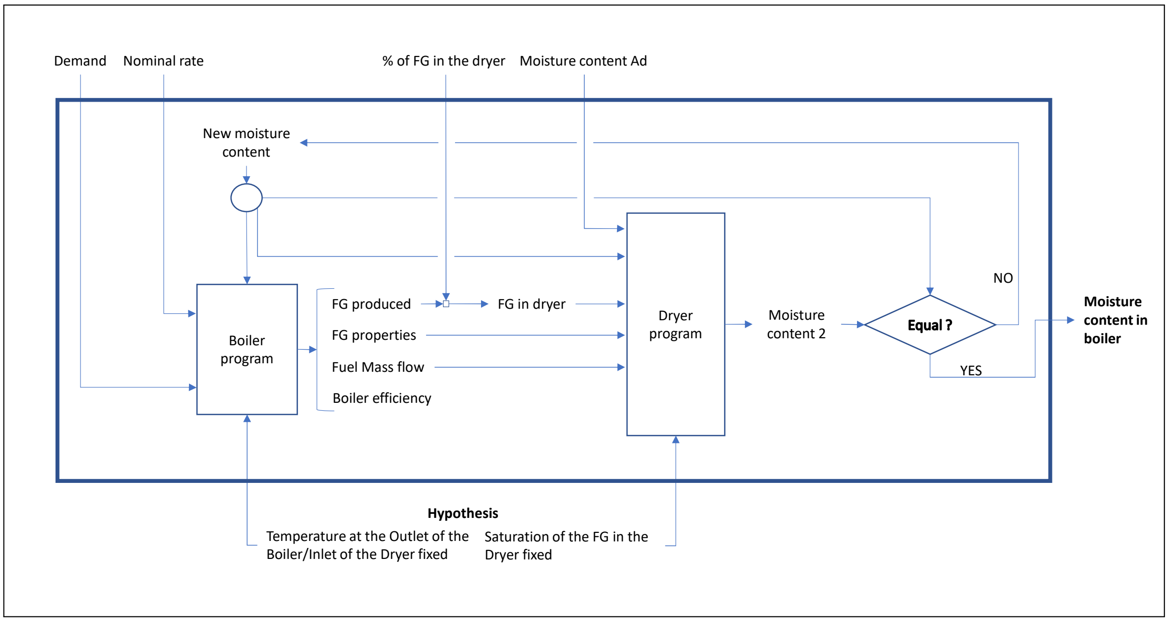


Figure 3.1 – Diagram of the loop programmed

3.1 Boiler

Modelling a biomass boiler and the combustion process requires complex equations and long computation times in order to obtain accurate results such as adiabatic temperature, combustion efficiency, losses in the furnance and level of incomplete combustion. The type of boiler (grate furnace, fluidised bed or pulverized fuel), the fuel and its composition or the ambient conditions affect the boiler modelling.

Apart from the combustion process and the transformation from chemical to heat energy, many issues must be considered in the modeling of the heat transfer in the exchanger. The following properties, which impose several restriction in the model, must be defined: the temperature of the heated fluid, its thermal properties such as the constant heat value or its possible phase change at a certain pressure level and the latent heat value required. For instance, if the heated fluid is water, depending on whether there is a phase change or not, it is possible to differentiate three exchangers: economizer (liquid), evaporator (liquid and vapour) or superheater (water vapour).

The heat transfer phenomenon differs in each of these three exchangers due to many factors. The velocity of the flue gas and its temperature in each exchanger, as well as its turbulence or the layout of the tubes in case of the heated fluid circulates in them, and other issues related with the heat source, vary the heat transfer coefficient. The exchanger material, thickness of tubes, the existance of radiation from the flame due to the location of each exchanger, friction losses, thermal losses, etc, are different aspect that must be considered in order to develop a model of the boiler and furnance.

Despite the accuracy that can provide, the resulting model tends to be rigid due to the particular aspects defined and boundary conditions. Each boiler technology presents different thermal kinetics and velocities of combustion and by-products (content of ash, char, volatiles, etc...) that also depends on the fuel properties. Modelling the heaf exchanger lead to fix upsetting parameters such as the type of exchanger, unit size... for certain optimal conditions. Usually these models also use empirical correlations that offer accuracy within the boundary conditions but are not valid outside them. Therefore depending on the modeling level, i.e. the quantity of assumptions considered, it would be less acurate but more flexible.

De Fusco, builds a simple model in his Master Thesis [20], where the estimation of the steam increment due to the decrease in fuel humidity is required in his analysis, that can adapt to different plants and operating conditions. The model has been built by defining two zones: internal fuel drying and thermal decomposition zone (gasification and oxidation). The products composition from gasification are computed by minimization of the Gibbs free energy of the system.

3.1.1 Description of the model

Techno-economic analysis and feasibility studies are often not accurate in general terms, but can provide conclusive results using relatively simple models. In this project, where the increase in efficiency due to the drying of the fuel is considered a reduction in fuel consumption rather than an increase in the energy produced, aspects such as the type of boiler and exchanger, heat transfer function or the temperatures of the heated fluid are not defined. Instead, the heat transferred to the fluid is considered as an input to the model and the temperature of flue gas varies linearly with the boiler load, regardless of the fuel moisture content and excess air are.

The combustion model developed is based on references [23, 50]. The composition of the FG and its mass flow rate per kilogram of dry fuel, considering the excess air fed to the combustion chamber, are calculated considering the simplified expresion for biomass fuel $C_mH_nO_xN_yS_z$. Ash is considered as inert for combustion reaction. The chemical reaction can be seen in equation (3.1).

$$\begin{aligned}
&C_mH_nO_xN_yS_z + \chi_w \cdot H_2O + (1+e) \cdot \kappa \cdot (O_2 + \beta \cdot N_2 + \gamma \cdot H_2O) + \\
&+ \chi \cdot \Big[(m-f) \cdot CO_2 + (\frac{n}{2} + \chi_w + \gamma) \cdot H_2O + (\beta + \kappa + \frac{y}{2}) \cdot N_2 + e \cdot \kappa \cdot O_2 + z \cdot SO_2\Big] \Rightarrow \\
&(1+\chi) \cdot \Big[(m-f) \cdot CO_2 + (\frac{n}{2} + \chi_w + \gamma) \cdot H_2O + (\beta + \kappa + \frac{y}{2}) \cdot N_2 + e \cdot \kappa \cdot O_2 + z \cdot SO_2\Big]
\end{aligned} \tag{3.1}$$

In this equation (3.1), e is the excess of air, χ_w is the mole fraction of water in biomass, β is the ratio between molar fraction of nitrogen and oxygen in wet air, γ is the molar fraction between water and oxygen in wet air and $\kappa = (m + \frac{n}{4} + \frac{x}{2} + z)$. f is the molar fraction of CO. χ is defined as the molar fraction of reciculated combustion gases. In this formulation, ashes are not recirculated and it is considered that they are captured before recirculation.

The boiler model is versatile and has many imputs to consider. It is possible to vary the fuel composition studied and allows to set different excesses air. In this project, neither the variation of the composition nor special air excesses will be studied. Instead, a fix type of wood (wood chips) and a excess air set as a function of load and fuel moisture will be considered. The model developed by Coppieters et al. do not consider the recircualtion of the flue gas and the production of CO is neglected because of the excess air is considerede enough to avoid its production. The equation (3.1) can be rewritten as follow (equation (3.2)):

$$\begin{aligned}&C_mH_nO_xN_yS_z + \chi_w \cdot H_2O + (1+e)\cdot\kappa\cdot(O_2 + \beta\cdot N_2 + \gamma\cdot H_2O) \Rightarrow \\ &m\cdot CO_2 + (\frac{n}{2} + \chi_w + \gamma)\cdot H_2O + (\beta + \kappa + \frac{y}{2})\cdot N_2 + e\cdot\kappa\cdot O_2 + z\cdot SO_2\end{aligned} \tag{3.2}$$

Once the fuel is selected, the excess air must be set in order to calculate the parameters from equation (3.2). In this model, the oxygen content in the flue gas is correlated to the boiler load and the fuel moisture content [49].

Medium-scale biomass combustion plants often control combustion efficiency by measuring the composition of the flue gases. Elements such as the concentration of oxygen O_2, carbon monoxide , carbon dioxide CO_2 or water vapour can be measured in order to control the efficiency of the boiler.

Thus, the concentration of oxygen is equivalent to excess air. The presence of carbon monoxide indicates the level of incomplete combustion, as does the concentration of carbon dioxide. In the latter case, the level of incomplete combustion would be measured by the difference between the measured and expected concentration of CO_2. On the other hand, the presence of water vapour would allow the measurement of fuel moisture by knowing the hydrogen composition of the fuel and the abosolute humidity of the air entering the boiler.

Futhermore, measuring the oxygen concentration in the FG is the most widely used method of controlling the combustion process. This is usually done by means of a lambda sensor. The signal produced by this sensor is interpreted, so that the velocity of the fans varies to maintain the oxygen concentration at the set value. The O_2 concentration set varies mainly as a function of the boiler load and the fuel humidity, among other possible parameters. These O_2 levels are usually defined by previously analysed experimental efficiency results.

It is known from industry that a certain biomass plant in a certain region receives fuel from surroundings with different moisture levels because it could vary throughout the year. Based on this expected moisture, the oxygen curve as a function of the load is selected to optimize combustion at each moisture level. In many cases two humidity levels are considered as a function of weather: high moist fuel, expected in winter, and drier fuel, which is received in summer due to drying out by ambient conditions.

The management of the boiler in order to optimize the combustion process and limit pollutant emissions involves adjusting different parameters and operating modes by measuring different variables such as heat demand, flue gas temperature or concentration. There are variables whose measurement cannot be continuos or depens on other variables and certain empircal or physical correlations. An example is fuel moisture. This can be measured either through samples taken periodically or through continuous measurement of the water vapour content in the flue gases.

Thus, it is common to find numerous parameters to be adjusted by the operators. Some of these adjustments are the amount of air entering as well as its distribution among the different levels of the air staging system, the introduction of fuel, the recirculation of exhaust gases, among others.

As said, these adjustments seek the optimal operation of the boiler, that is, a tradeoff between optimizing efficiency and pollutant emissions. In addition, the main energy losses reside in the heat of the exhaust gases, heat losses through the walls and losses from unburners. In this project, it has been assumed that all adjustments are aimed to minimizing the amount of excess air (i.e. energy in the exhaust gases) needed to obtain complete combustion (i.e. avoid unburned).

This excess air is controlled by measuring the oxygen present in the flue gases. This excess of oxygen should usually increase linearly at low loads and decrease as the boiler load increases. In addition, fuels with higher humidity require more excess oxygen. In a real situation, there is a minimum load below which the boiler cannot operate due to significant efficiency losses, combustion issues, etc as it is considered in the model.

The use of different set points and curves of oxygen content and excess air in order to control the efficiency of the boiler is commonly used in the industry. In figure 3.2 it is shown the excess oxygen curves defined in this project from the aforementioned. The values shown represent an example of boiler operation. As previously mentioned, it is common to find two oxygen levels for two fuel moisture levels depending on the season. Limits and set points in this project are an example of them. These values are retrieved from [49] and are as follow: the minimum load at which the boiler can operate is set at 30 % and the excess oxygen set decreases linearly until nominal conditions are reached. The excess air is defined as the total volume of air in the boiler compared with the stoichiometric volume.

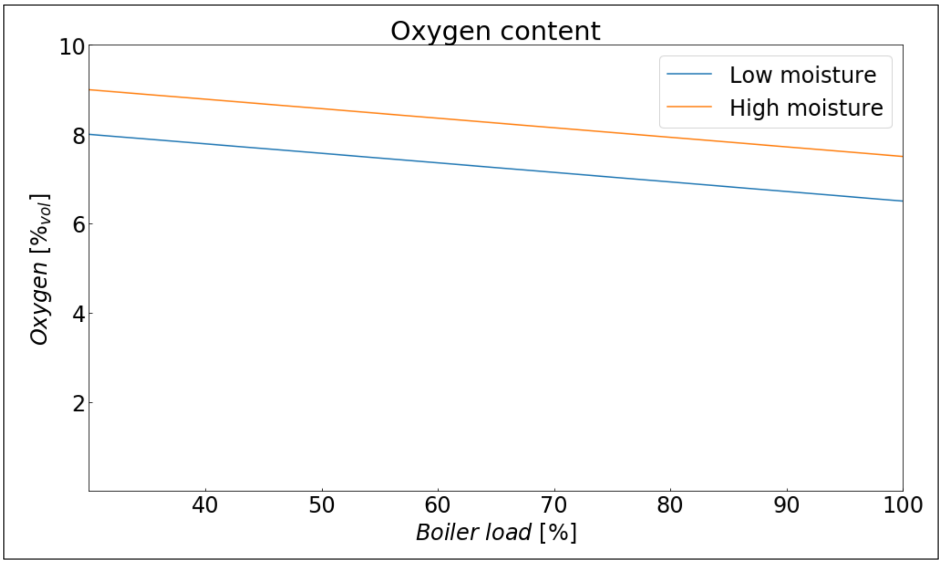


Figure 3.2 – Oxygen content in the FG as a function of the boiler load at the two fuel moisture levels

The maximum oxygen content at low load in the FG is 8 and 9 % for low and high moisture respectively, and 6.5 and 7.5 % at high load. This estimation of the excess air and oxygen calculation may have a significant impact on the results.

A wide difference in fuel humidity between summer and winter can be expected. In this project, the different humidity levels in the boiler are studied according to the amount of gas used in the dryer. For this reason, the humidity of the fuel in the boiler will not depend so strongly on the season but on the conditions of the dryer.

These authors [49] consider the limit between high and low humidity set at 35 $\%_{ad}$. However, this can induce large differences in results that can lead to incorrect conclusions when the final calculated moisture is close to this value. This is due to the drastic change in excess air. Instead, it has been estimated that fuel is considered to be very wet if it has a moisture content of more than 40 $\%_{ad}$. Also, fuel will be considered less humid below 30 $\%_{ad}$. The oxygen concentration considered in the intermediate values will be obtained as an interpolation of the two curves mentioned above.

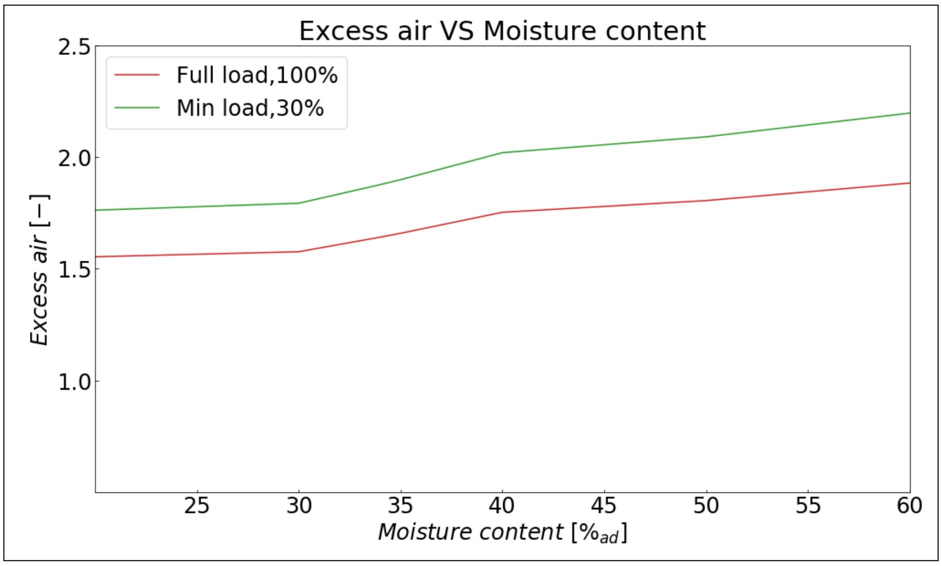


Figure 3.3 – Excess of air in the boiler as a function of the fuel moisture content at full and minimum load

In this model, heat demand is introduced as an input. The fluid to be heated has not been defined, thus its temperature is not considered at the inlet and outlet of the boiler, neither its mass flow rate nor its thermal properties. According to [51], the determination of the efficiency using the indirect method can be seen in equation (3.3). The *Heat losses* consist of:

- thermal losses by sensible heat of dry heat of dry flue gas and water vapor
- chemical losses by incomplete combustion
- thermal losses by radiation, convection an thermal conduction
- thermal losses in unburnt fuel
- thermal gain from latent heat of condensed water vapour in flue gas

$$\eta_{indirect} = 1 - \frac{Heat\ losses}{Heat\ in\ fuel} + \frac{Heat\ gain}{Heat\ in\ fuel}\ [-] \tag{3.3}$$

In this model, thermal losses has been considered as a percentage of the maximum heat capacity. Thermal losses in the flue gas is modeled by its mass flow, its constant heat value and temperature. Similarly chemical losses is calculated as a function of the concentration of CO in the flue gas. No chemical losses has been considered in this project. With theses considerations, the FG temperature can be deduced from the efficiency of the boiler or vice versa. As a result, this model allows to set either the efficiencty of the boiler or the flue gas temperature. The efficiency will vary as a function of the moisture content and the load. The latter asumption has been used in this project. Notice that the FG temperature varies as a function of the load.

Fuel High Heat Value has been calculated using the correlation (equation (3.4)) introduced by Sheng et al.[52] that is based on ultimate analysis. Where C and H are the weight percent on dry biomass basis and O is the sum of the contents of oxygen and other elements in the organic matter, i.e. $O = 100 - C - H - Ash$.

$$HHV = -1.3675 + 0.3137 \cdot C + 0.525 \cdot H + 0.064 \cdot O \ \ [MJ/kg] \tag{3.4}$$

Low Heat Value (LHV) in the different basis (db,daf,wb) have been calculated as described in section 1.2.1, equations (1.3) and (1.4). Considering now the direct determination of efficiency method defined in [51] and shown in equation (3.5):

$$\eta_{direct} = \frac{Q_{Out}}{Q_{In}} \tag{3.5}$$

The heat output, Q_{out}, is an input of this model. The heat input, Q_{In} is the sum of the chemical energy of the fuel in terms of heat value and the heat embebed in the air introduced for the combustion. This heat avalible considered is that after evaporatation of the fuel moisture. With this, equation (3.5) results in equation (3.6), where $\dot{m}_{air,ad}$ is the mass flow of air in the boiler per kg of fuel in wet basis/as delivered basis.

$$\eta_{direct} = \frac{Q_{demand}}{\dot{m}_{fuel,ad} \cdot (LHV_{ad} + \dot{m}_{air,ad} \cdot C_{p,air} \cdot T_{air})} \tag{3.6}$$

In the model, a loop with both determination of efficiency methods through equations (3.3) and (3.6), leads to the calculation of efficiency from a fixed FG temperature or vice versa. The most imporant variables extracted from the model in this project are the efficiency of the boiler, the fuel mass flow, the amount of FG and its properties (temperature, specific heat capacity, enthalpy, water vapor content, etc).

3.1.2 Justification of assumptions

In Mermoud et al. [53] and Haroutunian et al. [54] monitoring and a measurement campaign of two boilers (2 MW and 0.65 MW) supplying a distric heating system and the analysis of the impact of load variation on efficiency and emissions are studied. According to them, the excess air is between 1.5 and 2 at above 50 % load is a range that is in the usual recommended values. This excess air can exceed 2.5 at low load. As can be seen in figure 3.4, the oxygen concentration values assumed in this model lead to an excess air that complies with [53].

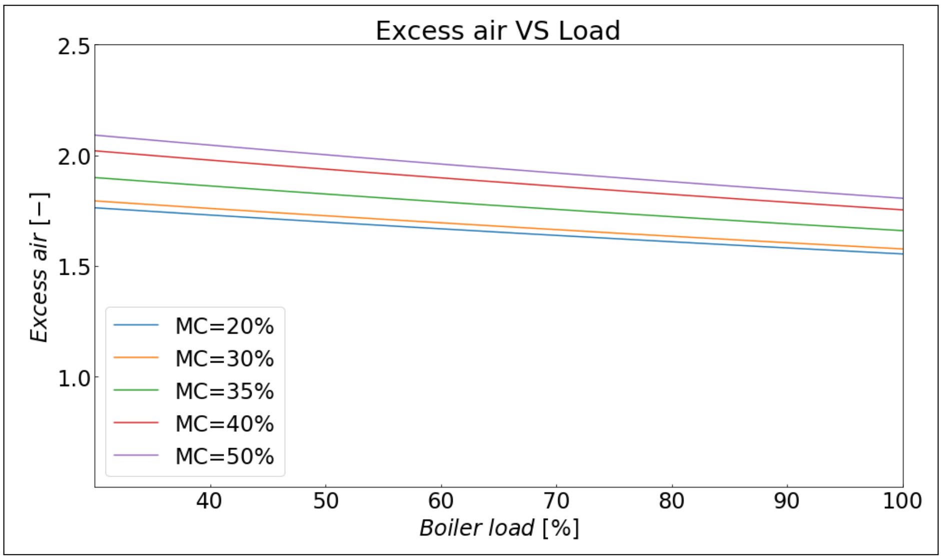


Figure 3.4 – Excess of air in the boiler at different moisture contents

Good et al. [51] set wall losses at 2 % of the full load heat. However, experimental measures in [53] on the walls of the boilers show temperatures close to ambient thanks to their excellent insulation. Consequenly, these thermal losses could be neglected in this case. In this model, both possibilities can be considered.

The typical efficiency values reported in the open literature and announced by manufacturers range between 80 to 90 % [49]. In the experimental study carried out by [53] and [54] show that the efficiency was barely influenced by the load. The efficiency calculation by the model is shown in figure 3.5.

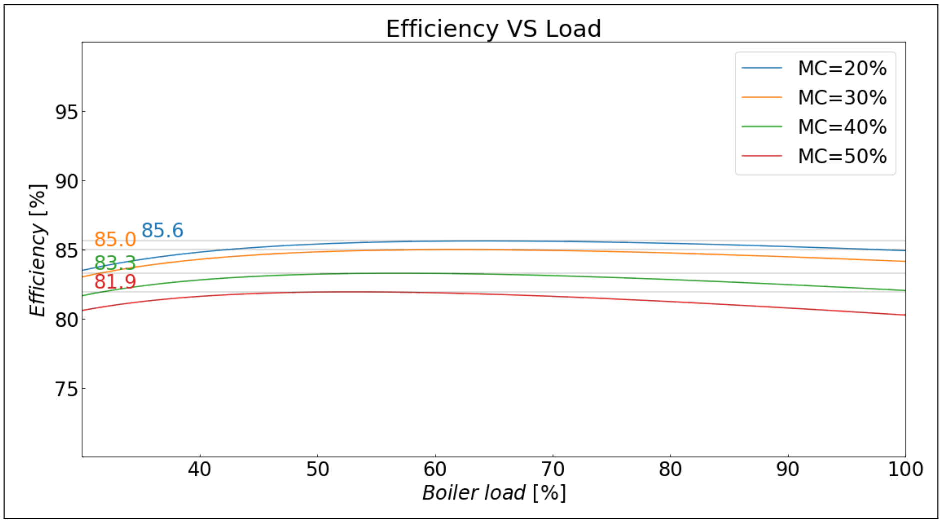


Figure 3.5 – Efficiency of the boiler at different fuel moisture content as a function of the load. 2 $\%_{P_N}$ wall losses.

Flue gas is defined as waste energy or secondary energy and it is considered as energy losses. These losses can be calculated by the product of the constant heat value, temperature and mass flow of flue gas. Fixing the rest of values, varing the temperature of FG, the efficiency is affected considerably.

In reference [53] and [54], it is shown that this temperature increase linearly when load increase because the exchanger efficiency is affected by velocities of the FG or adiabatic flame temperatures, among other factors. It is also mentioned that the highest temperature can be reduced by using an economizer. In this model, the exchanger is not modeled. It is not possible to know how load variation would affect the temperature of the FG, so experimental results would be needed. In this project, it is consider that the management of the boiler and exchanger will lead the flue gas to keep temperature at an acceptable values. In reference [51], temperatures around $170 - 190\ °C$ are measured at full load. The FG temperatures has been set at $100\ °C$ at 30 % of load and increases linearly up to $180\ °C$ at full load conditions. The possible effect of fuel moisture on this FG temperature has not been considered.

At low load, the temperature of the flue gas is lower, hence the losses embedded in the flue gas, which are lower in terms of enthalpy. On the other hand, the excess air setpoint at these low loads is greater, so the amount of gas per unit of energy in the form of fuel introduced will be higher. In addition, heat losses through the walls are set constant and equal to a percentage of the nominal heat rate, so the share of losses due to this issue will be higher at lower loads.

As the load increases, the share of losses through the walls in the energy balances decrease. Since the excess oxygen decrease, the flue gas mass flow per unit of energy introduced in the boiler will decrease. At a certain load for each fuel moisture content, the advantage of the decrease in excess oxygen setpoint and the share of the wall losses is diminished by the increase in the enthalpy of the flue gases. This can be seen in the figure 3.5.

This study covers the feasibility of using energy from te Flue Gas (FG) for drying biomass is studied. Therefore, it is important to know the energy available in this gas. By not considering the effect of fuel moisture on heat exchange efficiency, the enthalpy of the dry flue gas, i.e. without considering the water content, is constant whatever the moisture is and increases with the load. On the other hand, the enthalpy of total flue gas (dry flue gas + water vapour) will increase as the fuel moisture in the boiler increases. This variation is shown in figure 3.6.

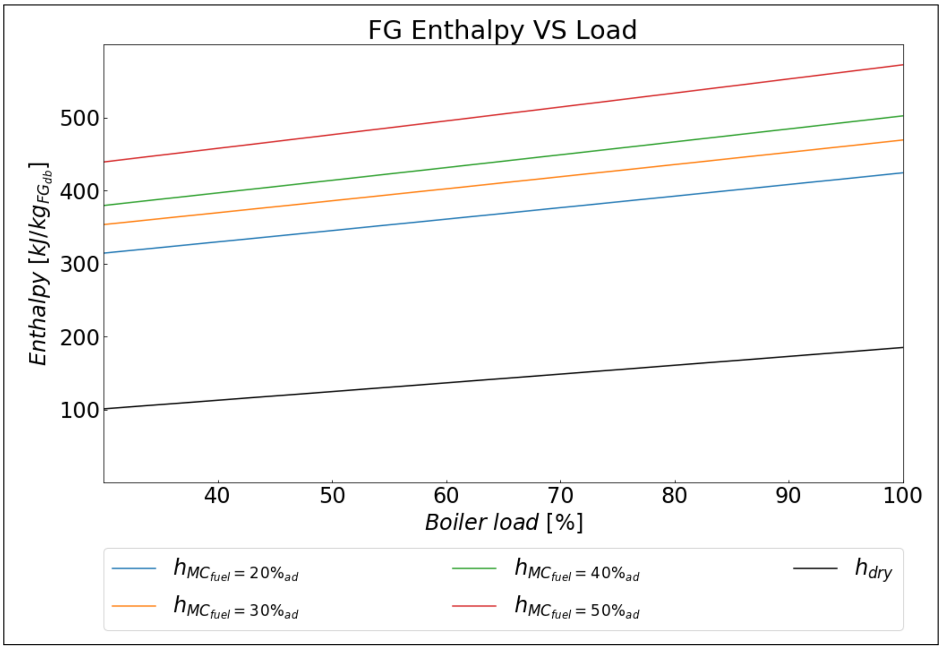


Figure 3.6 – Enthalpy of the flue gas at different fuel moisture levels. 2 $\%_{P_N}$ wall losses. FG temperature: linear increment from 100 °C at $P = 30\% \cdot P_N$ to 180 °C at $P = P_N$

3.2 Dryer

In this section, the thermodynamical model used in this study of biomass drying is described. The literature, along with the studies reviewed in chapter 2, show that there are different hypotheses and assumptions for estimating values of final fuel moisture content, the amount of drying agent needed or level of saturation vapor in this drying medium, depending on the variable to be studied. Moreover, these studies consider different size plants, with different heat source for drying purpose.

Some studies like those from Myllymaa et al. [39] and Li et al. [31], obtain these values from experimental results by drying a sample and measuring the moisture content, the drying time, drying medium velocity, etc. From this data and depending on the measured variables, such as bed height (in case of bed conveyor dryers) and water vapor content of the drying agent, the drying process can be modelled by interpolation or extrapolation of these results and other possible assumptions.

Other works like those from Holmberg et al. [30] [40] and De Fusco et al. [38] base their results on widely accepted hypotheses or ideals assumptions, some of which can be extrapolated from experimental studies. An example of these assumptions is the relative humidity of the drying agent at the outlet of the dryer.

3.2.1 Selection of the dryer

In the literature review done in this project, chapter 2, all the studies described choose as the type of dryer the one based on a bed conveyor band or belt dryer whathever the size in terms of mass flow of fuel or heat produced. In this project, the belt dryer is also the one chosen to model the drying of the biomass. Some hypohesis, assumptions and figures in the model assumed because of the definition of the type of dryer, such as the size of the dryer and the temperature of the fuel, will be described in following sections.

Belt conveyor dryers present several advantages over the rest of the dryers. Aspects such as operating temperatures, particles size or fire hazard, among other, are analysed in studies such as [20], [31] [29]. Table 3.1 offers a selection criteria done by Li et al. can be seen.

	Rotary	Flash	Belt	Fluidised-bed
Requires small particles	None	Yes	None	None
Heat recovery from dryer	Difficult	Difficult	Easy	Esasy
Fire hazard	High	Medium	Low	Medium
Air emissions	Medium	High	Low	Medium
Steam use	Yes	None	Yes	Yes

Table 3.1 – Considerations in dryer selection. Source: Li et al. [31]

This type of dryers operates at a relatively low temperature compared to others. In addition, the temperature range is wide and can vary from 30 $°C$ to around 200 $°C$. These temperature figures can be reached by the use of waste energy, such as back pressure steam, saturated water from a condenser or Flue Gas. This source can heat air but in some cases it is more interesting to use this waste stream as drying agent itself, what is possible in this dryers.

Fire hazard is lower than other considering the same drying agent and if this is flue gas instead of air preheated, the fire risk is even lower. Using steam as the drying medium reduce this fire hazard to a minimum.

The versatility of this dryer is also due to their ability to handle different sizes of biomass particles. The simplicity of the mechanisms used for drying offers a great robustness in the operation compared to others, with the exception of rotary dryer, which has the lowest maintenance costs for the same plant size, and the capacity can be as large as necessary. However, this means large conveyor surface and at large plant size or high dried fuel demand, other types of dryer can be more convenient in terms of efficiency, dimensions and economic benefits.

3.2.2 Description of the model

The model of the dryer is based on the energy and mass balance. The fuel mass flow and its initial moisture content is considered as known. The drying agent studied in this project is the FG from the boiler. Different composition of FG at different temperatues can be defined since the thermal properties of the FG are calculated in the model as a function of the composition, pressure and temperature of the FG. The drying agent mass flow is defined as a percentage of the amount of FG from the boiler.

Studying only the insulate drying model, The model would be as shown figure 3.7 where the final fuel moisture content would be calculated by the iteration between energy and mass balance. However, in this project, both dryer and boiler models are conected and the outputs of one affect the inputs of the other. The amount of FG in the dryer is conditioned by the humidity of the fuel in the boiler, which is the moisture content al the outlet of the dryer, as can be seen in figure 3.1.

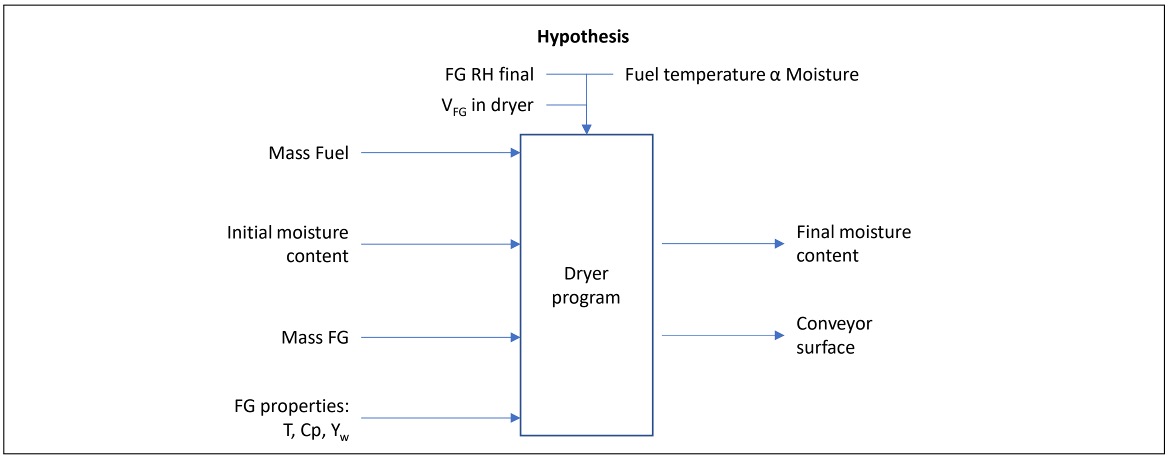


Figure 3.7 – Diagram of the dryer program

The two main equation of the model, as state above, are the energy and mass balance, equations (3.7) and (3.8), where $\dot{m}_{FG,db}$ is the FG mass in dry basis, i.e. invariant because this value is not affected by the water gain due to evaporation (FG without water vapor content), $\dot{m}_{Fuel,daf}$ is the fuel mass flow in dry ash free basis, which si also invariant. Enthalpies, h, and moisture contents, MC units are in dry basis or dry ash free basis. Heat losses, $Losses$ due to aspects such as wall losses can be considered as a percentage of the total FG heat exchanged.

$$(1 - Losses) \cdot \dot{m}_{FG,db} \cdot (h_{FG,in} - h_{FG,out}) = \dot{m}_{Fuel,daf} \cdot (h_{Fuel,out} - h_{Fuel,in}) \tag{3.7}$$

$$\dot{m}_{FG,db} \cdot (MC_{FG,out} - MC_{FG,in}) = \dot{m}_{Fuel,daf} \cdot (MC_{Fuel,in,daf} - MC_{Fuel,out,daf}) \tag{3.8}$$

As mentioned in section 2.2, articles such as Li et al. [31] consider the enthalpy of the drying agent as constant and temperature loss can be compensated by an increase in humidity according to the enthalpy calculation shown below in equation (3.11).

FG enthalpy is defined as can be seen in equations (3.9),(3.10),(3.11). The constant heat value of FG, $C_{p,FG}$ is calculated at the temperature of the FG and as a function of its composition, i.e. proportion of CO_2, O_2 and N_2. The thermal values for water vapour is evaluated in equation (3.10). Vapour constant heat value, $C_{p,vapor}$, is calculated at the FG temperature and vapor pressure. The latter is calculated using the water volume fraction in the FG and the saturation pressure. Latent heat value,L_v, i.e. the heat requiered for the phase change from liquid to vapour, is the one calculated at the vapour pressure. The suffix i refers to the inlet or outlet of the dryer.

$$h_{FG,dry,i} = C_{p,FG,i} \cdot (T_{FG,i} - T_{ref}) \ \ [kJ/kg_{db}] \tag{3.9}$$

$$h_{vapor,i} = C_{p,vapour,i} \cdot (T_{FG,i} - T_{ref}) + L_{v,i} \ \ [kJ/kg_{H_2O}] \tag{3.10}$$

$$h_{FG,i} = h_{FG,dry,i} + X_{FG,db,i} \cdot h_{vapor,i} \ \ [kJ/kg_{db}] \tag{3.11}$$

X is the water vapour content, i.e. the percetage in weight of water vapour in the flue gas. Similarly, MC is the fuel moisture content. The sufix refers to the weight reference (In these equations shown, mass of water/vapour per kg of dry ash free fuel/dry drying agent) The enthalpy of the fuel is calculated using similar equations, considering that all the water contained in the fuel is in liquid saturation phase:

$$h_{Fuel,dry,i} = C_{p,Fuel,i} \cdot (T_{Fuel,i} - T_{ref}) \ \ [kJ/kg_{daf}] \tag{3.12}$$

$$h_{water,i} = C_{p,liquid,i} \cdot (T_{Fuel,i} - T_{ref}) \ \ [kJ/kg_{H_2O}] \tag{3.13}$$

$$h_{Fuel,i} = h_{Fuel,dry,i} + MC_{Fuel,daf,i} \cdot h_{water,i} \ \ [kJ/kg_{daf}] \tag{3.14}$$

The specific heat capacity value of fuel is obtained from Gupta et al. [55], where it is described the measurement of specific heat and particle thermal conductivity for three woody biomass types: softwood, softwood bark and softwood char. Since the fuel in this project is wood chips, an average C_p between sofwood (equation(3.15)) and softwood bark (equation (3.16)) will be considered. Notice that the temperature, T, must be in Kelvins (K).

$$C_{p,Fuel,1} = (5.46 \cdot T - 524.77) \cdot 10^{-3} \ \ [kJ/kg_{Fuel}K] \tag{3.15}$$

$$C_{p,Fuel,2} = (3.69 \cdot T + 231.06) \cdot 10^{-3} \ \ [kJ/kg_{Fuel}K] \tag{3.16}$$

In line with this argument, the drying model has been developed with the similar assumptions made by De Fusco et al. [38] and Holmberg et al. [30, 40] related with the temperature of the fuel and the outlet FG conditions. Theses assumptions are necessary in order to calculate the outlet conditions in the dryer.

With regard to the fuel temperature at the outlet of the dryer, it can be calculated by a linear correlation with the final moisture content, as can be seen in equation (3.17) proposed in [30].

$$T_{Fuel,out} = \frac{T_{Fuel,in} - T_{FG,in}}{MC_{Fuel,daf,in}} \cdot MC_{Fuel,daf,out} + T_{FG,in} \tag{3.17}$$

The other assumption needed is the FG outlet conditions. It is foreseeable that, in order to optimize the performance of a dryer, the drying agent will come out in conditions in which it is not possible to dry anymore with that drying stream, i.e. the stream can not absorb more water from the fuel. This limitation is set by the relative humidity, in case of FG or air as drying medium, or saturation temperature in the case of steam. Concerning the case of air or FG, according to the psychrometric properties, at a certain temperature of the air/FG, there is a maximum water vapour content that the stream is capable to contain, and a decrease on this temperature will lead to water condensation. As known, at this point, it is said that the stream is fully saturated, i.e. relative humidity (RH) is at 100 %.

Since the drying stream loses temperature in favour of the fuel and gains water vapour, the relative humidity will tend to increase as much as possible. The design of the dryers seeks to avoid condensation inside them mainly due to corrosion issues. Holmberg et al. [40] and De Fusco et al.[38] consider that the drying medium would be fully saturated at the outlet of the dryer. In this project, it is considered that the relative humidity at the outlet of the dryer will be 90 %. This is a reasonable design value since it proposes a safety margin to avoid condensation inside the dryer.

Focusing now on the surface of the conveyor, which is the parameter that determines the size of the dryer, it is obtained as a result of dividing the air/FG mass flow by its velocity through the bed and by its density. The velocity is fixed and it is considered the same value as in [38], 0.65 m/s as well as the stream density, which is calculated at the temperature of the entrance of the dryer. In this project, this temperature is fixed by the boiler model at 180 $°C$ for design conditions (full load conditions). No temperature drop in the flue gas is considered between boiler and dryer.

Since it is not possible to dry totally the fuel, a minimum moisture content in the boiler is set at 10 $\%_{wb}$. Other values estimated less important are those such as the fuel temperature, 15 $°C$, and the pressure, 101325 Pa. It is also considered a pressure drop in the dryer equal to 1000 Pa as an estimation for the fans energy consumption.

3.2.3 Justification of assumptions

Acording to Pang et al. [56], which develop a mathematical model and an optimization for drying woody biomass using packed moving bed dryers, estimate the critical moisture, X_{cr}, for wood chips, around 55 $\%_{wb}$. As discussed in section 1.4.1, below the critical point, the drying rate is not constant anymore and it starts to decrease.

This links to drying experiments represented in [41] and [42], which initial moisture contents experimented are around $50-63$ $\%_{wb}$ in the case of [41]. In these studies, the constant drying rate period is relatively short in wood drying with moist air. This means that the temperature of the material rise almost constantly as the moisture content decrease. For these reasons, a simple linear correlation between moisture contents and temperature, as the one made by [40], will be considered here (equation (3.17)). Despite this equation is defined for air as drying medium, the thermal properties of air and FG are quite similar, and this allows to consider this equation in this project.

With respect to the relative humidity estimated at the outlet of the dryer, Holmberg et al.[30] conduct an experimentally study about the behaviour of drying time compared to the bed height, the ratio $\tau_u(Z)/Z$. It is concluded that at a certain drying medium temperature, there is a minimum bed height above which the drying medium is fully saturated during the first few minutes of drying.

This article [30] also mention the effect of the air velocity through the bed. The increase in the velocity affects the drying time and the bed height. Increasing the air velocity leads to a decrease in the drying time and higher mass transfer coefficient so the bed height ca be raised until the drying medium reachs saturation.

For the purpose of this project, the conclusion is that there is an optimal configuration of bed height and air velocity that allows the drying medium to reach the saturation level in real conditions.

The moist fuel can not be totally dried due to the the hydroscopic property of the biomass, which tends to absorb water until reaching the equibrium level that depends on the enviromental conditions. This means that the minimum moisture content will be limitated by the FG conditions at the outlet or the ambient conditions if it is not directly burned after drying. According to [33], the moisture content of a solid cannot be less than the equilibrium moisture content on the isotherm corresponding to the relative humidity of the air entering the dryer. Moreover, there is a moisture level below which, counterproductive effects start to occur, such as the production of CO during combustion. For theses reasons, and in order to be realistic, a minimum moisture content of 10 % is fixed and there is not possibility to dry below this value.

3.2.4 Validation of the model

In order to confirm the results extracted from the model, a comparison with the model proposed by De Fusco in his Master Thesis [20] has been done. The main assumptions made in the developed model comes from those by [20], so very similar results are espected.

The conveyor surface obtained by both models can be seen in figure 3.8. This results are obtained at a initial FG temperature of 180 $^\circ C$ for two initial moisture content, 60 and 50 $\%_{wb}$, the final relative humidity in both case is 100 %, i.e. the FG if fully saturated.

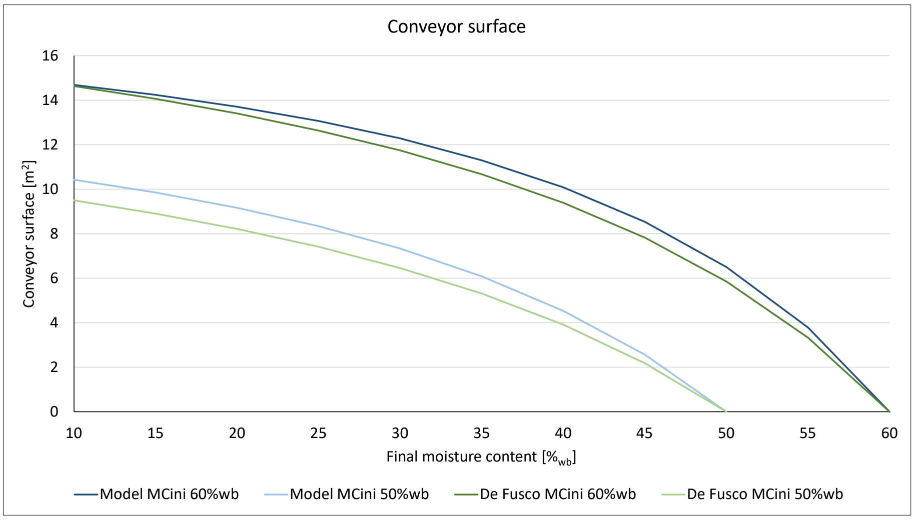


Figure 3.8 – Comparison of surface results obtained by De Fusco [20] and this model. $Size = 1\ tn_{daf}/h$, $T_{FG,in} = 180\ ^\circ C$, $V_{FG} = 0.65\ m/s$

There are two main differences between both models. Firstly, the constant heat capacity of dry fuel considered by de Fusco is retrieved from [57] and can be seen in equation (3.18) where T is temperature in Kelvins. Compared to the estimation considered in the model developed here, the values obtained with this equation are on a lower scale: 0.283 kJ/kgK versus 1.171 kJ/kgK at a fuel temperature of 15 $^\circ C$. This issue does not have a great impact in the results concerning to the conveyor surface but in the heat exchanged and the enthalpy variation in the FG and fuel. The impact of this difference is shown in figure 3.9. Since the enthalpy of liquid water is much higher than that of fuel, around 4.18 kJ/kgK, the effect of using one equation or the other does not seems to have an important impact.

$$C_{p,wood} = 266 + 1.16 \cdot (T - 273.15)\ \ [J/kg_{Fuel}K] \tag{3.18}$$

Secondly, both models differ on the initial water vapour content in the FG. De Fusco defines an initial relative humidity of 50 %, which mean, at high temperature, almost equal to zero mass of water vapour per mass of dry FG. In this model, since it is considered that

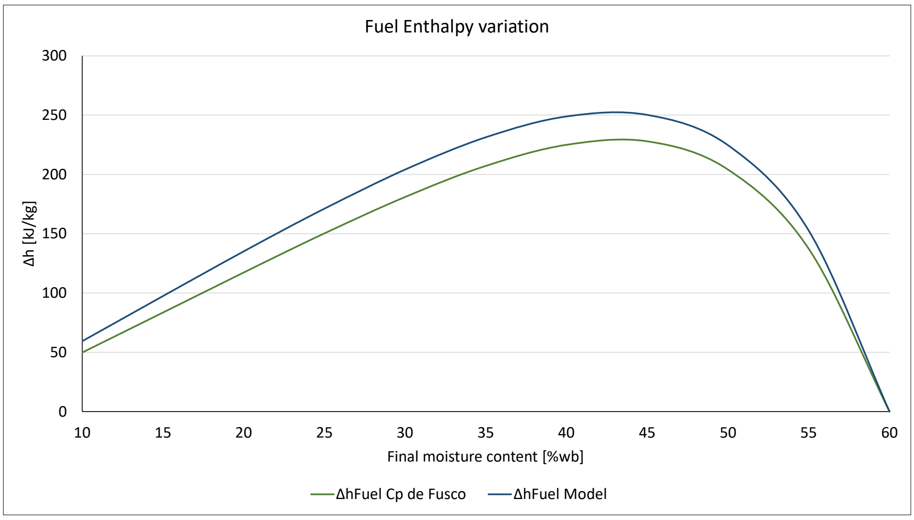


Figure 3.9 – Comparison of fuel enthalpy drop in results obtained by De Fusco [20] and this model. $Size = 1\ tn_{daf}/h$, $T_{FG,in} = 180\ °C$, $V_{FG} = 0.65\ m/s$

the FG used as drying medium is the one that comes from the boiler where the moist fuel is burned, the water vapour content in the FG will depend on the final moisture content of the fuel after being drying. This leads to have a higher FG enthalpy at the inlet of the dryer when the fuel is more moist (less dried).

The higher enthalpy due to the water vapour content compared to a dry FG leads to get higher temperatures of this drying medium after the dryer and, at this higher temperatures the FG is capable of absorb more vapour without condensing. However, the presence of water vapour in the FG compared to dry FG means that the capacity to evaporate and absorb water is impaired by the amount of water already contained before entering the dryer. In summary, if the FG has a higher "latent" enthalpy at the inlet of the dryer, the enthalpy at the oulet will be more valueable because it will be at higher temperature and it would be improbable the condensation in the dryer. On the other hand, the conveyor surface required will be more sensitive to the final RH set. Notice that this effect if more important at lower initial FG temperatures (figure 3.10).

In conclusion, the model is validated according to the one proposed by De Fusco. The perceived differences in terms of thermal properties of the FG and fuel does not have a great impact in the final conveyor surface estimated, but in the quality in terms of exergy.

3.3 Limitations

Results obtained by these models can provide a first estimatation of the energy and chemical processes that occur during the drying and subsequent combustion of the biomass.

In the boiler model, As state above, the combustion is not modeled and the FG composition is obtained by considering simplifications such as neglecting the CO formation considering a complete combustion. This may no be valid at low load.

Also related with the results at partial load, the correlation of the FG temperature as a function of the load is a linear correlation that does not consider the effect of the moisture content of the fuel. Moreover, this temperature figures set are an estimation that may not be totally correct since optimized boilers may have an economizer which optimizes the

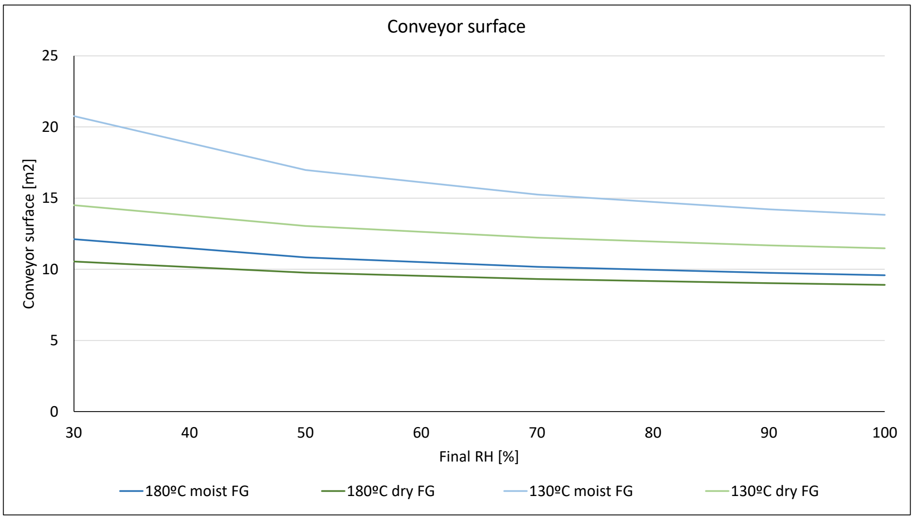


Figure 3.10 – Comparison of the conveyor surface required depending on the initial FG temperature and the presence or not of water vapour content in it. $Size = 1\ tn_{daf}/h$, Initial water vapour content of moist FG: 0.08 $kgH_2O/kg_{Fuel,db}$, $V_{FG} = 0.65\ m/s$. Drying from 50 to 10 $\%_{wb}$

global exchange efficiency. The estimation of the excess oxygen according to the boiler load and the moisture of the fuel is based on figures from industry but these values may varies according to the boiler designs.

In this project, it is considered that the heat produced in the boiler feeds a distric heat system, but not how the heat is exchanged, neither the return and supply temperatures. These variations may affect to the efficiency of the heat exchanger. Instead, as state above, it is studied the heat demand, whatever the mass flow and temperatures of water are.

With regard to the dryer, no drying rate or thermal process has been calculated but the energy and mass balance due to the lack of free information and data available found about drying rates, diffusion coeffiecients and other thermal kinetic factors. Instead, the model has been developed based on works previously accepted and calculation assumptions widely used.

Setting the relative humidity at a high level such as $90 - 100\ \%$ might seem to be a critical aspect of the model but since the inlet design temperature of the FG considered is relatively high ($180\ °C$), this consideration has less impact compared to drying at a lower initial temperature of the FG, as can be seen in figure 3.10. The higher FG temperature at the entrance leads to a higher temperature at the outlet. At this higher outlet temperatures, the variation in the absolute humidity with the relative humidity is less sensitive.

The calculation of the conveyor surface is defined by dividing the FG by its density and by an estimated velocity of 0.65 m/s. This latter assumtion may be the most sensitive value since in is not correlated with other thermal variables in this model, but it plays an important role in the heat transfer coefficient. However, this value is used by other authors and studies like those from De Fusco [38] [20] and Holmberg et al. [58], [30].

The decrease of the fuel moisture content and its effects are studied considering that the demand by the DH network is set. The possible rise in the nominal power rate is not considered. Instead, the efficiency improvement by drying the fuel leads to a lower fuel mass flow and therefore a saving in operating costs.

Chapter 4

Economic Model

This chapter addresses the economic feasibility of the project. Studies mentioned in chapter 2 and other articles found in the open literature present different casuistry depending on the considerations or assumptions made about beneficts and costs. The revenues derived from installing a dryer is different depending on the origin of the heat source considered and its cost, the drying agent, estimated efficiency improvement or overproduction of electricity, heat or steam obtained, among others. There are also researches in which economic analysis compare different types of dryers.

In this economic model, the viability of implementing a dryer is evaluated according to the size of the dryer and the operating time. It is well-known that, within certain constraints, the drier the fuel, the greater the efficiency of the boiler. However, the running time and the associated benefits may not cover the costs of a large dryer. Instead, project viability may be at fuel moisture content in the boiler intermediate between initial and minimum possible.

The feasibility of the project is studied mainly using the Net Present Value (NPV) and the payback period, which are purely economical aspects.The environmental consideration should be considered in a deeper analysis. As will be described in chapter 5, the aim of this project is to evaluate the feasability of drying biomass that will be burned in a boiler that heat the water stream of a small-medium scale Dristict Heat network. The drying medium considered is the boiler flue gas (FG)) itself, that is named in the literature as waste energy or secondary energy. Thus, the revenue is the fuel saved due to the improve in the efficiency because of the drying process.

As in chapter 3, the economic model has been developed in Python language.

4.1 Investment Cost

With regard to investment costs, a new model has been proposed that differs from those found in the literature and a comparative analysis of them will be made. The capital cost considered by De Fusco,Holmberg et al. and proposed by [48] is widely used in many works alike. Nevertheless, this could may be inaccurate for the estimation of small conveyor surfaces, in the range of 2 to 25 m^2, which correspond to the current values obtained in this project.

The average between two investment cost models has been considered for the estimation of the dryer capital cost. These models are proposed in [59]. In these models, a Free on board (FOB) cost is defined and the Bare module (BM) method is applied in order to obtain the cost of a fully installed and functioning dryer. Costs are defined in dollars and a factor of 0.92 €/$ is used. The estimate cost must be analysed considering an error marging of ± 30 %.

The FOB is defined in terms of a base cost multiplied by a ratio of sizes raised to the power "n": $Cost = Cost_{ref} \cdot (A/A_{ref})^n$. The Chemical Engineering Plant Cost Index (CEPCI) allows to updating the investment cost up a certain date and compare different capital cost obtained at different dates. In [59], the CEPCI index is equal to 1000.

One of the models retrieved from [59] is defined as a Belt/Band with flow through circulation (convection) and it includes fan, motor and feeders. This model presents two reference sizes for two different ranges that do not intersect, $1 - 10\ m^2$ and $20 - 40\ m^2$. At the intermediate range, between $10 - 20\ m^2$, a new funcion based on the equation aforementioned considering 10 m^2 as the reference size. Assuming that $Cost_{20m^2} = Cost_{ref,10m^2} \cdot (20m^2/10m^2)^n$, the cost function in this range is defined.

The second model defined is a solid band dryer based on convection and including fans, motors, conveyor feeder for band $> 6\ m$ and band width 1.2, 1.5, 2 and 2.8 m. Al the information about the develpment of the BM module is shown in table 4.1. All the factor estimated are defined by the model selected, such as Labor and Material factor, f_{LM*}, or general estimated itermediate values within the ranges suggested by [59] in order to develop the BM model.

Characteristic		Values		
		Belt Band		Solid Band
Base Cost	C_o	$322,000$ €	$414,000$ €	$202,400$ €
Reference area	A_o	10	20	10
Range of validity		$1 - 10\ m^2$	$20 - 40\ m^2$	$1.5 - 25\ m^2$
Scaling factor	n	1		1.04
Design and operational factors	f_d=f_{op}	1		1
Labor and Material factor	f_{LM^*}	2.4		2.4
Texes, freight and insurance factor	$f_{Freight}$	15 %		
Off-site + indirects for home office and field expenses factor	f_{Eng}	20 %		
Contractors fees factor	f_{Fees}	5 %		
Contingency for unexpected delays factor	f_{Delays}	10 %		
Chemical Engineering Plant Cost Index of reference	$CEPCI_{Ref}$	1000		

Table 4.1 – Parameters of the Bare module (BM) model proposed by [59]

In order to compare the investment cost model developed in this project with the others mentioned in the literature review (chapter 2), it is necessary to update all values to the same date or reference. In almost all the works analysed it is possible to trace the reference year. However, in the case of Myllymaa et al. [39], it will be assumed that the data are referenced to the year prior to the publication of the article, that is, 2017. Data will be updated using the CEPCI index and comparison will be analysed at the reference CEPCI index, 1000. Figure 4.1 shows the investment cost of the different models as a function of the size. The CEPCI indexes needed to upload all the investment models are obtained from [60].

CEPCI allows to reference and update costs and prices from other years to the current one. It also allows to compare costs from different years by referencing the different costs to the same year. According to [61], CEPCI is merely a model that should be cautiously applied. In particular, it should not be used to escalate cost for periods greater than five years. The unit costs over the years could increase due to the higher features added to the budgeted element, such as sensors or safety elements. On the other hand, unit costs could decrease due to cheaper materials, increased automation in manufacturing or even by a decrease in demand or increase in offer.

Notwithstanding the above, this index has been used to compare different investment costs, even though some of these models developed back to 2002. The conclusions on this comparison should take this aspect into account.

It is important to make special mention of the CEPCI indexes in the different studies previously analysed. Considering the updated investment costs, this cost in 2018 would be about 70 % higher than in 2002. For instance, the investment costs considering the model proposed by Holmberg et al. [30] [32] [41] [58] would about 70 % more expensive nowadays and the feasibility study would show results that could vary significantly. Again, a more in-depth comparison should be made because the updating of costs may be distorted due to the difference in years of the different models.

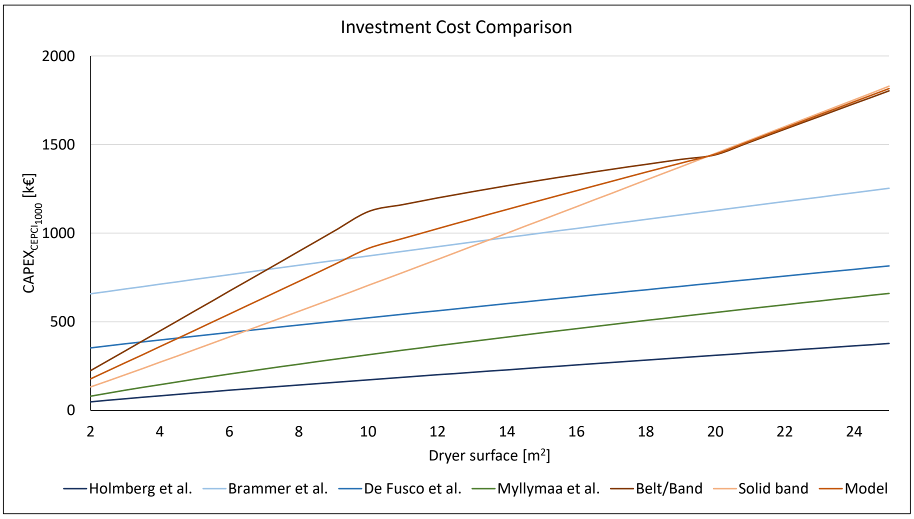


Figure 4.1 – Comparison of the estimated investment cost by the different authors analysed and those proposed

As can be seen in figure 4.1, the proposed model presents intermediate values to those obtained by the models analysed for the smallest areas. Due to the the scaling factor estimated by [59] is almost equal to 1, there is not capital cost advantage to building larger. The other models analysed show this adavantage at higher surfaces.

In conclusion, the values obtained in small areas with the proposed model are within the order of magnitude of the model used by De Fusco et al. [38]. The latter study estimates the costs as the average between Holmberg et al.[30] [58] and Brammer et al.[36]. It seems that models proposed by Holmberg et al. and Myllymaa et al. [39] may be underestimating the costs in small sizes. Nevertheless, it is possible that this model is more accurate than the one proposed here in larger sizes. Notice that the investment costs shown in figure 4.1 are compared at a CEPCI index of 1000. The cost analysis of the case studied in this project will be based on the CEPCI index in 2018: 615.9 [60].

4.2 Operation and Management Cost and Revenues

The operation of a DH or CHP plant and the biomass dryer have associated operating cost. Be it fixed or variable. Fixed costs are associated with operation and maintenance costs, i.e. constant costs each year that are not dependent on the running time yearly. This share of the total cost can be estimated as a perfectage of the capital cost. This way to calculate this fixed cost is commonly use in feasability analyses. In the literature review conducted in this project, different percentages have been estimated. De Fusco, in his Master Thesis [20], estimates 5 % of the investment cost in the base case. De Fusco et al. [38] study the mimimum plant size and minium running hours required at different O&M costs. Li et al. [31] estimates also a 5 % of the capital cost. Holmberg [32] consider 3 %.

In general, the literature available set this cost between 2 and 15 %. The higher percentages correspond to smaller installations, as the fixed costs tend to have a minimum cost independent of the size, such as the cost of the operators, the minimum space required, etc. Since in this project, the plant sizes considered can be defined as "small" or "medium"-scale, in the base case, the O&M cost will be fixed at 10 %.

The running of the dryer also has variable costs associated with the electrical costs of the fans that drive the FG. The power required by fans is defined in equation (4.1), where ΔP is the pressure drop in the dryer, η is the efficiency of the fan, $\rho_{FG,db}$ density of the FG and $\dot{m}_{FG,db}$ the mass flow rate of FG through the dryer.

$$P = \frac{\Delta P}{\eta \cdot \rho_{FG,db}} \cdot \dot{m}_{FG,db} \ [W_e] \tag{4.1}$$

Typical values for the efficiencty of the fans are around 95 %. Regarding the pressure drop, the literature offers different estimations of this value. Holmberg et al.[30] set values from 500 to 750 Pa; De Fusco et al.[38] consider 1500 Pa. Myllymaa et al.[39] obtain these values by experimental results and are in the range $500 - 900$ Pa. In this project, a pressure drop of 1000 Pa will be set.

The economic costs of this electricity expenditure are calculated by multiplying the power obtained in the equation by the cost of electricity and the running hours of the fan, i.e., the running hours of the plant. The price of electricity has been set at 184.8 €, all tax and levies included, that corresponds with the average price in Belgium in the second semester of 2019, for non-household consumers in the band $20 - 500MWh$ of electricity consumption [62].

The last variable cost of the plant considered in the model is the fuel cost. It accounts for the largest share of annual costs. The fuel mass flow is obtained from the biomass boiler as a function of the heat demand and the efficiency. As described in the thermodynamic model (chapter 3), according to the boiler model, once the FG temperature is set, and at a certain load, the efficiency will only depend on the fuel moisture content. The humidity affects the LHV_{ad} and the amount of air entering the boiler. With all that said, the fuel consumption will decrease when it is drier. Notice that, in this project, it is studied the optimum drying conditions in order to accomplish the demand instead of the increase in the heat/power production considering the same fuel mass flow.

In this model, the fuel price will be defined in $€/MWh_{LHV_{ad}}$, which is the price of the fuel per unit of energy available. Thus, the fuel cost will be calculated by multiplying the price by the LHV_{ad}, the fuel mass flow and the running hours (equation (4.2)).

$$C_{fuel} = b_{fuel} \cdot \dot{m}_{Fuel,ad} \cdot LHV_{ad} \cdot \tau \ \ [€] \tag{4.2}$$

In equation (4.2), τ is the running hours and b_{fuel} the fuel cost. It is important to mention that this equation, as well as the electricity costing described above, is valid when considering a constant load. Otherwise, the sum of the different annual charges should be considered, as can be seen in equation (4.3) and (4.4). Note that electricity (b_{elec}) and fuel (b_{fuel}) prices and biomass moisture are assumed to be constant over the entire period studied, as well as the pressure drop, the efficiency of fans and FG density.

$$C_{fuel} = b_{fuel} \cdot LHV_{ad} \cdot \sum_{i}^{n} m_{Fuel,ad,i} \cdot \tau_i \ \ [€] \tag{4.3}$$

$$C_{elec} = b_{elec} \cdot \frac{\Delta P}{\eta \cdot \rho_{FG,db}} \cdot \sum_{i}^{n} \dot{m}_{FG,db,i} \cdot \tau_i \;\; [€] \tag{4.4}$$

Figure 4.2 shows the contribution of each cost estimated in the total cost at a certain conditions described. It is expected a decrease on the fixed cost contribution since it does not depend on the running time, while fuel cost and electricity cost do. The latter can even be neglected, as Li et al.[31] do. It is the fuel expendidure what presents the greatest contribution. This contribution is very sensitive to biomass price. As will be described below, the benefits of drying biomass correspond to the fuel saved except for the costs of operating the dryer. The increase of fixed costs coupled with the decrease of biomass fuel prices can result in a worsening of the plant's operating costs rather than an improvement.

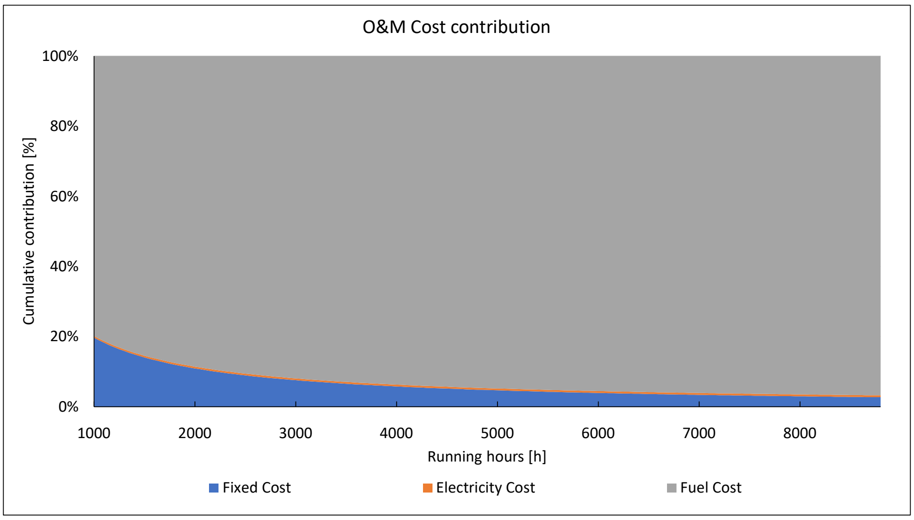


Figure 4.2 – Contribution of the different identified costs in the total costs as a function of the running hours. Plant size 6 MW, initial moisture content 40 $\%_{ad}$, final moisture content 22.3 $\%_{ad}$ (utilizing all available FG), Fuel cost 24 $€/MWh_{LHV_{ad}}$

The feasibility of installing a dryer will be studied mainly on the basis of two variables: the time of operation of the plant yearly, purely economic, and the share of FG used for drying. The second variable can also be converted into the dryer size (Conveyor surface) and final fuel moisture content. Notice that these variables are not proportional to the share of FG since the drier the fuel is, the less FG generated.

The objective is to find out from how many hours of operation are required for the installation of a dryer to be profitable, and which size and fuel final moisture content, make it feasible. At very low annual running hours and high investment cost (high fixed cost associated), the O&M cost may be higher than the cost of fuel saved.

4.3 Criteria decision

In order to establish a criterion for deciding on the viability of the project, different parameters that are sufficiently objective and consistent are set. In this project, the Net Present Value (NPV) will be evaluated as well as and the payback period. In addition, environmental issues such as post-treatment system solutions should be placed as paramount within further discussion that fall out the scope of this paper.

The NPV in this project is defined by equation (4.5), where C_{invest} is the investment cost, R_t the annual revenues, $C_{O\&M}$, operation and management cost and i_{dis} the discount rate.

$$NPV = -C_{invest} + \sum_{t=1}^{t=k} \frac{R_t - C_{O\&M}}{(1 + i_{dis})^t} \tag{4.5}$$

The investment cost C_{invest} is determined by the surface area, which in turn is defined by the amount of FG produced and its share used for drying. The operation and management, $C_{O\&M}$ cost is the sum of the fixed cost and the electricity cost defined in section 4.2.

The annual revenues, R_t (equation (4.6)), is defined as the cost of the saved fuel due to the boiler efficiency improvement compared to operating without dryer. On one hand, the FG share in the dryer will determines a fuel final moisture content. This fuel with a ceratin humidity will be burned. The more FG used for drying, the less humidity and the higher efficiency. This efficiency improvement means lower fuel consumption for the same demand. the difference to the fuel mass flow (in $kg_{fuel,daf}/s$) burned without drying will be greater. On the other hand, the greater the operating time results in a greater amount of saved fuel (in $kg_{fuel,daf}/year$).

$$R_t = C_{fuel,ini} - C_{fuel,dried} \tag{4.6}$$

In equation (4.6), $C_{fuel,ini}$ [$€/year$] is the cost of the fuel running without a dryer, i.e. initial conditions. $C_{fuel,dried}$ is the cost of the fuel drying it up to a certain final moisture content.

Li et al.[31] set the maximum payback period at $3-4$ years. De Fusco et al.[38] study the minimum viable plant size by determining the present value in the fifth year. Holmberg et al.[30] evaluate the amortisation time. In this project, the payback period required in order to consider feasible the invesment is set at 5-7 years. The life span is set at 15 years in this project. In the literature, values from 10 to $20-25$ years of lifespan can be found.

In addition to these economic criteria assumed in this project, other aspect such as environmental issues, post-treatment system dimensions... should be taken into consideration in a deeper analysis of the feasability.

4.4 Limitations

The assumptions described above imply certain restrictions and limitations when interpreting the results obtained. Some of these constraints may lead to interesting future work for the improvement of the model developed in this project.

The proportional raise of fixed cost, defined as percentage of capital cost, may not be appropiate in very wide size ranges. Fixed costs ($\%_{CAPEX}$) tend to be higher in small facilities due to minimum expenditure that is independent of size and arise from minimum operation conditions. For instance, the cost associated with the operators in charge of managing the dryer can be constant for a large range of sizes. On the other hand, it is the number of spare parts and or maintenance actions what vary the cost depending on the size. As a conclusion, setting fixed costs according to size may differ from a real estimation if the size range studied is very extensive.

Figures for medium-scale plant size are in the range $3-20$ MW and the dryer surface obtained for the minimum moisture content possible are around $4-24\ m^2$. This range can be compared with those obtained, for instance, by [39], whose capital cost model is valid from 0 up to 480 m^2. Thus, the variation of this percentage of the capital cost according to the size may no have a significant effect in the economic feasibility.

The estimated costs do not discriminate whether the drying agent is air or flue gas. Nor are there any references to this in De Fusco [20], whose study considers the same investment costs (table 2.2) for each element (dryer, fans...) without differentiating between FG or air. The use of FG as drying medium can incur extra costs due to post-treatment processes or the increased cost for facility materials to prevent effects such as corrosion or to resist acid condensation.

This aspect not only influences investment costs, but can also lead to higher maintenance costs. Further studies could determine this effect. In this study, the results that can be obtained are approximate and with wide margins of error. It is possible that this effect did not have a significant impact on the results obtained.

According to the thermodinamic model, improvement in the efficiencity decreases the amount of FG and and the excess air curves. This aspect could lead to smaller and affordable FG post-treatment systems and its maintenance cost.

The fuel price estimated is defined by a constant price per unit of energy available in the fuel, i.e. $€/MWh_{LHV_{ad}}$, whatever the fuel humidity as received is. This means that the price per kilogram of fuel will decrease proportionally with the fuel moisture content. The fuel price in the base case here studied, obtained from the industry, is set at 24 $€/MWh_{LHV_{ad}}$. Thus, the price per unit of mass decrease linearly from around 94 $€/tn_{fuel,ad}$ at a moisture content of $20\%_{ad}$ down to 40 $€/tn_{fuel,ad}$ at 60 $\%_{ad}$. This consideration may not be totally correct since the price per energy unit of dried fuel may be even more expensive than the price of a wetter fuel.

Chapter 5

Case Study. A Representative Distict Heating Plan

In accordance with what has been described in the previous chapters, a case study is proposed below. Firstly, the analysis will be carried out under nominal conditions, that is, constant heat demand equal to the nominal heat power of the boiler. the minimum operating time that makes the project viable will be studied along with the economically optimal fuel moisture content in the boiler. Secondly, the feasibility of the installation of a drying according to a realistic load curve will be studied. The analysis is framed in an existing medium-scale district heat production plant, in order to study the feasibility of introducing a biomass dryer for improvement in efficiency and fuel savings.

5.1 Description of the case study

In this project, the economic feasibility of installing a biomass dryer in a medium-scale district heat plant is studied. This plant consists of a biomass boiler as the main heat producer. An ancillary equipment, as can be a gas boiler, is considered to cover peak demand, i.e. demand above the nominal output of the biomass boiler. This ancillary equipment will also provide heat demand below the minimum capacity of the biomass boiler. The scheme of this case study is shown in figure 5.1.

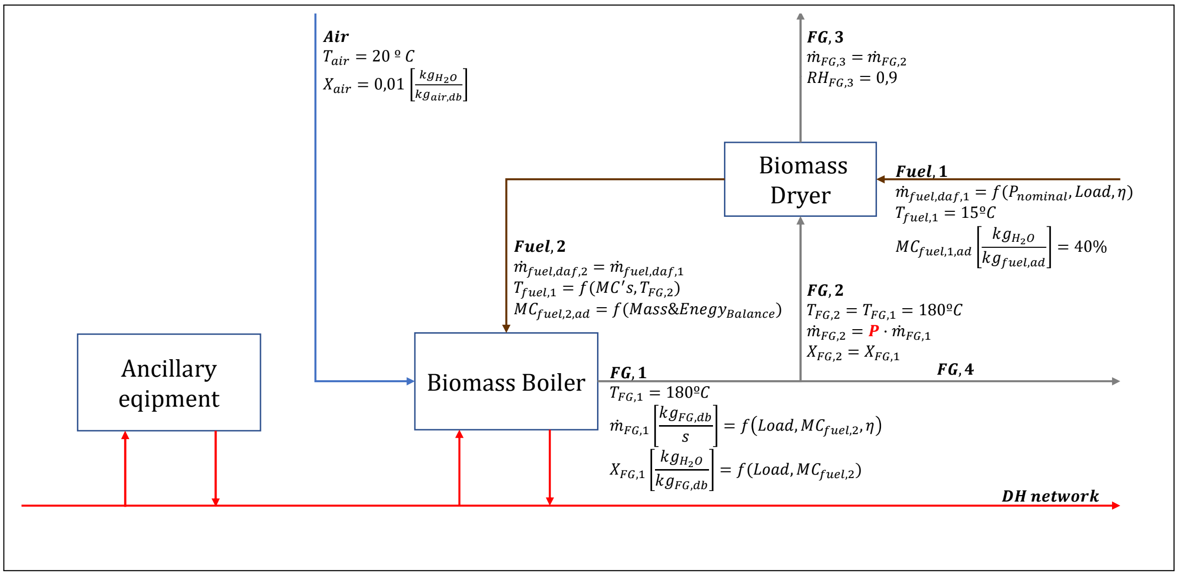


Figure 5.1 – Case study, Dristict Heat (DH) plant diagram

The drying medium used in the dryer is the biomass boiler's own Flue Gas (FG) flow. The energy available in this flow will be determined by the FG mass flow and its properties of temperature, absolute humidity and composition. As described above, the temperature of these gases is set at a function of the partial load without regard to the fuel humidity. As the energy available for drying the biomass is conditioned by the operation of the boiler, the fuel can be dried to a minimum water content, which will correspond to the use of all available energy.

Despite the existence of this limitation, the use of this waste energy is justifiable by various factors, both at the operational level and at the cost level:

- The valorisation of this FG and the use of it instead of considering it as a loss.
- Compared to using air, costs such as the investment in a heat exchanger or the cost of the heat source to heat this air are avoided.
- Less fire hazard compared to air since there is less O_2 concentration.

Although it is generally preferable for the fuel to be as dry as possible before entering the boiler, from an economic perspective, operating and investment costs may not compensate for this optimal energy performance. Under this premise, the improvement in the efficiency of the boiler, i.e. the fuel saving, as a function of the percentage, P, of the gas produced that is used for drying is analyzed. The share of FG set leads to a dryer conveyor surface, i.e. capital cost.

Once the dryer size is calculate, the fuel saved, in kg_{fuel}/s, is obtained. Due to the fixed and operating costs (electricity cost) of the dryer, there will be a minimum number of running hours after which the fuel saved will be at least sufficient to cover these costs. This point will depend strongly on the price of fuel and fixed costs.

The main parameters needed to define the thermal part of the case study are shown in table 5.1. Fuel composition is constant and defines the inputs of the boiler model programmed, along with the moisture content of the fuel before being dried and the FG temperature curve. The excess air in the boiler will be calculated as described in section 3.1.1. The FG dry composition will vary according to the excess air, while the water content will depend largely on the humidity of the fuel fed into the boiler. Notice that the FG velocity is used to calculate the conveyor surface.

Input themodinamic model		
Fuel composition	$[\%_{weight,daf}]$	
	C	50
	H	6
	O	44
	N	0
	S	0
Ash content	$[\%_{weight,db}]$	2.7
Plan size, Boiler nominal heat rate	$P_{boiler,nom}$ $[MW]$	6
Initial moisture content	$MC_{ini,ad}$ $[\%_{ad}]$	40
FG temperature at full load	$T_{FG,inlet,P_N}$ $[°C]$	180
FG temperature at minimum load (30 %)	$T_{FG,inlet,30\%P_N}$ $[°C]$	100
Initial Fuel temperature	$T_{Fuel,inlet}$ $[°C]$	15
Final FG relative humidity	RH_{out} $[\%]$	90
Velocity of FG in the dryer	V_{FG} $[m/s]$	0.65

Table 5.1 – Thermodynamic model input parameters

As previously explained, the variation in the humidity of the fuel in the boiler, as well as the boiler load, will determine the excess air and the absolute humidity of the FG.

Figure 5.2 shows the variation of the water vapour content of the FG and the oxygen concentration at nominal conditions (full load) according to the excess air curve defined in the boiler model. Notice that the excess oxygen set increase linearly betweeen 30 $\%_{ad}$ and 40 $\%_{ad}$.

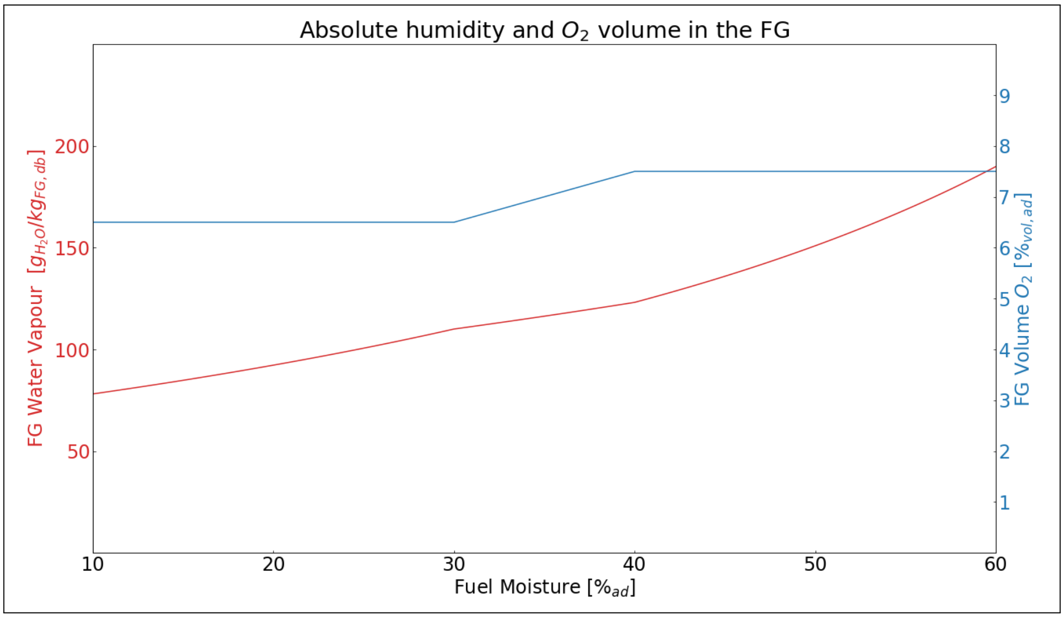


Figure 5.2 – Water vapour in FG and oxygen content as a function of the fuel moisture content at the inlet of the boiler (full load conditions)

As can be seen in figure 5.2, the wetter the fuel at the inlet, the more vapour content in the FG and the more enthalpy, but less capacity to absorb water from moisture evaporation of the fuel since the FG is higher partially saturated.

As a matter of fact, the temperature of the gases is estimated to vary according to the load, but not according to the fuel humidity in the boiler. This is in fact an approximation of the behaviour of a real boiler and the humidity of the fuel could have a non-negligible effect on the heat exchange phenomenon.

Concerning the dimensions of the dryer, it is calculated at nominal conditions. Below 100 %, the FG mass flow is maintained as long as possible. This means that, at a design percentage of 50 % of the FG produced at nominal conditions, the percentage of the produced gas introduced into the dryer will be greater as the load decreases. When 100 % of the FG produced is reached, it is not possible to maintain the nominal mass flow rate of the gas and the FG mass flow rate introduced will be less than the nominal mass flow.

In intermediate sizes, i.e. when the FG produced under nominal conditions is not fully used, the previous consideration assumes that the fuel-drying agent ratio in the dryer will be higher when the load decreases, so there will be more drying of the fuel. This advantage occurs until the gas introduced into the dryer reaches 100% of that produced by the boiler. After that, the final moisture of the fuel will increase again. Notice also that increasing the FG excess oxygen set when load decrease also increases the ratio of gas produced by the boiler to fuel fed into the boiler.

With regard to the economical aspects, table 5.2 shows the main parameters to consider. In this context, increasing the size of the dryer can lead to greater savings at the end of the dryer's lifespan. However, the increasing fixed cost can lead to longer payback periods. The price of fuel is a key aspect of the project's viability. In the case study a size of 6 MW

plant has been set. The increase in the size of the plant considered will affect the final savings and the viability of the project at lower operating times. Similar effect would be seen in the case of fixed cost.

Economic inputs		
Fixed cost	$C_{O\&M}$ $[\%_{Capex}]$	10
Fuel cost	$C_{fuel}\,[€/MWh]$	24
Electricity cost	$C_{elec}\,[€/MWh]$	184.8 [62]
Lifes pan	$years$	15
Discount rate	i_{dis} $[\%]$	5
CEPCI Index	$CEPCI_{2018}\,[-]$	615.9

Table 5.2 – Economic parameters

The improvement of plant efficiency can be analysed from two different perspectives. The first one is the boiler efficiency, calculated according to the equation(5.1) resulting from equation (3.6) at the boiler inlet. In this case, it is calculated with respect to the humidity conditions of the fuel entering the boiler.

$$\eta_{boiler} = \frac{Q_{boiler}}{\dot{m}_{fuel,2,ad} \cdot (LHV_{2,ad} + \dot{m}_{air,2,ad} \cdot C_{p,air} \cdot T_{air})} \tag{5.1}$$

Secondly, the global efficiency is calculated as shown in equation (5.2) where it is compared with the energy fuel as received, i.e. before entering the dryer. This second parameter enables a comparison of the efficiency improvement with respect to operation without a dryer. This parameter compares the energy delivered and the energy avaialable in the fuel as received while in the boiler efficiency calculation, η_{boiler}, the energy delivered is compared with the energy available in the fuel at the inlet of the boiler.

$$\eta_{global} = \frac{Q_{boiler}}{\dot{m}_{fuel,1,ad} \cdot (LHV_{1,ad} + \dot{m}_{air,1,ad} \cdot C_{p,air} \cdot T_{air})} \tag{5.2}$$

The suffixes 1 and 2 refer to the states of the fuel according to the figure 5.1. $\dot{m}_{air,1,ad}$ and $\dot{m}_{air,2,ad}$ are the amount of dry air introduced into the boiler per mass of moist fuel as received and dried respectively $[kg_{air,dry}/kg_{fuel,ad}]$. While the moisture content in state 1 is constant and equal to the initial fuel moisture, the moisture content in state 2 varies as a function of the parameter P, among others.

In the case of boiler operation out of the nominal conditions and following a demand curve, the equation is rewritten according to the equation (5.3). Notice that the $LHV_{1,ad}$ is constant whatever the load since it only depends on the properties of fuel before entering the system $Boiler + Dryer$. $\dot{m}_{air,1,ad,1}$ will depends on the boiler load and the fuel moisture content entering the boiler.

$$\eta_{global} = \frac{\sum P_{boiler,i} \cdot N_{h,boiler,i}}{\sum \dot{m}_{fuel,1,ad,i} \cdot N_{h,boiler,i} \cdot (LHV_{1,ad} + \dot{m}_{air,1,ad,1} \cdot C_{p,air} \cdot T_{air})} \tag{5.3}$$

In order to evaluate cases of nominal conditions and load curve, and to compare both cases,two parameters have been defined. Firstly, the demand covered, D_{cov} (equation (5.4)). This represents the share of the total annual energy demanded by the network that is covered by the biomass boiler. Notice that this value is equal to 1 in case of full load conditions since the DH network load is assumed to match the boiler nominal heat rate.

$$D_{cov} = \frac{\sum P_{Boiler,i} \cdot N_{h,boiler,i}}{\sum P_{Load,i} \cdot N_{h,load,i}} \tag{5.4}$$

Secondly, the load factor, LF,(equation (5.5)) represents the annual use of the boiler. This parameter compares the total energy supplied by the boiler with the maximum energy the boiler can provide during a whole year under nominal conditions.

$$LF = \frac{\sum P_{Boiler,i} \cdot N_{h,boiler,i}}{P_{boiler,nom} \cdot 8760} \tag{5.5}$$

In equations (5.4) and (5.5),$N_{h,boiler,i}$ is the number of hours that the heat power provided by boiler, $P_{Boiler,i}$, is repeated during a year. Equivalently, $N_{h,load,i}$ is the number or hours that the heat power demanded by the network, $P_{Load,i}$, is repeated during a year. $P_{boiler,nom}$ is the nominal heat power rate of the boiler.

In a load curve, the demand covered represents how optimal the size of the plant is in relation to the network demand. It is common that the main plant facilities are not sized to meet the demand peaks, as this would result in higher minimum loads ($\approx$ 30 % of nominal power) and lower usage. This effect can be seen in the LF. High values of LF represents that the boiler runs at loads closer to the nominal. It is possible to find facilities with high LF, but this may lead to lower demand covered.

5.2 Full Load

5.2.1 Description

The full load scenario studies the effect of the variation of the FG share used in the dryer and the number of hours of boiler running. This time will coincide with the hours of DH network demand, being the demand covered equal to 1. In other words, all the heat demanded by the network is supplied by the the boiler and this demand is constant and equal to the nominal heat power rate of the boiler. The Load Factor (LF) is then simplified as the total number of hours that the boiler is operating over the total number of hours in the year.

Therefore, in this case, the minimum number of operating hours at nominal heat power is studied in order to ensure that the dryer project is profitable, together with the optimal economic fuel humidity.

5.2.2 Results

Figure 5.3 shows the main results from energy and mass balance proposed in the model. These figures are plotted as a function of the FG share introduced into the boiler,P. It can be seen also the size of the dryer (conveyor surface) for 50 and 100 % of the FG share used.

The two efficiency parameters described are shown in red, in a scale plotted ranging from 70 to 100 %. The boiler efficiency increase from 82 % without dryer up to 84.7 % ulizing all the FG available, this represents an increase of 3.2 %. As state above, this improvement is due to the decrease of the fuel humidity at the entrance into the boiler and the reduction of the losses in the FG because of the decrease in the oxygen level set. However, with regard to the heat delivered compared to the fuel energy available as received, it can be seen that the efficiency increases up to 88.7 % by utilizing the entire

FG, what means an increase of 8.2 %. This efficiency improvement can not be neglected, since the small reduction in the fuel mass flow has a significant impact due to the running hours and the price of biomass.

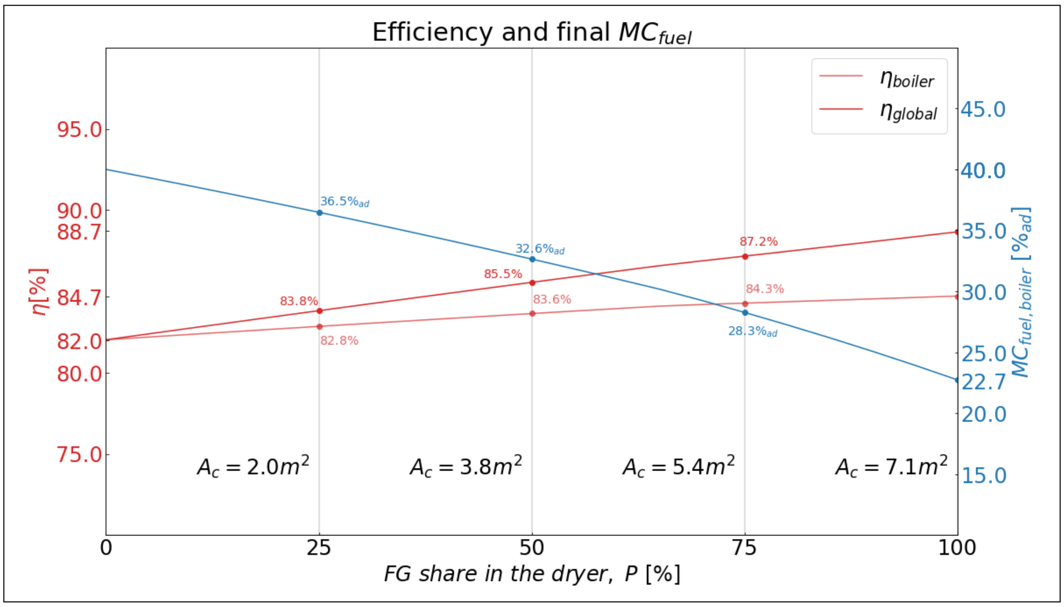


Figure 5.3 – Efficiency and fuel moisture content at the inlet of the boiler as a function of the share of FG used. Base case.

The final moisture content, i.e. the fuel humidity in the boiler is plotted in blue. Considering an initial value of 40 $\%_{ad}$, the minimum achievable humidity using all the waste heat in the FG from the system is around $22 - 23$ $\%_{ad}$[1].

As described in the section 3.1.1, the excess air in the boiler is calculated as a function of the target oxygen concentration in the flue gas. This excess oxygen set depends on the fuel moisture content in the boiler and its load. Two curves at two moisture content level and an their interpolation between these two levels have been set as a function of the load (figure 3.2).

The reduction of excess oxygen needed to achieve optimum combustion also means a reduction in the energy lost in the flue gases. In this case study, due to the assumptions considered of fuel moisture levels, between 40 and 30 $\%_{ad}$ of humidity occurs this joint effect in efficiency of decreasing excess oxygen content and fuel moisture.

Another issue highlighted in the previous sections is the variation in the amount of FG available for drying as a function of the humidity of the fuel in the boiler. By increasing efficiency and decreasing the fuel mass flow, the excess air will be less and therefore the drying agent flow produced will be lower. Figure 5.4 shows this effect as a function of the final moisture content reached.

The variation in the amount of FG produced between the initial conditions and the minimum humidity achieved is ≈ 17 %, which is equivalent to ≈ 3 tn_{db}/h avoided. Taking into consideration the operating hours, this drop in FG production may mean large amounts of waste energy being avoided and the corresponding emission of significant amounts of pollutants.

In addition, the reduction of the humidity of the fuel means a decrease in the enthalpy

---

[1] Considering the specific heat capacity of fuel, $C_{p,fuel}$, used by De Fusco [38] (equation 3.18), the final moisture content drops to $19 - 20$ $\%_{ad}$

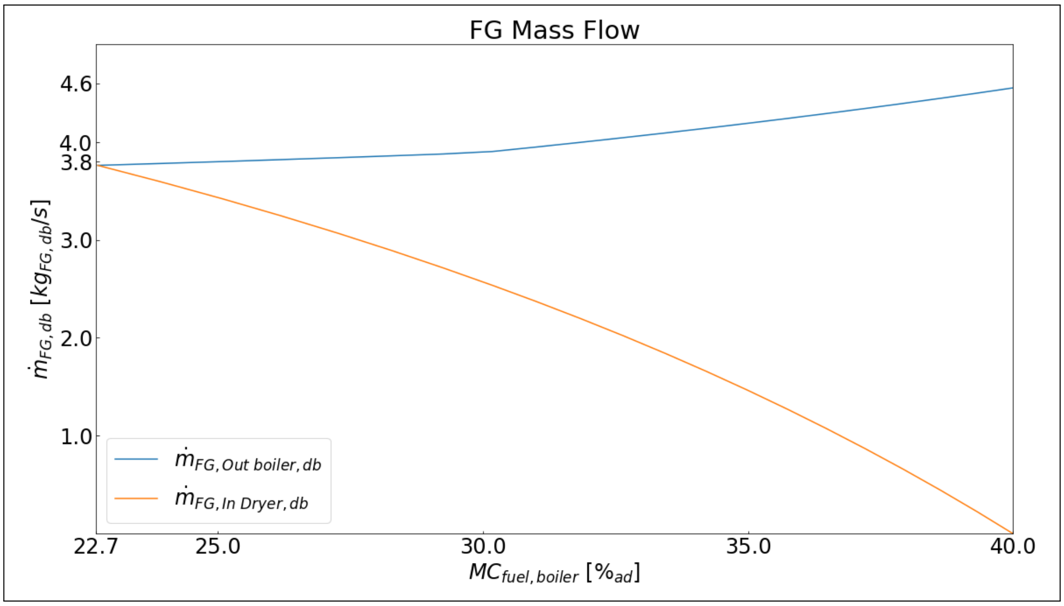


Figure 5.4 – Mass flows of FG produced and used in the dryer as a function of the fuel moisture content reached after being dried. Base case.

of the FG. On the one hand, the proposed boiler model does not consider the possible effect of either fuel moisture or ambient air conditions on the efficiency of the boiler exchanger. Thus, the FG temperature in this model depends only on the boiler load. Therefore, the sensitive enthalpy is constant at nominal conditions, regardless of the fuel moisture and other possible variables. On the other hand, the water vapour content in the FG does depend on the ambient conditions of the air and the fuel burned, both for its hydrogen content and its humidity. This implies a variation in the latent enthalpy of the gas.

Concerning the boiler secondary measures systems to reduce these pollutants, the reduction in the amount of FG produced due to the use of a dryer would mean a reduction in the size required for these systems and therefore lower investment and operating costs.This possible economic benefit has not been studied in this project.

Figure 5.5 presents the variation of the fuel mass flow, with two effects related to drying. On the one hand, drying the biomass means an increase in efficiency, which leads to lower consumption for the same heat demand (green line). On the other hand, the mass flow of fuel introduced into the boiler is considerably lower, falling by $\approx$ 30 % (from 2.63 to 1.87 Tn_{ad}/h) with respect to the initial conditions (orange line). However, the variation in fuel mass flow before the dryer decreases by less than 10 % (from $2,63$ to 2.41 Tn_{ad}/h). Notice that this two variables are the total mass flow of fuel (valuable fuel + ash + water). The evaporation rate is the water evaporated from the fuel transported by the FG (blue line), i.e. the difference between the other two curves.

This significant decrease in the total mass flow of fuel entering the boiler might cause an increase in the nominal capacity of the boiler: the useful heat contained in the fuel per unit of total mass (LHV_{ad} $[MJ/kg_{ad}]$) will be greater than in the initial conditions. In order to study the increase in the boiler capacity, several factors should be taken into account, such as the resistance of boiler materials to higher adiabatic flame temperatures or increases in flue gas temperature, among others.

However, in this case study, the possible increase in the nominal capacity will not be considered. Instead, the size of the boiler in terms of nominal heat power is supposed to

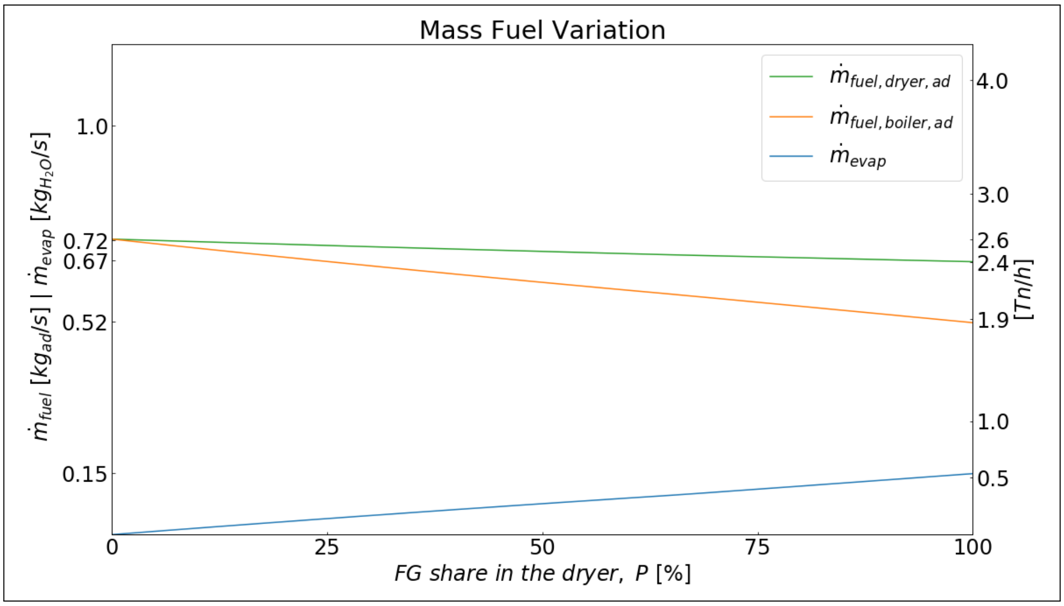


Figure 5.5 – Mass flows of fuel introduced into the dryer and into the boiler, and evaporation rate, as a function of the FG share used for drying. Base case.

be constant and equal to the initial value (6 MW in the base case).

There are two relevant aspects in line with the decrease in the fuel mass flow. On the one hand, the fuel transport systems to the boiler will be less loaded, as they will transport less fuel. On the other hand, by using all the available drying agent, about half a ton of water per hour is prevented from entering the boiler, which means about 90 $kg_{water}/h/MW_{produced}$, what can be valorised.

This evaporation means a higher energy density of the fuel. As mentioned above, the possible increase in boiler capacity requires a more throught analysis. The saturated flue gas resulting from the drying of the biomass could also be recovered, since the heat released by the condensation of the water vapour could be used. The more gas used for drying, the more water is removed from the fuel and the more energy can be extracted by condensing water vapour.

As a summary of what has been analysed so far, it is worth highlighting:

- The energy available in the waste heat (FG) can not be considered unlimited and therefore there is a minimum achievable fuel humidity.
- The energy available in the waste heat (FG) decreases as the fuel humidity in the boiler decreases. Less humidity means higher efficiency and therefore less fuel and air in the boiler. The enthalpy of the FG is also lower when the fuel humidity decrease since it contains less water vapour.
- The excess oxygen setting has a major effect on the model in calculations such as boiler efficiency.

As far as the viability of the project is concerned, the investment cost in the range of conveyor surfaces obtained ($< 10\ m^2$) increases linearly since there is not benefit of build larger (size factor, $n \approx 1$, in investment cost model). The margin of error considered is 30 %, as mentioned in [59]. Figure 5.6 shows this results and the moisture content as a function of the share of FG entering the dryer (i.e. the dryer size).

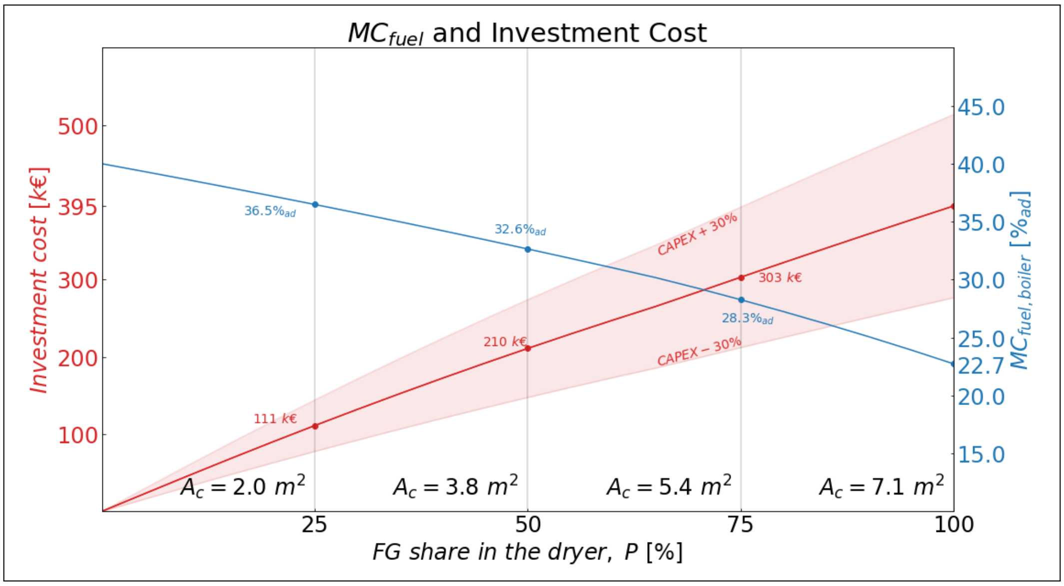


Figure 5.6 – Final fuel moisture content and investment cost as a function of conveyor surface. Base case.

This wide margin of error is an important variation and conditions considerably the viability of the project. Figure 5.7 shows the maximum present value achievable (red) at the end of the dryer's lifespan as a function of the running hours (Load Factor (LF)). In the case of the analysis of the realistic curve to be studied in the following section, the load factor obtained is 49.3 %. In blue, it can be seen the FG share that corresponds with this maximum present value.

It can be seen that the project begins to be profitable, i.e. positive balance at the end of the lifetime, from load factors around 70 %. The first profitable cases in terms of NPV occur in partial use of the available energy. It may be more feasible due to the increase in investment and fixed cost. This study presents two opposite general situations concerning to the NPV results: not investing in a dryer or using all the available energy. Optimal intermediate values of partial FG occurs due to model boundaries. The value of FG share of 65 % corresponds to the drying of the fuel just below 30 $\%_{ad}$ humidity. As described previously, at fuel humidity below 30 $\%_{ad}$, the excess oxygen in the flue gases is set constant. Therefore, drying below this moisture level would not compensate for the investment and fixed costs.

With regard to the margin of error considered for the capital cost, two main aspects should be highlighted:

- In the most favourable scenario, where the investment cost is 30 % lower, the results are analogous to those of the main result: at low operating times, the project is not feasible or begins to be viable by partially using the available energy. However, using all the energy to dry the fuel as much as possible soon becomes the best option (above 4000 *hours*). If the boiler were in operation all year round, the end-of-life NPV would be around 75 % higher than the main result (561 versus 320 *k*€).

- In the unfavourable scenario, the feasibility of the project occurs at very high load factors and only with a partial use of the available energy to reduce the excess air in the boiler.

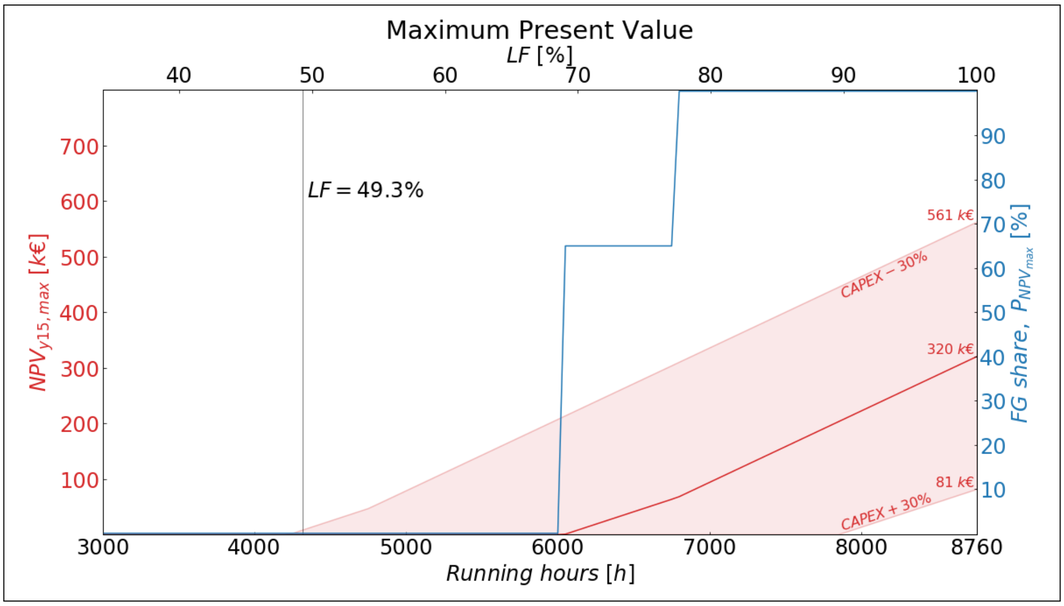


Figure 5.7 – Maximum NPV as a function of the running hours considered. Base case.

Once the feasibility has been analysed in terms of maximizing profits, or savings in this case, the payback period as a function of the running hours must be studied. Figure 5.8 shows the return on investment time for two FG shares: use half or all the energy available. These lead to two different sizes of the dryer and final moisture contents of fuel.

The results obtained show that in no case is the return period less than 5-7 years. However, achieving humidities around 30 $\%_{ad}$ means shorter return times. These results are due to the assumption that below 30 $\%_{ad}$ humidity, the excess oxygen remains constant and the differential improvement in efficiency by drying up to lower moisture content.

At larger dryer sizes and lower moisture contents, return times are higher, especially at low load factors. At LF close to 100 %, payback periods are similar. Thus, if the boiler is running for a lot of hours, the payback period is almost the same for the different shares of FG considered but the end-of-life benefits are greater when FG share is higher although the invesment cost is also higher. The uncertainty introduced by the margin of error of the investment costs and the associated annual fixed costs makes it possible that, in the most favourable case, using all the energy available is the best option, while in the most unfavourable case the implementation of a dryer must be rejected in any case.

In the literature reviewed, the estimated operating times to determine economic viability are 8000 h [38, 39] and 8400 h [31]. The figures 5.9 and 5.10 show the present value over the life of the project for each of these operating times. From the combined analysis of these graphs, the considerable effect of the estimated operating times on the viability of the project can be seen.

The investment uncertainty rise in proportion to the increase in size. Using all the energy available, in operating times of around 8000 hours, the project may even be economically unfavourable or bring in benefits of around half a million euros. On the other hand, at partial drying levels, this uncertainty is lower due to the smaller size and lower fixed costs involved. In both operation times, taking into account the return on investment time, it seems viable to partially use the available energy, although these return times can be very long in the most unfavourable case. However, there is not a huge difference of years between both cases and, regarding the NPV, The full use of the energy may be the

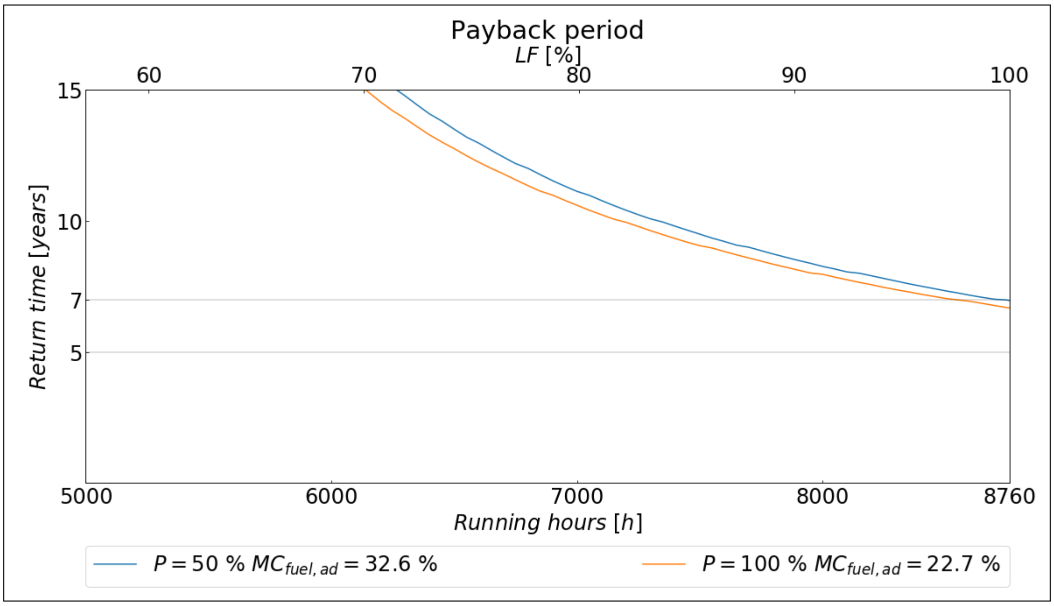


Figure 5.8 – Payback period of different FG share considered as a function of the running hours. Base case.

best option in case it is necessary to invest in biomass dryer.

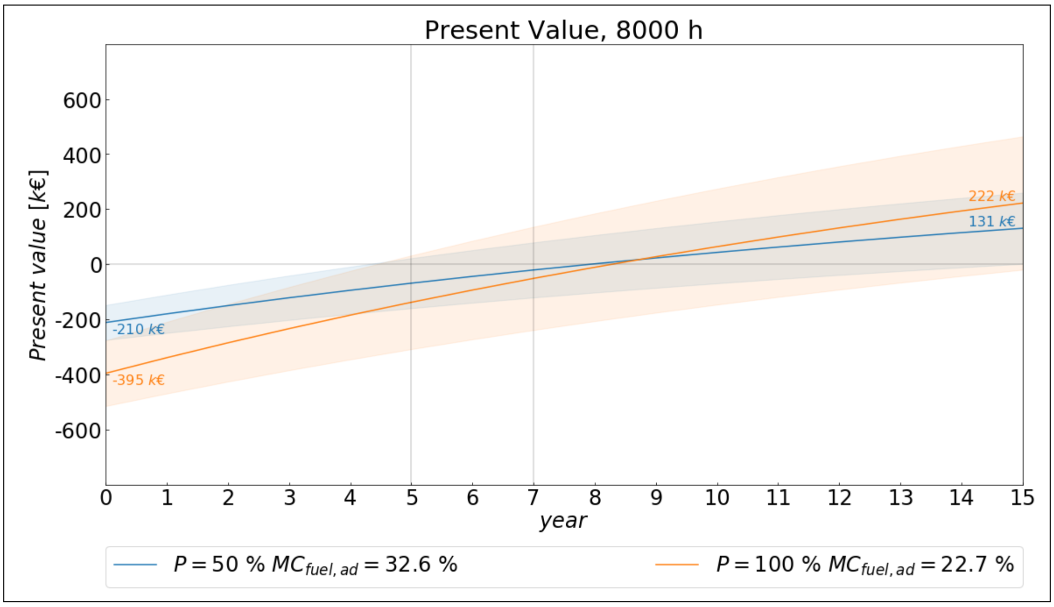


Figure 5.9 – Present value in two scenarios: Total and half use of FG, 8000 h. Base case.

As operating times decrease, the final economic benefits between partially drying or using all available energy approach, while investment costs remain the same. In fact, in the less favourable situation, partial drying will turn more viable. These values can also be seen in the figure 5.7, where, in the most unfavourable case, the maximum saving values occur at partial drying levels, there being an small benefit only at high operating times.

The results obtained for the base case ($Size = 6\ MW$, $MC_{fuel,ini,ad} = 40\ \%_{ad}$, $C_{fuel,ad} = 24$ €$/MWh_{ad}$, $C_{O\&M} = 10\ \%_{Capex}$) show that the viability of the implementation of drying is closely linked to the estimated operating times. The achievement of the minimum return time required would only be possible with partial use of waste energy

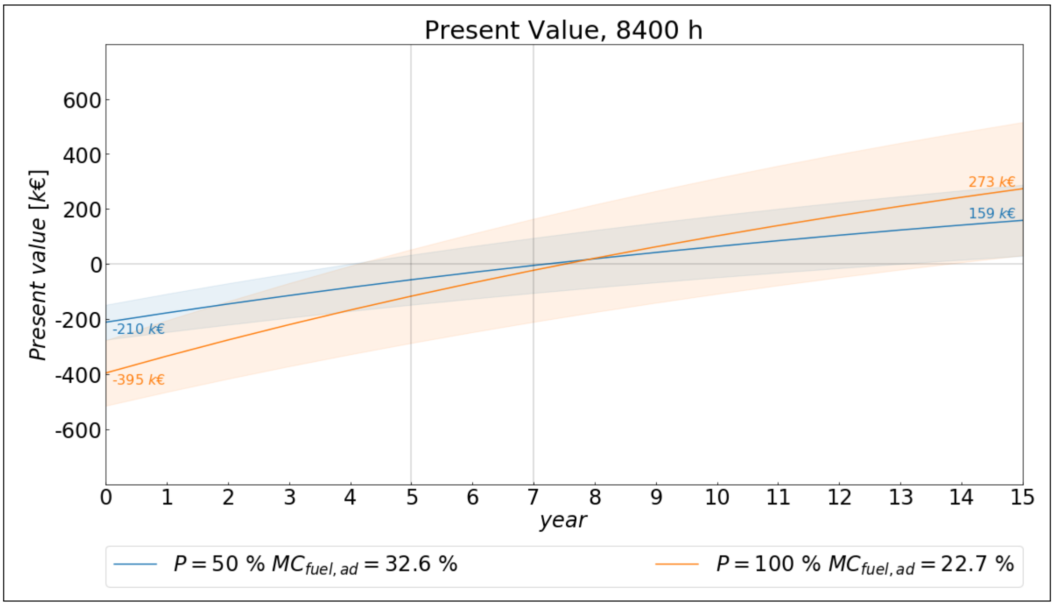


Figure 5.10 – Present value in two scenarios: Total and half use of FG, 8400 h. Base case.

from the boiler and high load factors, considering the mean values obtained. The economic savings obtained at high load factors by using all available energy are about 40 % higher compared to using half of this available gas.

In conclusion, in the proposed base case, a business case will exist only if the boiler operates for almost the entire year because the optimum humidities from an economic and thermal point of view[2] are the same. In other words, the more the fuel can be dried, the better, since the increase in fuel savings cover the increase in investment and fixed costs.

The fixed costs are linked to the investment costs and set at 10 %, so the annual savings on medium LF will be strongly influenced by the margin of error of the investment costs, as well as by the price of fuel. A decrease in this fixed cost will result in lower profit error margins for the same load factor and use of waste energy available.

Studying the margin of error of the proposed investment cost model indirectly allows a sensitivity analysis. As can be seen in the different figures shown above, investment costs have an important impact on the feasibility of the project. As shown in figures 5.7, 5.9 and 5.10, A change of ±30 % means a change of about ±70 % of the profit at the end of the lifespan at high operating times. Notice that this ±30 % also influences the estimated fixed costs. In the case where the investment cost was 30 % lower, the payback period would be less than 7 years from about 6500 hours of operation, and less than 5 years from 7500 hours.Therefore, the viability of the project is very sensitive to investment costs and special attention should be paid to these costs in a more detailed feasibility studies.

5.3 Real curve

5.3.1 Description

Once the system is studied under nominal conditions, a case of a realistic demand curve will be analyzed. The real curve has been obtained from a real plant with and adaptation

---

[2] As previously mentioned, the optimal humidity thermally speaking is the one that maximizes the efficiency of the boiler, that is, the minimum possible humidity achievable with the available energy. On the other hand, the economically optimal humidity is the one that maximizes the NPV.

to the values according to the nominal heat power of the boiler,6 MW. This adaptation has proceeded as follows:

- It is assumed that the demand is covered by a biomass boiler and an ancillary system.
- The biomass boiler has an operating range between 30 and 100 % of the nominal heat power.
- It is assumed that the biomass boiler is designed to optimize the demand covered by it, i. e., to maximize the total energy supplied by the biomass boiler with respect to the total demand.
- The real curve is readjusted by dividing this one by the optimal nominal heat power calculated according to the previous point and multiplying by the new heat power, i.e. the nominal heat power of the base case.

In summary, the actual demand curve has been translated into load factor values (Demanded heat power by the network / boiler nominal heat power). This load curve is multiplied by the size of the DH plant studied.

As a result of this readjustment, the curve shown in figure 5.11 is obtained. Since this is a Distict Heat (DH) demand curve, it can be deduced that the summer months correspond to the $3000 - 6000\ h$. In this period, a large part of the energy demand will be supplied by the ancillary systems, since the power demand does not usually reach 30 % of the nominal heat power.

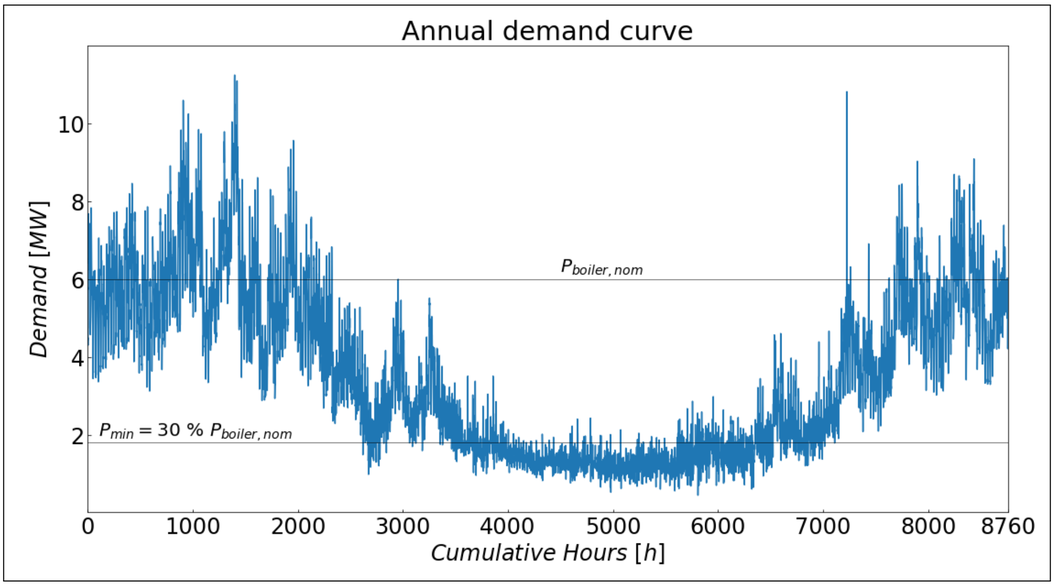


Figure 5.11 – Network demand curve. Base case.

Similarly, during the winter months, there are peaks in demand that must also be covered by ancillary systems while the biomass boiler is operating at nominal load. In order to analyze the demand covered by the biomass boiler and its operating load level, the monotonous decreasing demand curve has been developed. The original data in figure 5.11 presents a total of 8610 data records for the entire year. In order to reduce computation times, the demand values have been averaged, condensing them into 100 levels of average demand and the corresponding number of hours that this average demand is repeated.

The monotonous decreasing demand curve is shown in figure 5.12. The total demand covered by the biomass boiler is, as mentioned above, the maximum possible and equal to 84.2 % of the total demand. As mentioned above, the biomass boiler will only operate in the range $[30\ \% \cdot P_{boiler,nom} < P_{boiler} < P_{boiler,nom}]$. Therefore, at peaks of demand greater than 6 MW, ancillary equipment will also be used whereas below the minimum load, only this ancillary equipment will operate to meet the demand.

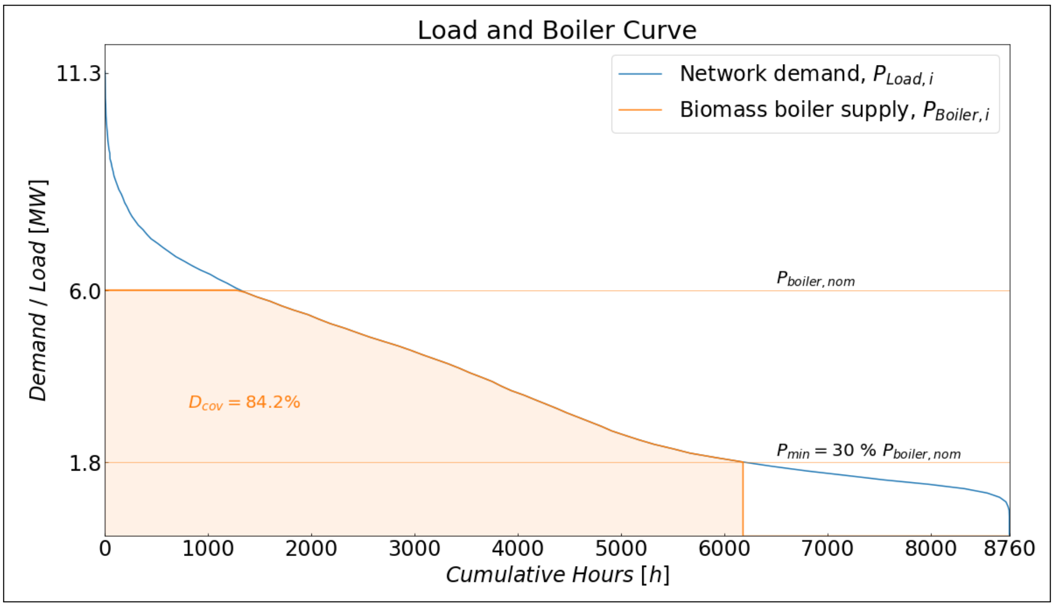


Figure 5.12 – Monotonous decreasing demand and boiler load curves. Base case.

As mentioned in the limitations section of the model (section 3.3) and discussed in section 5.2.2, the potential increase in the nominal capacity of the boiler due to the decrease in the humidity of the incoming fuel to the biomass boiler has not been considered. This increase could increase the demand covered by the boiler when it reaches high values. However, there could also be an increase in the minimum possible load on the boiler. Since it is unknown how the boiler will behave and it is not possible to estimate this variation due to different aspects, such as the chemical-thermal process or the resistance of the materials,the capacity of the boiler is considered to remain constant.

The Load Factor (LF) of the biomass boiler obtained is 49.2 %. It is important to highlight that in this case the boiler operates at different load levels leading to the different corresponding efficiencies estimated, unlike the case of constant demand and equal to the nominal boiler capacity. The increase in excess air at low loads can lead to an increase in the drying agent/dried fuel ratio in the dryer and therefore lower fuel moisture at the inlet of the boiler. Efficiency can be thus less impaired by the effect of partial loads.

The variation in efficiency as a function of load for different humidity levels, considering the FG temperature curve above mentioned can be seen in figure 3.5. In this case study, the fuel moisture entering the boiler varies as a function of the load, since it varies in different ways: The enthalpy of the flue gas will be lower at lower load values. On the other hand, in intermediate dryer sizes, the objective is to maintain the incoming gas flow, so the ratio $FG - fuel$ will be higher in the dryer.

According to the assumptions on which the boiler and dryer models are based, the efficiency and humidity values achieved vary depending on the load and the size of the dryer. This is shown in the figure 5.13, where the lighter curves correspond to the moisture of the fuel entering the boiler (secondary axis).

Following the criteria for the estimation of the dryer size, it is dimensioned according to the FG mass flow used in nominal conditions. When intermediate sizes are considered, i.e. partial use of the gas at nominal conditions, it is considered that the FG mass flow in the dryer will remain constant until 100 % of the gas produced is reached. As can be seen in the figure 5.13, in the case of $P = 50\% \dot{m}_{FG_{nom}}$, the gas mass flow rate in the dryer remains constant up to about $45 - 50$ % of the load. This means that the share of FG used with respect to that produced in the dryer increases from 50 % at nominal conditions to 100 % at load levels of $45 - 50$ %. Thus, below this level, this curve is consistent with the curve of $P = 100\% \dot{m}_{FG_{nom}}$. Below this load level, the gas produced is less than the design FG mass flow in the dryer.

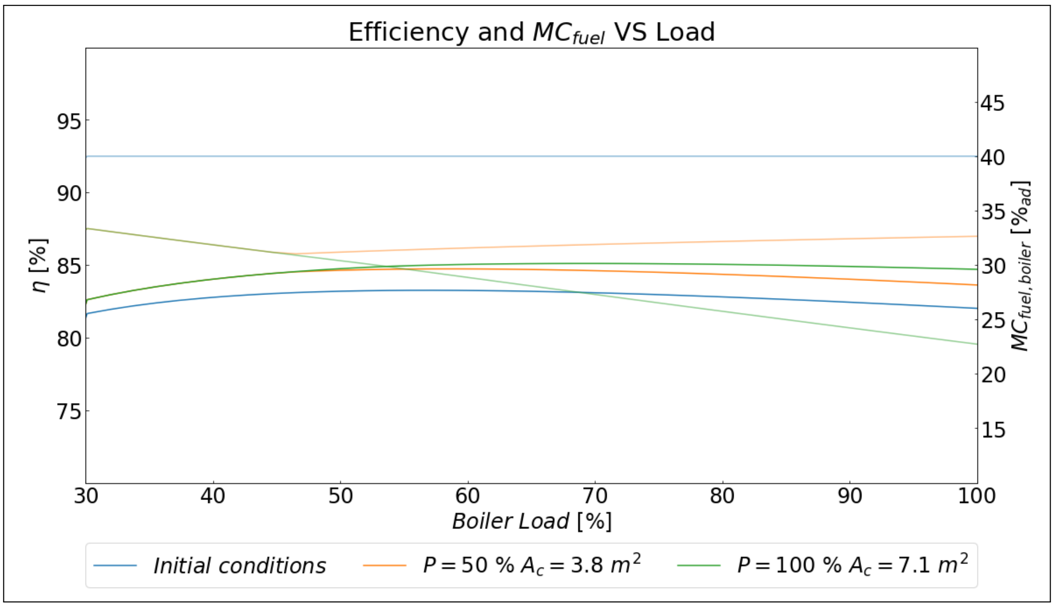


Figure 5.13 – Main results of efficiency and fuel moisture as a function of the boiler load. Base case. Note: lighter curves corresponds to MC_{fuel} levels.

As can be seen, by keeping the gas flow in the dryer constant, the $FG - fuel$ ratio in the dryer is higher and therefore a higher drying of the fuel is obtained compared to nominal conditions, which increases the global efficiency.

5.3.2 Results

The efficiency of the boiler throughout the annual period at different levels of FG share used and the efficiency at initial conditions of operation can be seen in figure 5.14. For more than 70 % of the boiler's operating time, it operates over 50 % of its nominal capacity, and about 20 % at full load.

According to the proposed model, at high partial loads, maximum efficiency is achieved. At low loads, losses through the walls are significant. At very high loads, the energy losses through the flue gases due to the increase of their temperature are more critical. Notice that as the load decrease, the $FG\ produced - fuel$ ratio will increase due to the increase in the O_2 content in the FG set.

These results should be interpreted with some caution. In this study, the effect of the boiler load has been modelled through different aspects: on the one hand, the variation of the temperature of the flue gas. The linear correlation between load and temperature leads to lower flue gas enthalpies al lower loads, which is beneficial. On the other hand, the excess oxygen curve also depends on the humidity of the fuel. This leads to a higher proportion of FG produced, which is detrimental in terms of efficiency.

Other aspects that may considerably affect the performance of the boiler at partial loads have not been considered. The control of excess air is carried out in order to minimise the formation of CO, which is formed due to incomplete combustion. Thus, it can be assumed that the presence of CO can be neglected even at low loads. However, a variation in load may affect the efficiency of the heat exchanger, and may aggravate or mitigate the loss of efficiency. The variation in FG temperature as a function of the load is intended to represent this effect. As state above, in this case it has been assumed that the efficiency of the exchanger will increase as the load decreases.

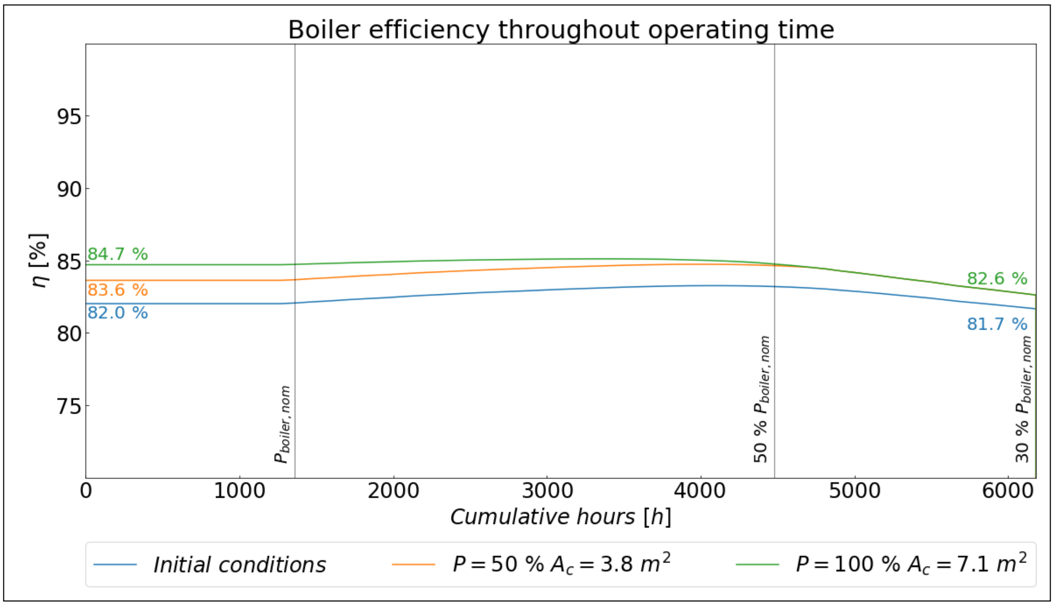


Figure 5.14 – Boiler efficiency throughout the monotonous decreasing demand curve . Base case.

As a result of the increase in the proportion of O_2 required in the boiler, there is an increase in the $FG-fuel$ ratio in the dryer when load decrease down to 30 %. This means that more amount of FG is available per kilogram of fuel. However, the dry enthalpy (without considering the water vapor content in the FG) is lower as the load decrease since the FG temperature decrease linearly.

The variation in the fuel moisture content throughtout the operating time can be seen in figure 5.15. Some aspects can be interpreted from this graph. Firstly, taking into account the $P = 100\%\dot{m}_{FG_{nom}}$ curve, it can be seen that, although the $FG - fuel$ ratio increases, the FG temperature is lower and, therefore, the energy interchangeable with the fuel decreases. This implies an increase in the humidity of the fuel entering the boiler. Secondly, according to the $P = 50\%\dot{m}_{FG_{nom}}$ curve, by maintaining the FG mass flow as much as possible, the $FG-fuel$ ratio is even greater and this counterbalances the decrease in the FG temperature, obtaining a drier fuel. This also can be seen in figure 5.13.

The dryer has been designed to operate under nominal conditions, i.e. according to the air mass flow nominal and the velocity through the bed, which is set at 0.65 m/s. Values such as the conveyor velocity and the height and width of the bed will then be set to obtain the desired FG RH at the outlet of the dryer and to dry the fuel in the

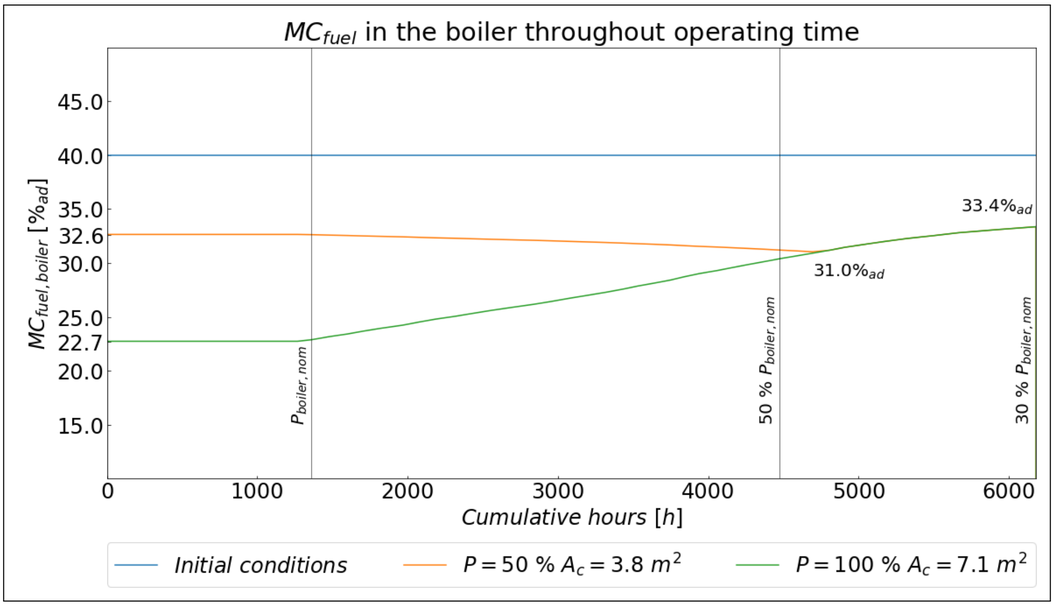


Figure 5.15 – Fuel humidity in the boiler throughout the monotonous decreasing demand curve . Base case.

most optimal way. Out of the nominal conditions, the mass flow of fuel and FG will decrease. Drying efficiency may be expected to decrease due to operation out of the design conditions. However, industrial dryer can be operated in order maximize efficiency by varying the bed height, conveyor velocity or by narrowing the flue gas passage areas to maintain the optimum velocity. Therefore, in this project it is considered that no additional energy loss will occur due to operation at partial loads.

In addition, as shown in the section 3.2.4, due to the high temperature and enthalpy of the FG introduced, the temperature at the outlet will also be high. The absolute humidity of the FG at relatively high temperatures ($60 - 100$ $°C$) is less sensitive to the variation in relative humidity. Therefore, the variation of RH of the flue gas at the outlet of the dryer due to partial load operation may not have a considerable influence.

As can be seen in figure 5.14 and according to the efficiency variation as a function of the boiler load (also seen in figure 5.13), the maximum efficiency of the boiler estimated by this model occurs at high partial loads. These load levels occur during most of the time the boiler is in operation. Therefore, a slightly higher global efficiency is to be expected compared to constant operation under nominal conditions, i.e. the case studied before. In figure 5.16 this can be seen in the initial situation, i.e. without dryer.

By increasing the FG share, the time the Flue Gas (FG) mass flow in the dryer is kept constant is less throughout the load boiler curve, so the fuel moisture will increase as the boiler load decreases. This leads to a greater variation in overall efficiency between nominal conditions and minimum load levels (84.7 % at full load down to 82.6 % at minimum load, what means a decrease of 2.5 % compared to a decrease of 0.36 % if initial conditions are considered). For this reason, in this study, the overall efficiency of the realistic case becomes lower than under nominal conditions when considering the larger dryer sizes.

A comparison of the global efficiency in the two cases studied shows hardly any difference between nominal condicions and realistic curce. The explanation resides in the efficiency of the bolier, which varies slightly with the load and that the greatest effect on efficiency is determined by the moisture of the fuel. Furthermore, maximum efficiency values do not occur at nominal load, but at high partial loads. Notice that, as can be seen in the graph, the boiler operates above 50 % load for most of the time, corresponding to maximum efficiency at operatings points. Although the curve studied is representative of a district heating network, aspects such as incorrect dimensioning of the heating system or its operation as an ancillary system, among others, could lead to low global system *Dryer + Boiler* efficiencies.

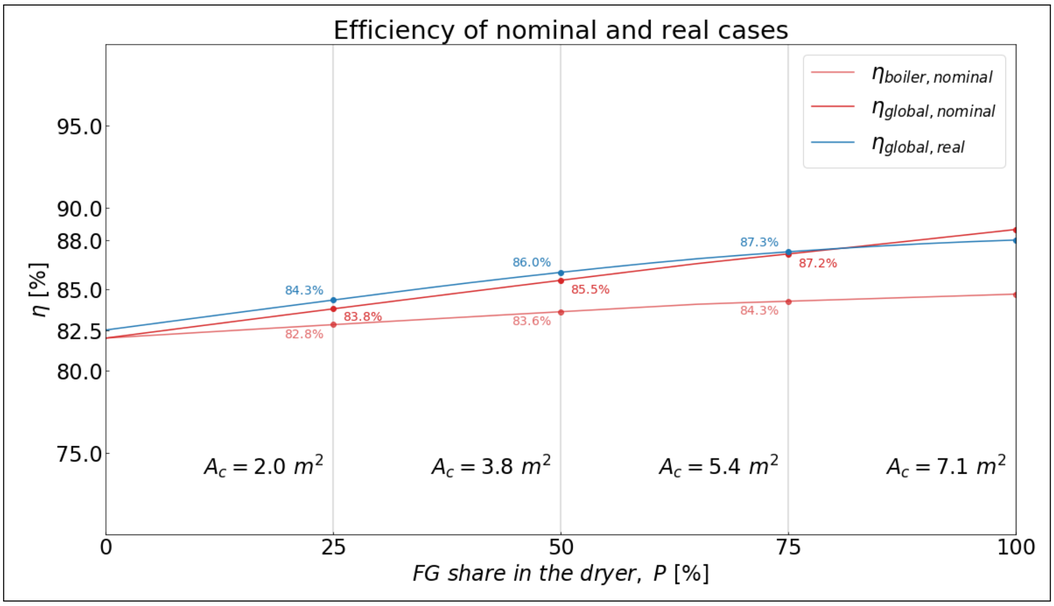


Figure 5.16 – Global efficiency, Full load and realistic demand curve comparison. Base case.

As a first approximation, taking into account the system Load Factor (LF), i.e. the number of equivalent hours operating under nominal conditions compared to the total possible in one year, it can be deduced:

- As mentioned above, the annual boiler load factor is 49.3 %. This load factor is not completely comparable to that obtained by considering the operation under nominal system conditions. However, the global efficiency, η_{global}, of the realistic case is very similar to the base case of nominal operation (figure 5.16). Therefore, in an approximation and attending to the graph in figure 5.7, it can be observed that there will not be business case under the assumptions made and the demand curve considered. The equivalent hours are not sufficient to compensate for the investment in the dryer and the associated operating costs.

- The size of the boiler is optimized for this demand curve. Looking at the monotonous decreasing demand curve(figure 5.12), there is a big difference between the maximum and minimum network demands. In addition, the boiler operates most of the time in partial loads. This results in the fact that, despite a high and optimal covered demand, the boiler load factor is relatively low. An alternative that would increase this load factor and the covered demand would be the use of several boilers with lower nominal heat power instead of one.

- Even the most favourable case of investment costs does not result in an acceptable business plan. If the investment costs are, on the contrary, 30 % higher, the investment could even mean an additional annual cost due to the higher fixed costs. These aspects can be seen in figure 5.17.

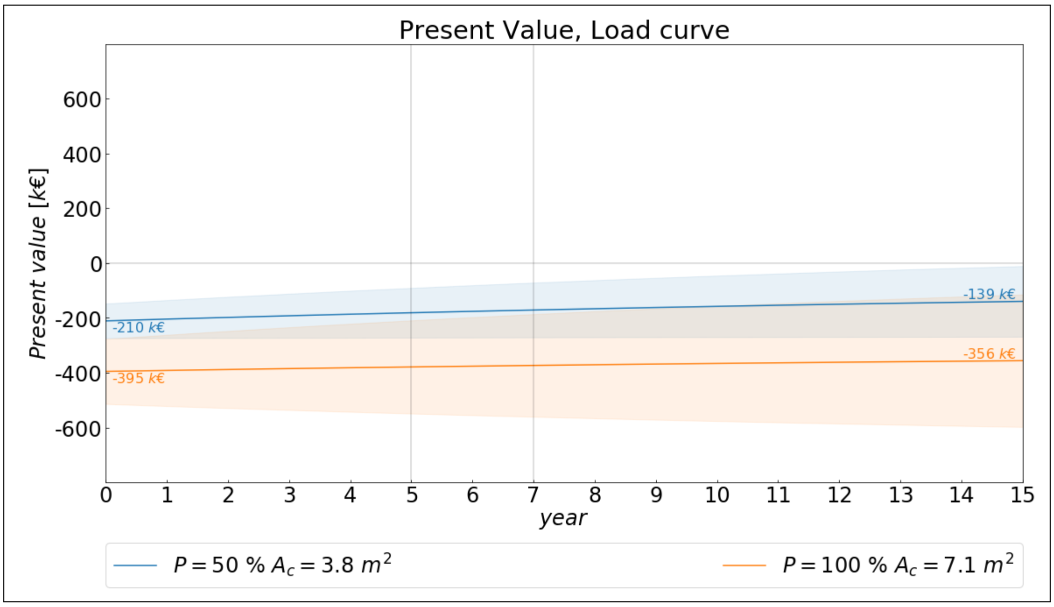


Figure 5.17 – Net present value according to the realistic demand curve. Base case.

5.4 Sensitivity analysis

This section addresses the impact on the feasibility of four of the design parameters set in the base case will be analysed. Concerning the thermodynamic model, the effect of the initial fuel moisture set will be analysed. Moreover, the minimum size of the plant that represents a business case under the model assumptions set in the base case will be evaluated. Regarding the rest of the thermodynamic parameters set:

- Composition of the woody biomass used as fuel in the industry, in terms of dry matter, slightly varies. The present work analyzes in broad terms the viability of the project. Compared to the margin of error to be assumed in this study, the fuel composition (dry matter) has no significant impact. The effect of the different compositions could be very important in more specific feasibility studies. For example, for the design of the boiler or the gas after-treatment systems. However, a change in fuel type, for example from wood to municipal waste (incineration plant), could have an appreciable effect at this analysis level.

- The relative humidity, RH, is set according to the optimization parameters of the dryer. The objective is to dry with the least amount of drying agent possible, avoiding condensation. This minimises the drying surface area required and therefore the size. On the other hand, when high drying agent temperatures are considered, the absolute humidity$(kg_v/kg_{FG,db})$ of the gas is not very sensitive to the relative humidity set (if high RH level is considered) (as explained in section 3.2.4).

- The FG temperature curve set is an approximation of reality that may or may not be true. Industrial boilers often have economizers that allow increased heat exchange efficiency at high loads and are bypassed at low loads to prevent condensation. Other hypotheses regarding these temperatures could be considered. The figures used in this study are selected from real studies and measurement campaigns in other boilers. In more in-depth studies, these values are not estimated according to other researches, but are obtained from the measurement or from the design of the boiler (for instance, CFD models).

- In this particular developed model, the FG velocity set does not influence the dryer thermodynamics, but it does influence its size. Studying its impact would lead to possible errors, as the surface of the dryer would vary, but the thermodynamic effect could not be quantified. Again, a more in-depth analysis would be required.

Concerning the economical parameters, two of them will be analysed: the fixed costs and the fuel cost. In both cases, there are studies in the literature in which these parameters differ considerably. Taking into account that the base case studied here does not present positive profitability results, this analysis intends to evaluate the effect of these parameters and the minimum (in the case of fuel costs) and maximum (in the case of fixed costs) values that would result in a business opportunity. Regarding the rest of the economic parameters, the figures set are those typically found in the literature and widely used in economic analysis.

5.4.1 Size of the DH plant

Due to the developed model, there is no variation in the thermodynamic results due to the fact that the efficiency and the fuel humidity achieved are the same, as well as the $FG_{produced} - Fuel$ ratio. However, in a more in-depth analysis, possible effects such as the installation of economizers and possible lower amounts of oxygen in the gases in the larger sizes should be taken into account. Therefore, the analysis conducted only focuses on the economic aspects.

In figure 5.18, the conveyor surface of the dryer obtained from this sensitivity analysis is seen. The proposed capital cost model expects a lower unit cost when the dryer surface is higher than 10 m^2. There is thus a higher benefit when the plant size is larger than ≈ 8 MW in case of full use of the FG energy. In the case of $P = 50$ %, this advantage will be reached from ≈ 16 MW plant size.

Regarding the payback period, the project begins to be viable when larger plant sizes are considered. In addition, the return on investment occurs at shorter operating hours. The return on investment as a function of operating hours and for different plant sizes is seen in figure 5.19. As the size of the plant increases, fixed and investment cost do so considerably. Regarding fuel savings, $(\dot{m}_{fuel,ad,initial} - \dot{m}_{fuel,ad,drying}) \cdot C_{fuel}$, it increase even more after a certain number of operating hours.

If the size of the studied plant is greater, it is foreseeable that the benefits at the end of the useful life will be greater considering the same hours of operation. However, in the scenario where it is decided to invest when the operating hours are low, the economic losses are greater the larger the plant size. For instance, if the operating hours were 4000 h, the economic losses when $P_{boiler,N} = 6$ MW rise to approximately 300 $k€$. In the case of $P_{boiler,N} = 20$ MW, the losses are about 430 $k€$. As seen, the losses per unit of heat power are lower, from ≈ 50 $k€/MW$ euros to ≈ 20 $k€/MW$.

Figure 5.20 demostrates the end-of-life benefits of the dryer for an operating time of 8000 hours. The dashed line shows the realistic case. The effect of the variation in the unit cost of the dryer explained previously can be seen. When $P = 100$ %, the benefits vary

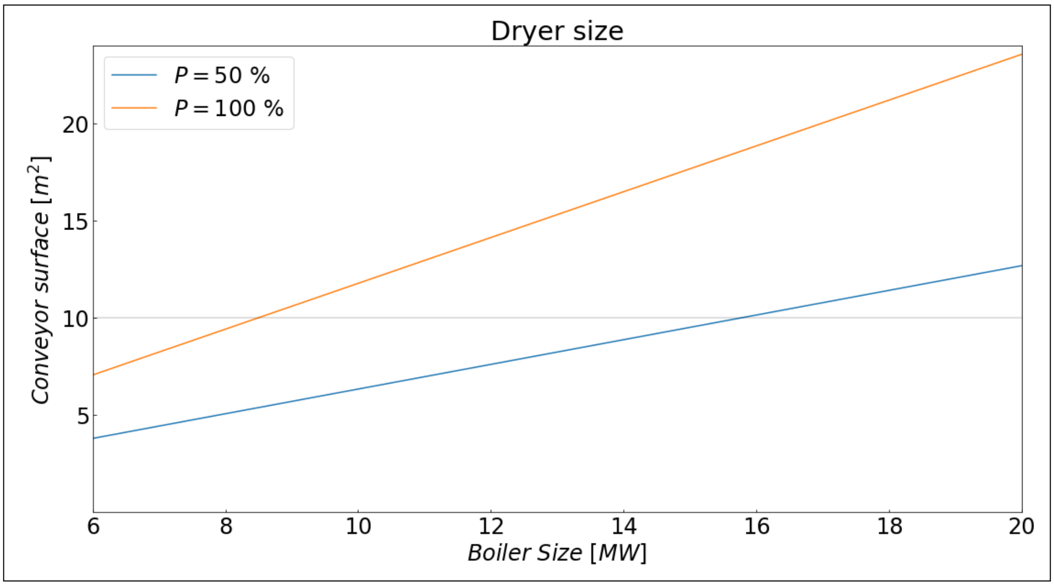


Figure 5.18 – Dryer size at two FG shares as a function of the boiler size

from $\approx 40\ k€/MW$ if the plant size is 6 MW up to$\approx 65\ k€/MW$ if 20 MW is considered. This represents an overal increase of $\approx 75\ k€$ per each MW that the plant is larger than the base case.

If the plant double the base case size (12 MW), the benefits would almost triple. In the scenario where the plant size was triple (18 MW), the benefits would be slightly more than five times the base case results.

Nevertheless, the savings generated in the realistic case($R_t - C_{O\&M}$) are not capable of providing a return on investment. The network demand curve may be decisive in a profitable installation of a dryer. Larger plant size would mean greater losses, despite what might be expected, mainly due to the shape of the demand curve and the plant design. A more modular design, with several biomass boiler, could lead to more favourable results.

In conclusion, the project is not economically profitable even at high operating hours under the premises and assumptions of the base case. However, a greater size would bring an acceptable payback period for more frequent operating times. If the size of the plant was doubled, the project would become viable from about 7500 h in a payback period of approximately 7 years. If the size were triple, the project would be viable from 7000 h of running hours.

5.4.2 Fuel Initial Moisture Content

There are several factors that can condition the type of wood available, as well as its humidity. The location of the plant is a relevant factor. In order to avoid transport costs, the use of the fuel in its surrounding areas is usually efficient, although it is not always possible to acquire fuel with the convenient humidity. On the other hand, weather conditions and seasons also affect fuel moisture. For these reasons, an analysis of the effect of the fuel initial moisture of has been carried out.

As mentioned in chapter 3, the initial moisture of the fuel conditions the final achievable moisture, since the initial enthalpy is higher and the wetter it is, the more energy is needed

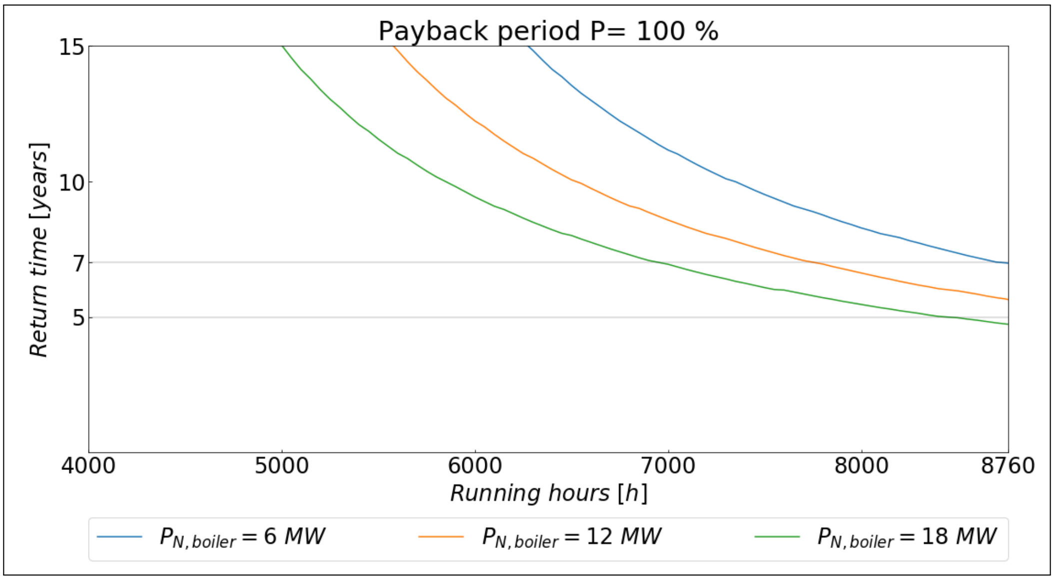


Figure 5.19 – Payback period of different plant sizes if all the FG energy available is used, i.e., the largest dryer sizes

to obtain the same moisture drop[3].

In figure 5.21, the efficiency variation, $\Delta\eta_{global} = \eta_{global,P} - \eta_{boiler,initial}$, as well as the humidity drop in the dryer, $\Delta MC_{fuel,boiler} = MC_{fuel,ini,ad} - MC_{fuel,boiler,ad}$, are shown. Notice that, using 100 % of the FG and with initial humidity below approximately 30 – 32 %, the minimum humidity possible is reached, 10 %, as explained in previous chapters. Therefore, either 100 % flue gas would not be used or other operating parameters would be modified. A close look at figure 3.3 helps understand this graph.

The definition of excess air curves set as a function of humidity and load has a relatively important effect on the model results. The model establishes a constant excess of oxygen above 40 $\%_{ad}$ and below 30 $\%_{ad}$. This results in two facts: on the one hand, when the initial humidity is higher than 40 $\%_{ad}$, the excess oxygen set drops and, together with the increase in the LHV, this leads to an extra improvement in efficiency. On the other hand, if the initial humidity is less than 30 $\%_{ad}$, there will be no variation in the excess oxygen in the FG and, therefore, the improvement in efficiency will be purely due to the increase in LHV.

For this reason, the greatest efficiency improvements occur when all available FG energy is used and when initial fuel humidities are between 40 – 45 $\%_{ad}$. There is enough energy to dry the fuel below 30 $\%_{ad}$, so the excess oxygen curve changes considerably from the maximum to the minimum values. As seen, these results are strongly conditioned by assumptions about the operation pattern of the boiler and the levels of excess oxygen. Other hypotheses such as a continuous decrease in excess oxygen as the fuel becomes drier or considering excess oxygen constant could also be valid. The latter is widely used in early feasibility analysis.

Regarding the payback period, the maximum improvement in efficiency occurs when the initial humidity is around 40 $\%_{ad}$, which coincides with the value considered in the base case. Therefore, a better business case is not to be expected, but the opposite. This is shown un figure 5.22.

If the fuel had a humidity of 50 $\%_{ad}$, the return time would increase by at least one year

---

[3] Drying from 40 $\%_{ad}$ to 30 $\%_{ad}$ requires less energy than drying from 50 $\%_{ad}$ to 40 $\%_{ad}$.

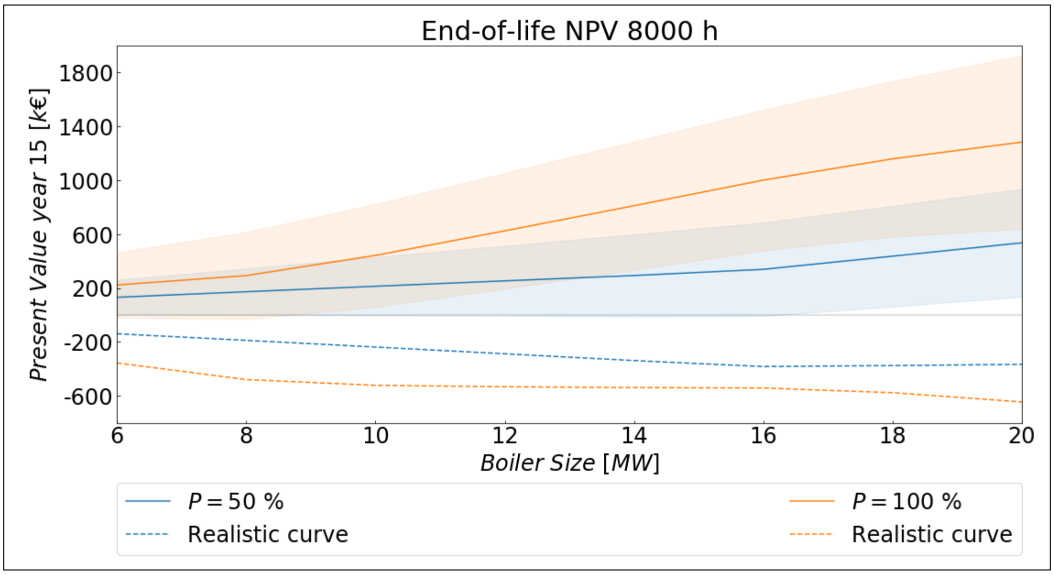


Figure 5.20 – NPV at the dryer end-of-life considering 8000 h operation at two FG shares as a function of the plant size considered. The dashed lines represents the realistic case. Note: the shaded area represents the margin of error for investment costs

with respect to the base case. If the humidity was 30 $\%_{ad}$, this increase would be more than 3 years. This would occur in the best case scenario, when the boiler load factor is ≈ 100 % (≈ 8760 h). In any case, the results, according to the decision criteria, show that the project is not viable whatever the fuel humidity. However, this sensitivity analysis would be very interesting if other economic parameters from the model were more favourable. In that case, the range of humidities within which the project would be viable should be evaluated.

In conclusion, as a rule of thumb, the drier the fuel, the better. The energy available in the gas is decisive, since it limits the possible drop in humidity. This available energy and the humidity drop also depends on the initial fuel humidity. The availability of an additional source of drying agent could be interesting to study when the fuel is very wet. However, this does not ensure viability, as the size of the dryer would be larger and therefore higher investment costs.

5.4.3 Fixed Costs of Operation

The parameter of fixed maintenance and operating costs are constant regardless of operating time, so they will have less influence as running hours increase. This has been estimated as a percentage of the investment costs, 10 %, regardless of the size of the dryer. Therefore, the fixed costs increase proportionally with the size of the dryer. This may lead to an overestimation of these costs in large sizes and an underestimation if the dryer is small. An analysis of the impact of this parameter on viability is studied in this section.

Figure 5.23 shows the effect of fixed costs on the return period as a function of operating time. As seen, reducing fixed costs by half, the project becomes viable from 7000 operating hours and the payback period is reduced by 2 years in the best case scenario (≈ 100 % boiler load factor, ≈ 8760 h). If the fixed costs increase by 50 %, the return time could increase by more than 3 years in the best scenario.

Compared to the effect of the size of the plant, the variation of the return period is more sensitive, since to reach an acceptable return period from 7000 hours, the size considered

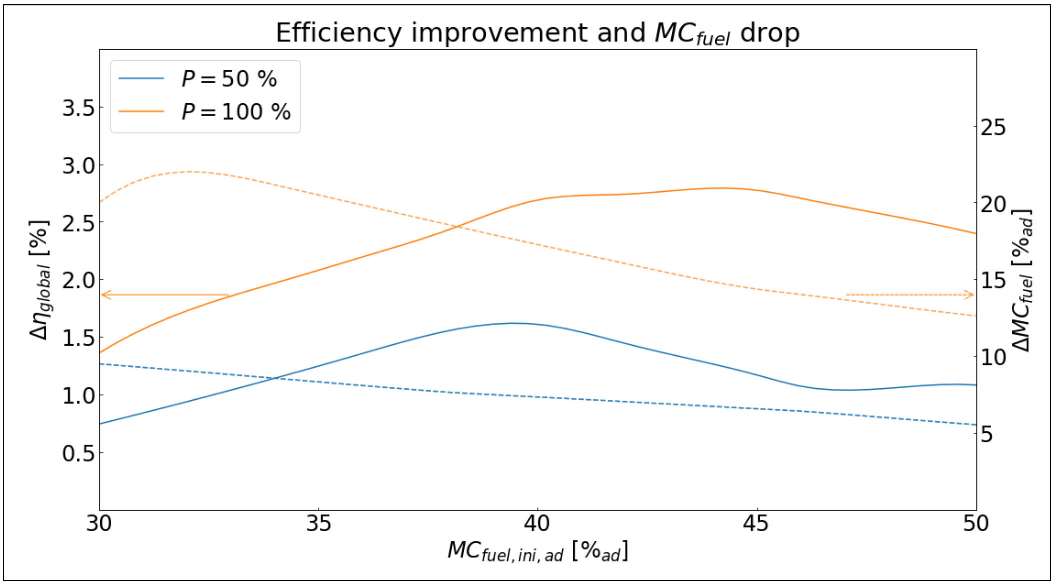


Figure 5.21 – Variation of the global efficiency of the system *Boiler* + *Dryer* and the fuel humidity drop at two FG share levels.

should be triple. In any case, above 10 % the fixed costs begin to be very high and the fuel savings do not compensate sufficiently for these costs to obtain an acceptable return time whatever the hours of operation.

Regarding the final economic benefits, these increase linearly as the fixed costs decrease. If 8000 hours of operation and the use of all the FG energy available ($P = 100$ %) are considered, a 50 % decrease in fixed costs down to 5 $\%_{CAPEX}$ leads to a 90 % increase in profits, i.e. almost double. This can be seen in figure 5.24. In addition, the model for estimating the cost of investment and its margin of error (±30 %) must be taken into account. Since fixed costs are calculated as a percentage of the capital cost, the uncertainty of the results will increase along with them. In case $P = 50$ %, this represents an uncertainty of $\approx \pm 95$ $k€$ when fixed costs are 5 $\%_{CAPEX}$ and increases to $\approx \pm 160$ $k€$ when fixed costs are 15 $\%_{CAPEX}$.

Concerning the realistic case, the results also improve linearly, as is to be expected. However, a reduction in fixed costs does not lead to a viable project either. In case $P = 100$ %, by reducing fixed costs to half of that of the base case, the results improve by about 60 % compared to the base case results.

5.4.4 Fuel Cost

Many studies indicate that fuel costs have a very important impact on the viability of this type of project. Increasing plant efficiency can lead to two scenarios. On the one hand, maintaining the flow of incoming fuel means a greater production of steam or useful energy. An increase in steam production means an increase in the electricity produced with the subsequent higher operating profit margin. On the other hand, the case considered in this porject of having a certain demand for steam or useful energy, the incoming fuel will be lower and, therefore, there will be a saving in fuel.

The first sign of this high sensitivity of viability to the price of fuel is reflected in the payback period. An increase of 10€$/MWh$ in the price with respect to the base case (+40 %) means that the project goes from not being viable even with a ≈ 100 % of boiler

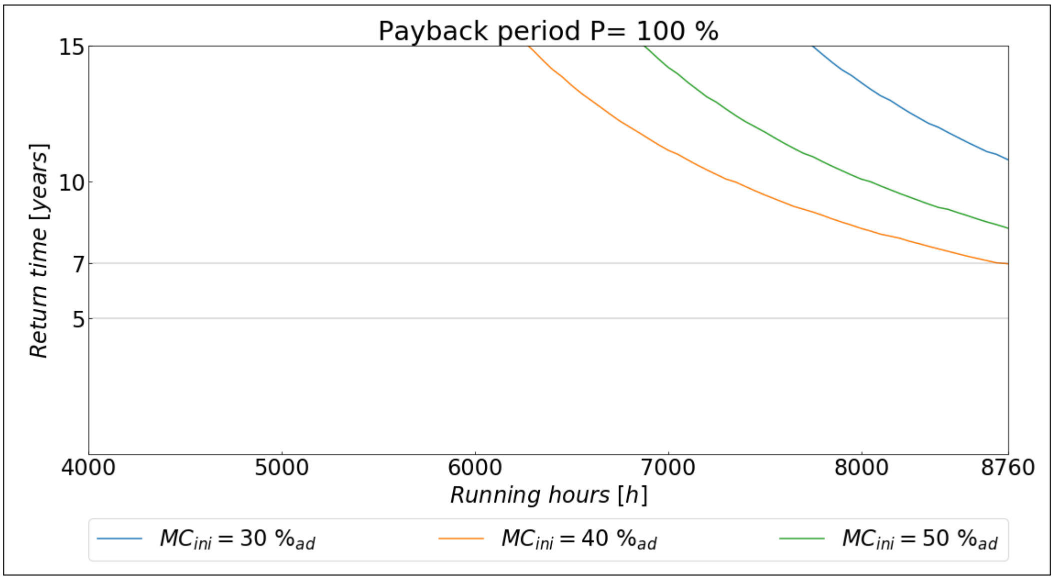


Figure 5.22 – Payback period of different initial fuel moistures if all the FG energy available is used, i.e., the largest dryer sizes

load factor ($\approx 8760\ h$) to obtaining a return period of less than 5 years from 7200 hours and less than 7 years from 6000 operation hours.

In terms of end-of-life savings, the benefits vary linearly as the price of fuel changes. With 8000 hours of operation and use of all the available FG energy ($P = 100\ \%$) is considered, the increase of 1 €$/MWh$ results in an increase of 45k€ of benefit. Thus, an increase of about 8 % in price (from 24 to 26 €$/MWh$) results in a $\approx$ 42 % growth in profit. An increase of $\approx$ 42 % (34 €$/MWh$) would doble profit, as shown in figure 5.26. Comparing these values with those obtained through the size sensitivity analysis, the benefits increase by 75 € for each MW that the plant is larger.

In the realistic case analyzed, the global load factor of the boiler is 49.3 %, which would correspond to approximately 4300 hours. Notice that, since efficiency varies according to boiler load, the global annual efficiency differs slightly from that obtained under nominal conditions (see figure 5.16). Therefore, the results at the same level of load will be slightly different.

According to the results of the return on investment period shown in the figure 5.25, there is no economically viable project for any of the fuel costs considered in this analysis. These results are also shown in figure 5.26, where the results improve linearly. In the most favourable case (34 €$/MWh$), a recovery of the investment could be achieved at the end of the project useful life if the use of FG energy were partial and considering the favourable margin of error of the capital costs.

In summary, since the higher operating cost of the plant owes largely to fuel consumption, the estimation of its price is key for the study of the project's viability. In more in-depth feasibility studies, an important aspect to be analysed in depth will be the fuel price and fluctuation over periods of months and years.

5.5 Conclusions

The most efficient way to produce thermal energy is through a combustion process. This, along with the global objective of reducing emissions from the final energy production,

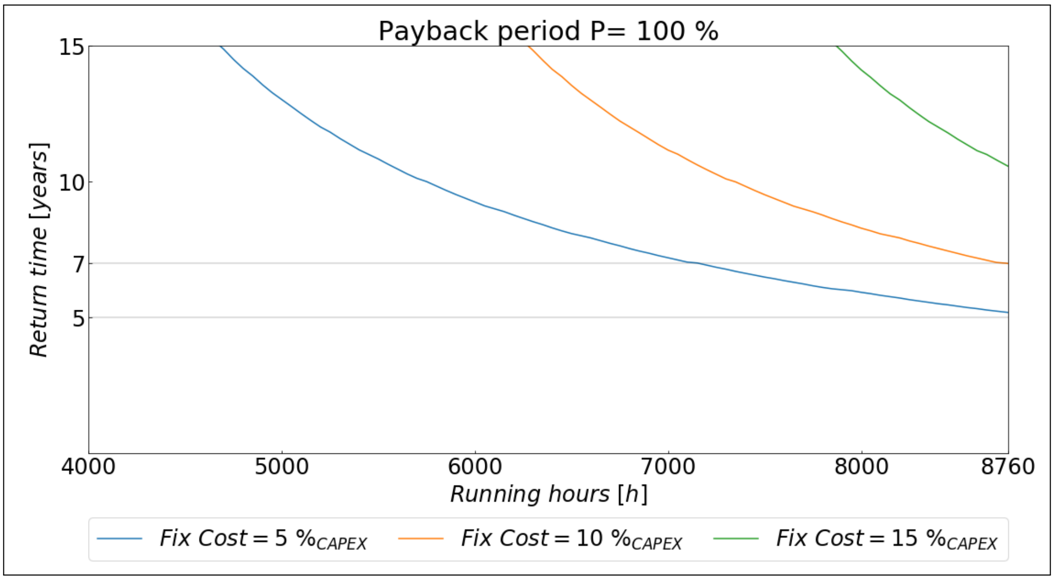


Figure 5.23 – Payback period of different fixed costs if all the FG energy available is used, i.e., the largest dryer sizes.

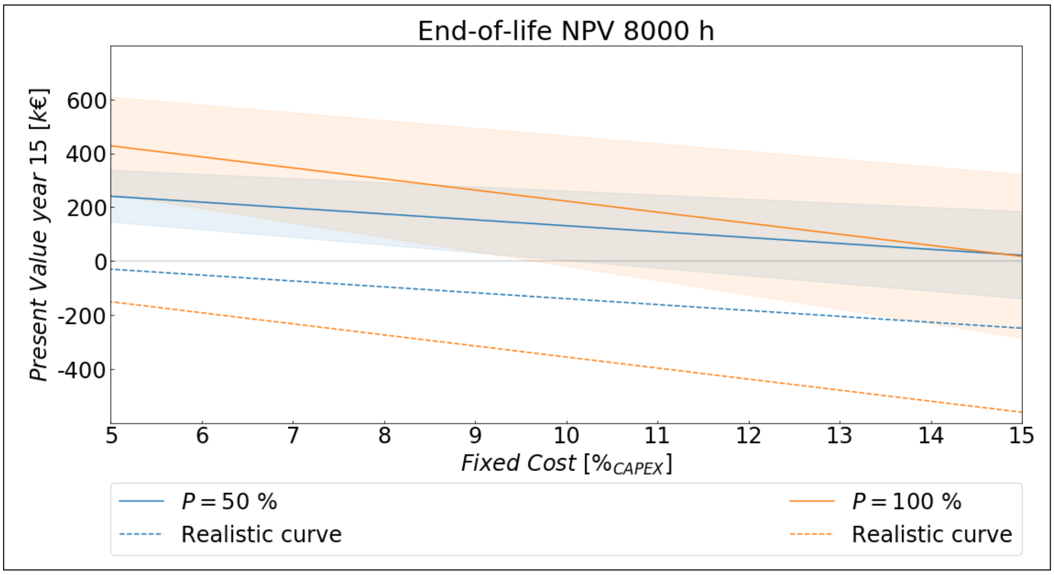


Figure 5.24 – NPV at the dryer end-of-life considering 8000 h operation at two FG shares as a function of the fixed cost considered. The dashed lines represents the realistic case. Note: the shaded area represents the margin of error for investment costs.

allows biomass a viable replacement for fossil fuels. In contrast to fossil fuels and coal, biomass has certain variations in composition and water content. The moisture can even vary depending on the season. In addition, biomass combustion plants generally aim to use the sources of biomass located near the plant, in order to avoid transport costs and to take advantage of waste or residues from, for example, the wood or husk food industry.

The moisture contained in the fuel results in a significant decrease in the density of useful energy available in the fuel. In addition, this water content can lead to several problems in the operation and management of the plant. The different boiler technologies have humidity limits for their operation. These constraints may be due either to from the capacity of the auto-thermal and self-supporting combustion or to from the maximum temperature that can be resisted by the boiler materials, among other factors.

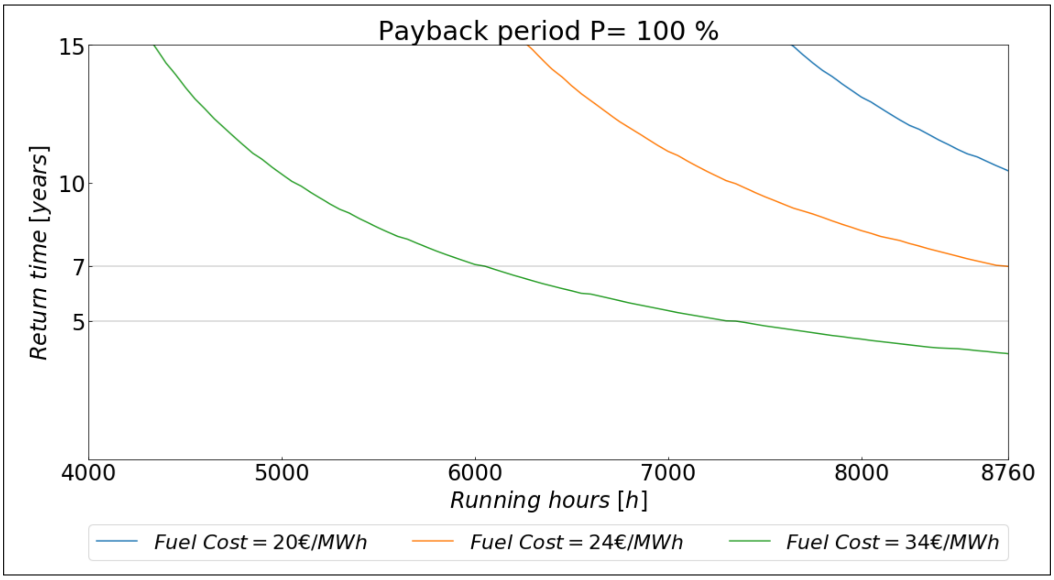


Figure 5.25 – Payback period of different fuel prices if all the FG energy available is used, i.e., the largest dryer sizes.

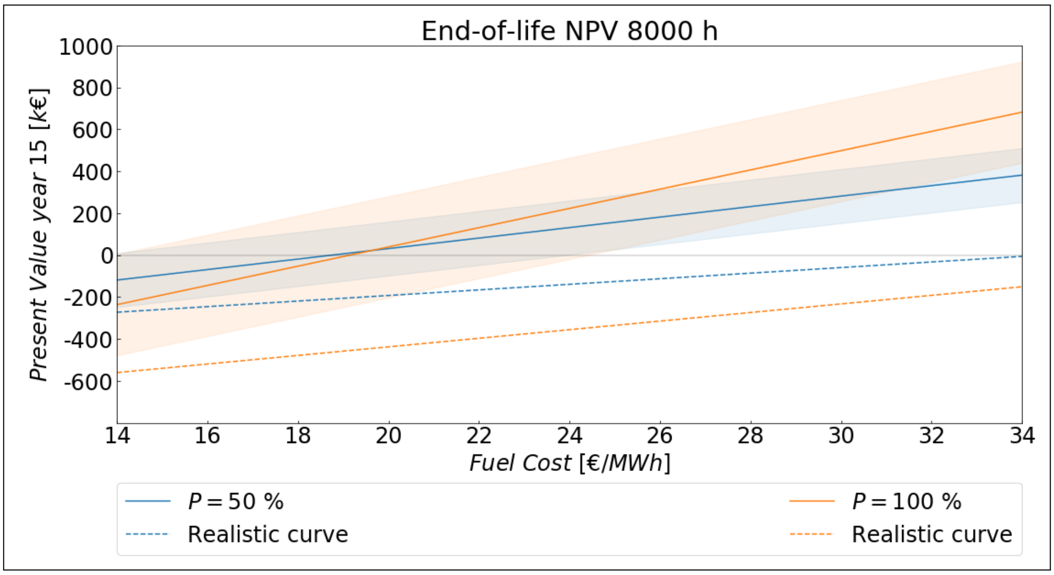


Figure 5.26 – NPV at the dryer end-of-life considering 8000 h operation at two FG shares as a function of the fuel cost considered. The dashed lines represents the realistic case. Note: the shaded area represents the margin of error for investment costs.

There are different techniques and methods to improve the efficiency of biomass boilers. Pre-drying of the fuel or condensation of the flue gases are some of the most studied technologies in order to improve the efficiency of energy production.

For biomass drying, the drying agents commonly used are preheated air, flue gas or superheated steam. The main advantage of using flue gases is their recovery for direct use since it is considered as a waste energy or secondary energy from an industrial process or a boiler. The use of pre-heated air implies the need to heat it. Although the heat source could be another energy waste, it would require the installation of an exchange, which would increase the investment cost. Superheated steam would also be of interest as a drying agent if it was not produced directly for the purpose of drying, but results from an industrial process or produced by a secondary energy source.

In this work, the techno-economical feasibility of installing a dryer in a medium-scale district heat plant has been studied. As a drying agent, the boiler flue gas itself has been considered. The dryer is of the belt-type. The limitation of the available energy for drying and its variation as a function of the boiler operation determine the final achievable humidity. The partial use of FG has also been studied to analyse the possible existence of an economically optimal intermediate humidity value. This partial use would mean lower annual profits, but lower investment costs which may also mean earlier return on investment.

Since this project considers a district heat plant, the economic feasibility has been studied in the form of fuel savings. The minimum number of hours after which the project is profitable has been studied. The high investment costs coupled with high fixed costs leads to a highly questionable viability of the project even considering high hours of operation due to the excessive return on investment time. At this high running hours the fuel saved is higher, so the fixed costs of the dryer are covered.

The analysis carried out requires a broad interpretation of the results. Aspects such as the size of the plant, the associated fixed costs, the price of fuel or the initial humidity of the fuel are determining factors for the viability of the project. In addition, one of the conclusion from this analysis is that other alternatives for improving efficiency utilizing the flue gas produced should also be investigated, such as the flue gas condenser.

The model proposed for the analysis has certain shortcomings described in the corresponding chapters. Some of them are the flue gas temperature curve as a function of the load, the modelled excess air or the estimation of the fuel temperature at the dryer outlet.

These values have been estimated through rules of thumb or experimental results from other works. The viability studied is based on fuel savings. Since this saving is produced as a consequence of the increase in efficiency, this is a key variable to estimate. As previously mentioned, in the model developed the efficiency is obtained from indirect analysis, i.e. from the calculation of losses, which are mostly the heat losses embedded in the flue gas. Therefore, a correct estimate of the excess oxygen needed for each level of load and humidity of the fuel entering the boiler is important as well as the temperature of this FG.

The boiler model based on the indirect calculation of the efficiency allows to introduce a humidity level, FG temperature and load as inputs to the program. Most of these values, in addition to the excess oxygen curves, have been estimated. However, the model allows the introduction of real data of flue gas temperature, humidity, etc.. In the present project, a realistic demand curve has been considered. In future works, it would be interesting to introduce experimental values such as FG temperature or a more realistic oxygen content curves.

Regarding the dryer model, the simple mass and energy balance is relatively accurate in techno-economical analysis. As mentioned, the fixed relative humidity does not have a major influence on high RH ranges. This is because high flue gas inlet temperatures also mean relatively high flue gas temperatures at the dryer outlet. The rest of the assumptions have also been estimated from previous studies. One of the major weakness of the model lies in the estimation of the flue gas velocity through the bed, since this value is not linked in the model to any thermodynamic variable. However, the assumptions made in this project are widely used and verified by both the literature and the experimental results.

In the same way as with the boiler, it would be interesting to obtain real values in order to carry out a more accurate study and to avoid assumptions such as the linear correlation between the temperature of the fuel and the moisture contents at the dryer inlet and outlet.

The Investment cost model developed has a margin of error which, together with the high fixed costs estimated, leads to considerable uncertainty. Furthermore, the investment cost models used, compared to others, do not consider the existence of a decrease in the unit cost of area when the area increases. This means that there is not benefit for build larger. However, this developed model can be expected to be more accurate than those studied in the literature. This can be affirmed since the surface ranges for which the models studied in the literature are very large and, in general, are used for the calculation of the investment cost of larger dryers.

In summary, the techno-economic analysis carried out shows that the viability of the project only may occurs at high running hours. Variables such as fixed costs, fuel price or plant size can considerably influence the feasibility of the project. Given the significant sensitivity of these variables, it is important to notice that this base case represents a district heating plant under certain conditions and assumptions. However, it provides an overview of the economic feasibility of implementing a dryer. Therefore, in a more focused analysis in a specific case, values such as excess oxygen curves set in the boiler, fixed dryer costs or fuel price must be carefully determined.

With the described plant layout (one biomass boiler and ancillary systems) it is not possible to achieve these high values of equivalent hours or annual load factors. This is due to the drastic drop in demand in the summer months. During these months, demand may not even reach the minimum operating capacity of the biomass boiler.

There are aspects involved in the drying of biomass that have not been assessed in this project and that could lead to a more favourable business case. Some examples are the recovery of saturated FG, smaller secondary measures, lower CO_2 and pollutant emissions, lower load on the boiler feed system (in case of a retrofitting), etc. Other negative aspects could also be studied in terms of additional costs such as the treatment and management of the water extracted, condensation of pollutants or the greater complexity of the system.

The sensitivity analysis carried out shows that the viability of the project is significantly affected by the size of the plant and the fuel price, as well as fixed costs. The most sensitive parameter of the project is the fuel price, since it represents the highest operating cost. Concerning to the plant size, notice that the sensitivity will also be conditioned by the price of biomass fuel.

This analysis considers the running hours as a variable, unlike other studies found in the literature. The minimum operating time required to obtain a profitable investment has been studied in terms of Net Present Value (NPV) and payback period. The results obtained show that a high number of operating hours is required to achieve acceptable profitability, even under the most favourable fixed cost and fuel price conditions and for the largest plant sizes analysed.

In the realistic case analyzed, a low number of operating hours is observed, above 6000 total hours and about 4300 equivalent hours in nominal conditions (full load). Despite covering more than 84 % of the network demand, the biomass boiler load factor, 49.3 %, is low. Other more advanced configurations of the hot water production system should be analysed, such as two biomass boilers with different nominal heat powers, besides the ancillary systems. This would result in a higher demand covered by the biomass boilers and a longer operation time at high load rates.

Bibliography

[1] Eurostat. *Complete Energy Balance.* URL: `https://appsso.eurostat.ec.europa.eu/nui/submitViewTableAction.do` (cit. on pp. 1, 2).

[2] Cristina Calderon and Martin Colla. *Biomass for Heat.* Statistical report. Bioenergy Europe. Bioenergy Europe, 2019 (cit. on pp. 1, 3).

[3] Cristina Calderon and Martin Colla. *Bioelectricity.* Statistical report. Bioenergy Europe. Bioenergy Europe, 2019 (cit. on p. 1).

[4] Thierry Lecomte et al. *Best Available Techniques (BAT) Reference Document for Large Combustion Plants.* Science for Policy report. Joint Research Centre, 2017 (cit. on pp. 1, 2, 12).

[5] Kevin Sartor and Pierre Dewallef. "Integration of heat storage system into district heating networks fed by a biomass CHP plant". In: *Journal of Energy Storage* 15 (2018), pp. 350 –358. ISSN: 2352-152X. DOI: `https://doi.org/10.1016/j.est.2017.12.010` (cit. on p. 3).

[6] ENGIE. *Our Biomass Power Plants.* 2019. URL: `https://corporate.engie-electrabel.be/local-player/biomass/` (cit. on p. 3).

[7] GALINDO MARINA et al. *Efficient district heating and cooling markets in the EU: Case studies analysis, replicable key success factors and potential policy implications.* Publications Office of the European Union, 2016. ISBN: 978-92-79-65048-2. DOI: `10.2760/371045` (cit. on p. 3).

[8] Cristina Calderon and Martin Colla. *Biomass Supply.* Statistical report. Bioenergy Europe. Bioenergy Europe, 2019 (cit. on pp. 3–5).

[9] IEA Bioenergy. *Fossil vs biogenic CO2 emissions.* IEA Bioenergy. URL: `https://www.ieabioenergy.com/iea-publications/faq/woodybiomass/biogenic-co2/` (cit. on p. 4).

[10] Harmanpreet Singh. "Study of Emission Characteristics and Noise of Dual Fuel Engine Run on Blends of Diesel and Producer Gas from Biomass Materials". PhD thesis. Aug. 2016 (cit. on pp. 5, 6).

[11] Shang-Tian Yang. "Chapter 1 - Bioprocessing – from Biotechnology to Biorefinery". In: *Bioprocessing for Value-Added Products from Renewable Resources.* Ed. by Shang-Tian Yang. Amsterdam: Elsevier, 2007, pp. 1 –24. ISBN: 978-0-444-52114-9. DOI: `https://doi.org/10.1016/B978-044452114-9/50002-5` (cit. on p. 7).

[12] Panagiotis Grammelis, Nikolaos Margaritis, and Emmanouil Karampinis. "2 - Solid fuel types for energy generation: Coal and fossil carbon-derivative solid fuels". In: *Fuel Flexible Energy Generation.* Ed. by John Oakey. Boston: Woodhead Publishing, 2016, pp. 29 –58. ISBN: 978-1-78242-378-2. DOI: `https://doi.org/10.1016/B978-1-78242-378-2.00002-X` (cit. on p. 7).

[13] Julien Blondeau. *Biomass introduction. Properties of biomass.* Renewable Enegy Technology course (MECA-H414-0-201920),Brussels Faculty of Engineering (BRUFACE). ULB and VUB, 2019. Brussels: Brussels Faculty of Engineering (BRUFACE). ULB and VUB, 2019 (cit. on p. 8).

[14] A. Robinson et al. “Combustion characteristics of biomass fuels”. In: Jan. 1998 (cit. on pp. 7, 8, 20).

[15] B.K. Behera and A. Varma. *Bioenergy for Sustainability and Security*. Springer International Publishing, 2018. ISBN: 9783319965383 (cit. on p. 8).

[16] A. Sharma, A. Shukla, and L. Aye. *Low Carbon Energy Supply: Trends, Technology, Management*. Green Energy and Technology. Springer Singapore, 2018. ISBN: 9789811073267 (cit. on p. 8).

[17] A. Dufour. *Thermochemical Conversion of Biomass for the Production of Energy and Chemicals*. Wiley, 2016. ISBN: 9781119137702 (cit. on p. 8).

[18] Chun-Yang Yin. “Prediction of higher heating values of biomass from proximate and ultimate analyses”. In: *Fuel* 90.3 (2011), pp. 1128 –1132. ISSN: 0016-2361. DOI: `https://doi.org/10.1016/j.fuel.2010.11.031` (cit. on p. 8).

[19] P. Basu. *Biomass Gasification and Pyrolysis: Practical Design and Theory*. Elsevier Science, 2010. ISBN: 9780080961620 (cit. on p. 9).

[20] Lucio De Fusco. “Feasibility of drying systems for small and medium size biomass combustion plants”. Politecnico di Torino. Master’s Thesis. Politecnico di Torino (cit. on pp. 9–11, 15, 19, 20, 23, 32, 39, 43–46, 48, 52, 58).

[21] Forest Products Laboratory. *Wood handbook - Wood as an engineering material*. General Technical Report FPL-GTR-190. U.S.Department of Agriculture, Forest Service, Forest Products Laboratory, 2010 (cit. on p. 9).

[22] J.F. RICHARDSON, J.H. HARKER, and J.R. BACKHURST. “CHAPTER 16 - Drying”. In: *Chemical Engineering (Fifth Edition)*. Ed. by J.F. RICHARDSON, J.H. HARKER, and J.R. BACKHURST. Fifth Edition. Chemical Engineering Series. Oxford: Butterworth-Heinemann, 2002, pp. 901 –969. ISBN: 978-0-08-049064-9. DOI: `https://doi.org/10.1016/B978-0-08-049064-9.50027-8` (cit. on p. 9).

[23] Sjaak van loo and Jaap Koppejan. *The Handbook of Biomass Combustion and Cofiring*. Jan. 2008, pp. – (cit. on pp. 10–13, 20, 32).

[24] B.M Jenkins et al. “Combustion properties of biomass”. In: *Fuel Processing Technology* 54.1 (1998), pp. 17 –46. ISSN: 0378-3820. DOI: `https://doi.org/10.1016/S0378-3820(97)00059-3` (cit. on p. 10).

[25] Ayhan Demirbas. “Combustion characteristics of different biomass fuels”. In: *Progress in Energy and Combustion Science* 30.2 (2004), pp. 219 –230. ISSN: 0360-1285. DOI: `https://doi.org/10.1016/j.pecs.2003.10.004` (cit. on pp. 10, 20).

[26] Ingwald Obernberger and Friedrich Biedermann. “Biomass Energybiomassenergy-Heat Provision in Modern Large-Scale Systemsbiomassenergyheat provision”. In: *Renewable Energy Systems*. Ed. by Martin Kaltschmitt et al. New York, NY: Springer New York, 2013, pp. 343–381. ISBN: 978-1-4614-5820-3. DOI: `10.1007/978-1-4614-5820-3_316` (cit. on pp. 13, 14).

[27] *PULVERIZED COAL FURNACES*. URL: `http://thermopedia.com/content/1066/` (cit. on p. 15).

[28] W A Amos. “Report on Biomass Drying Technology”. In: (). DOI: `10.2172/9548` (cit. on pp. 15, 17, 18).

[29] A.S. Mujumdar. *Handbook of Industrial Drying*. Advances in Drying Science and Technology. CRC Press, 2014. ISBN: 9781466596665 (cit. on pp. 15, 17, 39).

[30] Henrik Holmberg and Pekka Ahtila. “Comparison of drying costs in biofuel drying between multi- stage and single-stage drying”. In: *Biomass and Bioenergy* 26.6 (2004), pp. 515 –530. ISSN: 0961-9534. DOI: `https://doi.org/10.1016/j.biombioe.2003.09.007` (cit. on pp. 15, 24, 25, 27–29, 39, 41, 42, 45, 46, 48, 49, 51).

[31] Hanning Li et al. “Evaluation of a biomass drying process using waste heat from process industries: A case study”. In: *Applied Thermal Engineering* 35 (2012), pp. 71 –80. ISSN: 1359-4311. DOI: `https://doi.org/10.1016/j.applthermaleng.2011.10.009` (cit. on pp. 15, 17–20, 23, 25–29, 38–40, 48, 50, 51, 62).

[32] Henrik Holmberg. “Biufuel drying as a concept to improve the energy efficiency of an industrial CHP plant”. Doctoral Dissertation. Helsinki University of Technology. ISBN: 978-951-22-8649-2 (cit. on pp. 15, 48).

[33] Pauline M. Doran. “Chapter 11 - Unit Operations”. In: *Bioprocess Engineering Principles (Second Edition)*. Ed. by Pauline M. Doran. Second Edition. London: Academic Press, 2013, pp. 445 –595. ISBN: 978-0-12-220851-5. DOI: `https://doi.org/10.1016/B978-0-12-220851-5.00011-3` (cit. on pp. 15, 42).

[34] Barry Fox, Giovanni Bellini, and Laura Pellegrini. “Chapter 14 - Drying”. In: *Fermentation and Biochemical Engineering Handbook (Third Edition)*. Ed. by Henry C. Vogel and Celeste M. Todaro. Third Edition. Boston: William Andrew Publishing, 2014, pp. 283 –305. ISBN: 978-1-4557-2553-3. DOI: `https://doi.org/10.1016/B978-1-4557-2553-3.00014-3` (cit. on pp. 15, 16).

[35] Munna Verma et al. “Drying of biomass for utilising in co-firing with coal and its impact on environment – A review”. In: *Renewable and Sustainable Energy Reviews* 71 (2017), pp. 732 –741. ISSN: 1364-0321. DOI: `https://doi.org/10.1016/j.rser.2016.12.101` (cit. on pp. 17, 19, 20).

[36] J.G. Brammer and A.V. Bridgwater. “The influence of feedstock drying on the performance and economics of a biomass gasifier–engine CHP system”. In: *Biomass and Bioenergy* 22.4 (2002), pp. 271 –281. ISSN: 0961-9534. DOI: `https://doi.org/10.1016/S0961-9534(02)00003-X` (cit. on pp. 19, 27, 48).

[37] Hans Hartmann et al. *Future Bio Tec Low emission operation manual for chimney stove users*. Oct. 2012 (cit. on p. 20).

[38] Lucio De Fusco, H. Jeanmart, and J. “A modelling approach for the assessment of an air-dryer economic feasibility for small-scale biomass steam boilers”. In: *Fuel Processing Technology* 134 (2015), pp. 251 –258. ISSN: 0378-3820. DOI: `https://doi.org/10.1016/j.fuproc.2015.01.043` (cit. on pp. 22, 25–29, 39, 41, 42, 45, 48, 49, 51, 58, 62).

[39] Tiina Myllymaa, Henrik Holmberg, and Pekka Ahtila. “Techno-economic evaluation of biomass drying in moving beds: The effect of drying kinetics on drying costs”. In: *Drying Technology* 37.10 (2019), pp. 1201–1214. DOI: `10.1080/07373937.2018.1492615`. eprint: `https://doi.org/10.1080/07373937.2018.1492615` (cit. on pp. 24–28, 38, 47–49, 52, 62).

[40] Henrik Holmberg and Pekka Ahtila. “Evaluation of energy efficiency in biofuel drying by means of energy and exergy analyses”. In: *Applied Thermal Engineering* 25.17 (2005), pp. 3115 –3128. ISSN: 1359-4311. DOI: `https://doi.org/10.1016/j.applthermaleng.2005.04.005` (cit. on pp. 24–26, 39, 41, 42).

[41] Henrik Holmberg and Pekka Ahtila. “DRYING PHENOMENON IN A FIXED BED UNDER THE BIO FUEL MULTI STAGE DRYING”. In: (Jan. 2002) (cit. on pp. 26, 42, 48).

[42] P. Perre, M. Moser, and M. Martin. “Advances in transport phenomena during convective drying with superheated steam and moist air”. In: *International Journal of Heat and Mass Transfer* 36.11 (1993), pp. 2725 –2746. ISSN: 0017-9310. DOI: `https://doi.org/10.1016/0017-9310(93)90093-L` (cit. on pp. 26, 42).

[43] Tiina Myllymaa et al. “Wood Chip Drying in Fixed Beds: Drying Kinetics and Economics of Drying at a Municipal Combined Heat and Power Plant Site”. In: *Drying Technology* 33.2 (2015), pp. 205–215. DOI: `10.1080/07373937.2014.945179`. eprint: `https://doi.org/10.1080/07373937.2014.945179` (cit. on p. 26).

[44] Tiina Myllymaa et al. “Upgrading of soot sludge from waste to fuel by means of low temperature drying in fixed beds”. In: *Fuel Processing Technology* 138 (2015), pp. 509 –519. ISSN: 0378-3820. DOI: `https://doi.org/10.1016/j.fuproc.2015.06.030` (cit. on p. 26).

[45] Henrik Holmberg, Pekka Ahtila, and Olli Ahtila. “Experimental Study on Drying of Bark in Fixed Beds”. In: *Drying Technology* 29.8 (2011), pp. 953–960. DOI: `10.1080/07373937.2010.551305`. eprint: `https://doi.org/10.1080/07373937.2010.551305` (cit. on p. 26).

[46] Henrik Holmberg and Pekka Ahtila. “Optimization of the bark drying process in combined heat and power production of pulp and paper mill”. In: 2005 (cit. on p. 27).

[47] Christian Fyhr and Anders Rasmuson. “Some aspects of the modelling of wood chips drying in superheated steam”. In: *International Journal of Heat and Mass Transfer* 40.12 (1997), pp. 2825 –2842. ISSN: 0017-9310. DOI: `https://doi.org/10.1016/S0017-9310(96)00340-7` (cit. on p. 27).

[48] David J Brennan. *Process Industry Economics. An International Perspective.* English. Institution of Chemical Engineers, 1998. ISBN: 0 85295 391 7 (cit. on pp. 27, 29, 46).

[49] Thibault Coppieters and Julien Blondeau. “Techno-Economic Design of Flue Gas Condensers for Medium-Scale Biomass Combustion Plants: Impact of Heat Demand and Return Temperature Variations”. In: *Energies* 12 (June 2019), p. 2337. DOI: `10.3390/en12122337` (cit. on pp. 30, 32–34, 37).

[50] K. Sartor, S. Quoilin, and P. Dewallef. “Simulation and optimization of a CHP biomass plant and district heating network”. In: *Applied Energy* 130 (2014), pp. 474 –483. ISSN: 0306-2619. DOI: `https://doi.org/10.1016/j.apenergy.2014.01.097` (cit. on p. 32).

[51] J Good et al. *Determination of the Efficiencies of Automatic Biomass Combustion Plants: Evaluation of Different Methods for Efficiency Determination and Comparison of Efficiency and Emissions for Different Operations Modes.* Technical Report. Internation Energy Agency: Paris, France, 2006 (cit. on pp. 35–37).

[52] Changdong Sheng and J.L.T. Azevedo. “Estimating the higher heating value of biomass fuels from basic analysis data”. In: *Biomass and Bioenergy* 28.5 (2005), pp. 499 –507. ISSN: 0961-9534. DOI: `https://doi.org/10.1016/j.biombioe.2004.11.008` (cit. on p. 35).

[53] Floriane Mermoud et al. “Impact of load variations on wood boiler efficiency and emissions: in-situ monitoring of two boilers (2 MW and 0.65 MW) supplying a district heating system”. eng. In: *Archives des Sciences* 68 (2015). ID: unige:83435, pp. 27–38 (cit. on pp. 36, 37).

[54] Anthony Haroutunian et al. *Audit’bois : Analyse énergétique, environnementale et économique du chauffage à distance au bois à Genève : Retour d’expérience sur l’installation de Cartigny.* fre. ID: unige:32223. 2013 (cit. on pp. 36, 37).

[55] Murlidhar Gupta, Jin Yang, and Christian Roy. “Specific Heat and Thermal Conductivity of Softwood Bark and Softwood Char Particles”. In: *Fuel* 82 (May 2003), pp. 919–927. DOI: `10.1016/S0016-2361(02)00398-8` (cit. on p. 41).

[56] S. Pang and Q. Xu. “Drying of Woody Biomass for Bioenergy Using Packed Moving Bed Dryer: Mathematical Modeling and Optimization”. In: *Drying Technology* 28.5 (2010), pp. 702–709. DOI: `10.1080/07373931003799251`. eprint: `https://doi.org/10.1080/07373931003799251` (cit. on p. 42).

[57] P. Basu. *Biomass Gasification, Pyrolysis and Torrefaction: Practical Design and Theory.* Elsevier Science, 2013. ISBN: 9780123965431 (cit. on p. 43).

[58] Henrik Holmberg and Pekka Ahtila. “Simulation Model for the Model-Based Control of a Biofuel Dryer at an Industrial Combined Heat and Power Plant”. In: *Drying Technology* 24 (Dec. 2006), pp. 1547–1557. DOI: `10.1080/07373930601030747` (cit. on pp. 45, 48).

[59] D.R. Woods. *Rules of Thumb in Engineering Practice.* Wiley, 2007. ISBN: 9783527611126 (cit. on pp. 46–48, 60).

[60] Chemical Engineering. *Pant Cost Index.* URL: `https://www.chemengonline.com/site/plant-cost-index/` (cit. on pp. 47, 48).

[61] W.M. Vatavuk. “Updating the CE plant cost index”. In: *Chemical Engineering* 109 (Jan. 2002), pp. 62–70 (cit. on p. 47).

[62] Eurostat. *Electricity prices for non-household consumers - bi-annual data (from 2007 onwards).* URL: `https://appsso.eurostat.ec.europa.eu/nui/submitViewTableAction.do` (cit. on pp. 49, 56).

Appendix A

ResultClass definitions

Description of the variables defined for the analysis in the class "Result" for the analysis of the economic viability and the thermal results. The code is shwon in the annex B.8.

Class	Instance / SubClass	Instance of SubClass	Description	index/units	type	Size/Shape
Results	P_Boiler		Nominal rate of the Boiler	[W]	float	-
	Mcini_ad		Humidity of Fuel initial (Without Dryer)	[%ad]	float	-
	FuelC		Fuel Cost	[€/MWh]	float	-
	FixC		Fix Cost of O&M cost	[%ofCAPEX]	float	-
	T_FG_boiler_nom		Temperature of the FG at the outlet of the Boiler	[K]	float	-
	T_FG_dryer_nom		Temperature of the FG or FG and Air mix at inlet of the Dryer	[K]	float	-
	PartialFG		Array of Partial FG. Share of the FG introduced into the Dryer	[-]	array	len(PartialFG)
	RhoFG		Average IN-OUT density of the FG in the Dryer for each PartialFG	[kg/m3]	array	len(PartialFG)
	Ac		Conveyor surface for each PartialFG	[m2]	array	len(PartialFG)
	Tau		Array of running time. Quantity of hours	[h/year]	array	len(Tau)
	Hours		Array that contains the quantity of hours that a value of the Demand Curve takes place	[h]	array	len(Hours)
	Load		Demand Curve	[W]	array	len(Hours)
	P_BoilerCurve		Heat provided by the Boiler with respect to the Demand Curve	[W]	array	len(Hours)
	P_BoilerFactor		Load Factor of the Boiler with respect to the Demand Curve	[-]	array	len(Hours)
	FullLoad	PartialFG	Array of Partial FG. Share of the FG introduced into the Dryer	[-]	array	len(PartialFG)
		Tau	Array of running time. Quantity of hours	[h/year]	array	len(Tau)
		T_FG_boiler	FG temperature at the outlet of the boiler	[K]	float	-
			Initial Conditions. Without Dryer			
		MassFuelini_daf	Mass flow of Fuel in initial conditions, dry ash free basis	[kgFueldaf/s]	float	-
		Ashini_ad	Ash content per kg of total fuel , initial conditions	[(kgAsh/kgFuelAd)*100], [%]	float	-
		MassFuelini_ad	Mass flow of Fuel in initial conditions, As delivered basis	[kgFuelad/s]	float	-
		Effini	Boiler efficiency in initial conditions	[-]	float	-
		LHVini_daf	LHV of Fuel in initial conditions, dry ash free basis	[J/kgFueldaf]	float	-
		LHVini_ad	LHV of Fuel in initial conditions, As delivered basis	[J/kgFuelad]	float	-
		m_air_esp_daf_ini	Amount of air introduced in the boiler per kg of Fuel dry ash free basis, initial conditions	[kgAir/kgFueldaf]	float	-
		m_air_esp_ad_ini	Amount of air introduced in the boiler per kg of Fuel As delivered basis, initial conditions	[kgAir/kgFueldad]	float	-
		MC_FGini_wb	Humidity of FG at the outlet of the Boiler, per kg of FG total, initial condtions	[kgH2O/kgFGwb]	float	-
		MC_FGini_db	Humidity of FG at the outlet of the Boiler, per kg of FG dry basis, initial conditions	[kgH2O/kgFGdb]	float	-
			Drying contidions			
		MCfin_ad	Humidity of Fuel at the inlet of the Boiler after being dryed by the PartialFG	[%ad]	array	len(PartialFG)
		Effi	Boiler efficiency with Fuel dried	[-]	array	len(PartialFG)
		MassFuel_daf	Mass flow of Fuel dried, dry ash free basis	[kgFueldaf/s]	array	len(PartialFG)
		MassFuel_ad	Mass flow of Fuel dried, As delivered basis, at the inlet of the system Dryer+Boiler	[kgFuelad/s]	array	len(PartialFG)
		MassFG_Boiler	Mass flow of Flue Gas at the outlet of the Boiler, dry basis	[kgdb/s]	array	len(PartialFG)
		MassAir	Mass flow of Air that is mixed with the FG from the boiler, dry basis	[kgdb/s]	array	len(PartialFG)
		MassFG_Mixer	Mass flow of Flue Gas at the inlet of the Mixer, dry basis	[kgdb/s]	array	len(PartialFG)
		MassFG_Dryer	Total Mass flow FG+Air at the inlet of the Dryer, dry basis	[kgdb/s]	array	len(PartialFG)
		Ash_Boiler_ad	Ash content per kg of total fuel at the inlet of the Boiler	[(kgAsh/kgFuelAd)*100], [%]	array	len(PartialFG)
		MassFuel_Boiler_ad	Mass flow of Fuel dried, As delivered basis, at the inlet of Boiler	[kgFuelad/s]	array	len(PartialFG)
		LHV_daf	LHV of Fuel, dry ash free basis, at the inlet of the Boiler	[J/kgFueldaf]	array	len(PartialFG)
		LHV_ad	LHV of Fuel, Ash delivered basis, at the inlet of the Boiler	[J/kgFuelad]	array	len(PartialFG)
		m_air_esp_daf	Amount of air introduced in the boiler per kg of Fuel dry ash free basis	[kgAir/kgFueldaf]	array	len(PartialFG)
		m_air_esp_ad	Amount of air introduced in the boiler per kg of Fuel at the inlet of the Boiler As delivered basis	[kgAir/kgFueldad]	array	len(PartialFG)
		ExcessAir	Amoung of Air in excess in the Boiler	[mol/molFuel]	array	len(PartialFG)
		MC_FG_boiler_wb	Humidity of FG at the outlet of the Boiler/Inlet of the Mixer, per kg of FG total	[kgH2O/kgFGwb]	array	len(PartialFG)
		MC_FG_boiler_db	Humidity of FG at the outlet of the Boiler/Inlet of the Mixer, per kg of FG dry basis	[kgH2O/kgFGdb]	array	len(PartialFG)
		MC_FG_dryer_wb	Humidity of FG at the inlet of the Dryer, per kg of FG total	[kgH2O/kgFGwb]	array	len(PartialFG)

Class	Instance / SubClass	Instance of SubClass	Description	index/units	type	Size/Shape
	FullLoad	MC_FG_dryer_db	Humidity of FG at the the inlet of the Dryer, per kg of FG dry basis	[kgH2O/kgFGdb]	array	len(PartialFG)
		EffiGlobal	Global efficiency of the system Dryer+Boiler	[-]	array	len(PartialFG)
		me	Evaporation rate of water in the dryer	[kgH2O/s]	array	len(PartialFG)
		HeatExc	Heat exchanged in the dryer	[kW]	array	len(PartialFG)
		Drying_Agent	Amount of Mix FG+Air in the Dryer per amount of Fuel dried, dry ash free basis	[kgMix/kGFueldaf]	array	len(PartialFG)
		Drying_Agent_FG	Amount of FG in the Mixer per amount of Fuel dried, dry ash free basis	[kgFG/kGFueldaf]	array	len(PartialFG)
		Heat_Consum	Heat exchanged per amount of water evaporated	[kJ/kgH2O]	array	len(PartialFG)
		Heat_Consum_e	Heat exchanged per amount of Fuel dried, dry ash free basis	[kJ/kgFueldaf]	array	len(PartialFG)
		TFGout	Temperature at the outlet of the Dryer	[K]	array	len(PartialFG)
		LF	Load Factor of the Boiler	[-]	array	len(Tau)
		DemCov	Demand covered by the Boiler. By default, in FullLoad Case, =1	[-]	float	-
		Years	lifespan of the project	[years]	int	-
		NPVSolid	Net Present Value of Solid Band model	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPVSolid_max	Net Present Value of Solid Band model if CAPEX is 30% lower	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPVSolid_min	Net Present Value of Solid Band model if CAPEX is 30% higher	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPVBelt	Net Present Value of Belt Band model	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPVBelt_max	Net Present Value of Belt Band model if CAPEX is 30% lower	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPVBelt_min	Net Present Value of Belt Band model if CAPEX is 30% higher	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPVHolmber	Net Present Value of Holmberg model	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPVBrammer	Net Present Value of Brammer model	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPV	Net Present Value, Belt and Solid Average	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPV_max	Net Present Value, Belt and Solid Average if CAPEX is 30% lower	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		NPV_min	Net Present Value, Belt and Solid Average if CAPEX is 30% higher	[k€]	matrix 3D	[len(Tau),years,len(PartialFG)]
		RTHolmberg	Return time of investment, Holmberg model	[years]	matrix	[len(Tau),len(PartialFG)]
		RTBrammer	Return time of investment, Brammer model	[years]	matrix	[len(Tau),len(PartialFG)]
		RTSolid	Return time of investment, Solid Band model	[years]	matrix	[len(Tau),len(PartialFG)]
		RTBelt	Return time of investment, Belt Band model	[years]	matrix	[len(Tau),len(PartialFG)]
		RT	Return time of investment, Belt and Solid Average	[years]	matrix	[len(Tau),len(PartialFG)]
		Capex_Solid	Capita Cost of Belt Band model	[k€]	array	len(PartialFG)
		Capex_Belt	Capita Cost of Solid Band model	[k€]	array	len(PartialFG)
		Capex	Capita Cost Average	[k€]	array	len(PartialFG)
		AnnualRev_Belt	Annual Revenue of Solid Band model	[k€/year]	matrix	[len(Tau),len(PartialFG)]
		AnnualRev_Solid	Annual Revenue of Belt Band model	[k€/year]	matrix	[len(Tau),len(PartialFG)]
		AnnualRev	Annual Revenue Average	[k€/year]	matrix	[len(Tau),len(PartialFG)]
		Opex_Belt	O&M Cost of Solid Band model	[k€/year]	matrix	[len(Tau),len(PartialFG)]
		Opex_Solid	O&M Cost of Belt Band model	[k€/year]	matrix	[len(Tau),len(PartialFG)]
		Opex	O&M Cost Average	[k€/year]	matrix	[len(Tau),len(PartialFG)]
		FixC_Belt	Fix Cost of Solid Band model	[k€/year]	array	len(PartialFG)
		FixC_Solid	Fix Cost of Belt Band model	[k€/year]	array	len(PartialFG)
		FixC	Fix Cost Average	[k€/year]	array	len(PartialFG)
		FuelCost	Fuel Cost	[k€/year]	array	[len(Tau),len(PartialFG)]
		ElecCost	Electricity Cost	[k€/year]	array	[len(Tau),len(PartialFG)]
		NPV15max	Maximum Present Value of the project at the end of lifespan of each running time considered	[k€]	array	len(Tau)
		PartialFG15max	Share of FG in the Dryer corresponding to the NPV15max	[-]	array	len(Tau)
		NPV15max_max	Maximum Present Value of the project at the end of lifespan of each running time considered if CAPEX is 30% lower	[k€]	array	len(Tau)
		PartialFG15max_max	Share of FG in the Dryer corresponding to the NPV15max_max	[-]	array	len(Tau)
		NPV15max_min	Maximum Present Value of the project at the end of lifespan of each running time considered if CAPEX is 30% higher	[k€]	array	len(Tau)
		PartialFG15max_min	Share of FG in the Dryer corresponding to the NPV15max_min	[-]	array	len(Tau)

Class	Instance / SubClass	Instance of SubClass	Description	index/units	type	Size/Shape
	FullLoad	NPV5max	Maximum Present Value of the projectat the fifth year of each running time considered	[k€]	array	len(Tau)
		PartialFG5max	Share of FG in the Dryer corresponding to the NPV5max	[-]	array	len(Tau)
		NPV5max_max	Maximum Present Value of the projectat the fifth year of each running time considered if CAPEX is 30% lower	[k€]	array	len(Tau)
		PartialFG5max_max	Share of FG in the Dryer corresponding to the NPV5max_max	[-]	array	len(Tau)
		NPV5max_min	Maximum Present Value of the projectat the fifth year of each running time considered if CAPEX is 30% higher	[k€]	array	len(Tau)
		PartialFG5max_min	Share of FG in the Dryer corresponding to the NPV5max_min	[-]	array	len(Tau)
		MCmin_opt_eco	Optimal humidity for the NPV15max	[%ad]	array	len(Tau)
		Effimax_opt_eco	Optimal efficiency for the NPV15max	[-]	array	len(Tau)
	LoadCurve	PartialFG	Array of Partial FG. Share of the FG introduced into the Dryer	[-]	array	len(PartialFG)
		Load	Demand Curve	[W]	array	len(Hours)
		Hours	Array that contains the quantity of hours that a value of the Demand Curve takes place	[h]	array	len(Hours)
		Hours_cum	Array that contains the cumulative quantity of hours that a value of the Demand Curve takes place	[h]	array	len(Hours)
		P_BoilerFactor	Load Factor of the Boiler with respect to the Demand Curve	[-]	array	len(Hours)
		T_FG_boiler	FG temperature at the outlet of the boiler	[K]	array	len(Hours)
		Initial Conditions. Without Dryer				
		MassFuelini_daf	Mass flow of Fuel in initial conditions, dry ash free basis	[kgFueldaf/s]	array	len(Hours)
		Ashini_ad	Ash content per kg of total fuel , initial conditions	[(kgAsh/kgFuelAd)*100], [%]	float	-
		MassFuelini_ad	Mass flow of Fuel in initial conditions, As delivered basis	[kgFuelad/s]	array	len(Hours)
		Effini	Boiler efficiency in initial conditions	[-]	array	len(Hours)
		LHVini_daf	LHV of Fuel in initial conditions, dry ash free basis	[J/kgFueldaf]	float	-
		LHVini_ad	LHV of Fuel in initial conditions, As delivered basis	[J/kgFuelad]	float	-
		m_air_esp_daf_ini	Amount of air introduced in the boiler per kg of Fuel dry ash free basis, initial conditions	[kgAir/kgFueldaf]	array	len(Hours)
		m_air_esp_ad_ini	Amount of air introduced in the boiler per kg of Fuel As delivered basis, initial conditions	[kgAir/kgFueldad]	array	len(Hours)
		Drying contidions				
		MCfin_ad	Humidity of Fuel at the inlet of the Boiler after being dryed by the PartialFG	[%ad]	matrix	len(Hours), len(PartialFG)
		Effi	Boiler efficiency with Fuel dried	[-]	matrix	len(Hours), len(PartialFG)
		MassFG_Boiler	Mass flow of Flue Gas at the outlet of the Boiler, dry basis	[kgdb/s]	matrix	len(Hours), len(PartialFG)
		MassFuel_daf	Mass flow of Fuel dried, dry ash free basis	[kgFueldaf/s]	matrix	len(Hours), len(PartialFG)
		MassFuel_ad	Mass flow of Fuel dried, As delivered basis, at the inlet of the system Dryer+Boiler	[kgFuelad/s]	matrix	len(Hours), len(PartialFG)
		MassAir	Mass flow of Air that is mixed with the FG from the boiler, dry basis	[kgdb/s]	matrix	len(Hours), len(PartialFG)
		MassFG_Mixer	Mass flow of Flue Gas at the inlet of the Mixer, dry basis	[kgdb/s]	matrix	len(Hours), len(PartialFG)
		MassFG_Dryer	Total Mass flow FG+Air at the inlet of the Dryer, dry basis	[kgdb/s]	matrix	len(Hours), len(PartialFG)
		Ash_Boiler_ad	Ash content per kg of total fuel at the inlet of the Boiler	[(kgAsh/kgFuelAd)*100], [%]	matrix	len(Hours), len(PartialFG)
		MassFuel_Boiler_ad	Mass flow of Fuel dried, As delivered basis, at the inlet of Boiler	[kgFuelad/s]	matrix	len(Hours), len(PartialFG)
		LHV_daf	LHV of Fuel, dry ash free basis, at the inlet of the Boiler	[J/kgFueldaf]	matrix	len(Hours), len(PartialFG)
		LHV_ad	LHV of Fuel, Ash delivered basis, at the inlet of the Boiler	[J/kgFuelad]	matrix	len(Hours), len(PartialFG)
		m_air_esp_daf	Amount of air introduced in the boiler per kg of Fuel dry ash free basis	[kgAir/kgFueldaf]	matrix	len(Hours), len(PartialFG)
		m_air_esp_ad	Amount of air introduced in the boiler per kg of Fuel at the inlet of the Boiler As delivered basis	[kgAir/kgFueldad]	matrix	len(Hours), len(PartialFG)
		ExcessAir	Amoung of Air in excess in the Boiler	[mol/molFuel]	matrix	len(Hours), len(PartialFG)
		MC_FG_boiler_wb	Humidity of FG at the outlet of the Boiler/Inlet of the Mixer, per kg of FG total	[kgH2O/kgFGwb]	matrix	len(Hours), len(PartialFG)
		MC_FG_boiler_db	Humidity of FG at the outlet of the Boiler/Inlet of the Mixer, per kg of FG dry basis	[kgH2O/kgFGdb]	matrix	len(Hours), len(PartialFG)
		MC_FG_dryer_wb	Humidity of FG at the inlet of the Dryer, per kg of FG total	[kgH2O/kgFGwb]	matrix	len(Hours), len(PartialFG)
		MC_FG_dryer_db	Humidity of FG at the the inlet of the Dryer, per kg of FG dry basis	[kgH2O/kgFGdb]	matrix	len(Hours), len(PartialFG)
		EffiGlobal	Global efficiency of the system Dryer+Boiler	[-]	array	len(PartialFG)
		me	Evaporation rate of water in the dryer	[kgH2O/s]	matrix	len(Hours), len(PartialFG)
		HeatExc	Heat exchanged in the dryer	[kW]	matrix	len(Hours), len(PartialFG)
		Drying_Agent	Amount of Mix FG+Air in the Dryer per amount of Fuel dried, dry ash free basis	[kgMix/kGFueldaf]	matrix	len(Hours), len(PartialFG)

Class	Instance / SubClass	Instance of SubClass	Description	index/units	type	Size/Shape
	LoadCurve	Drying_Agent_FG	Amount of FG in the Mixer per amount of Fuel dried, dry ash free basis	[kgFG/kGFueldaf]	matrix	len(Hours), len(PartialFG)
		Heat_Consum	Heat exchanged per amount of water evaporated	[kJ/kgH2O]	matrix	len(Hours), len(PartialFG)
		Heat_Consum_e	Heat exchanged per amount of Fuel dried, dry ash free basis	[kJ/kgFueldaf]	matrix	len(Hours), len(PartialFG)
		TFGout	Temperature at the outlet of the Dryer	[K]	matrix	len(Hours), len(PartialFG)
		LF	Load Factor of the Boiler	[-]	float	-
		DemCov	Demand covered by the Boiler. By default, in FullLoad Case, =1	[-]	float	-
		Years	lifespan of the project	[years]	int	-
		NPVHolmber	Net Present Value of Holmberg model	[k€]	matrix	Nyears,len(partialFG)
		NPVBrammer	Net Present Value of Brammer model	[k€]	matrix	Nyears,len(partialFG)
		NPVSolid	Net Present Value of Solid Band model	[k€]	matrix	Nyears,len(partialFG)
		NPVSolid_max	Net Present Value of Solid Band model if CAPEX is 30% lower	[k€]	matrix	Nyears,len(partialFG)
		NPVSolid_min	Net Present Value of Solid Band model if CAPEX is 30% higher	[k€]	matrix	Nyears,len(partialFG)
		NPVBelt	Net Present Value of Belt Band model	[k€]	matrix	Nyears,len(partialFG)
		NPVBelt_max	Net Present Value of Belt Band model if CAPEX is 30% lower	[k€]	matrix	Nyears,len(partialFG)
		NPVBelt_min	Net Present Value of Belt Band model if CAPEX is 30% higher	[k€]	matrix	Nyears,len(partialFG)
		NPV	Net Present Value, Belt and Solid Average	[k€]	matrix	Nyears,len(partialFG)
		NPV_max	Net Present Value, Belt and Solid Average if CAPEX is 30% lower	[k€]	matrix	Nyears,len(partialFG)
		NPV_min	Net Present Value, Belt and Solid Average if CAPEX is 30% higher	[k€]	matrix	Nyears,len(partialFG)
		RTHolmberg	Return time of investment, Holmberg model	[years]	array	len(PartialFG)
		RTBrammer	Return time of investment, Brammer model	[years]	array	len(PartialFG)
		RTSolid	Return time of investment, Solid Band model	[years]	array	len(PartialFG)
		RTBelt	Return time of investment, Belt Band model	[years]	array	len(PartialFG)
		RT	Return time of investment, Belt and Solid Average	[years]	array	len(PartialFG)
		Capex_Belt	Capita Cost of Solid Band model	[k€]	array	len(PartialFG)
		Capex_Solid	Capita Cost of Belt Band model	[k€]	array	len(PartialFG)
		Capex	Capita Cost Average	[k€]	array	len(PartialFG)
		AnnualRev_Belt	Annual Revenue of Solid Band model	[k€/year]	array	len(PartialFG)
		AnnualRev_Solid	Annual Revenue of Belt Band model	[k€/year]	array	len(PartialFG)
		AnnualRev	Annual Revenue Average	[k€/year]	array	len(PartialFG)
		Opex_Belt	O&M Cost of Solid Band model	[k€/year]	array	len(PartialFG)
		Opex_Solid	O&M Cost of Belt Band model	[k€/year]	array	len(PartialFG)
		Opex	O&M Cost Average	[k€/year]	array	len(PartialFG)
		FixC_Belt	Fix Cost of Solid Band model	[k€/year]	array	len(PartialFG)
		FixC_Solid	Fix Cost of Belt Band model	[k€/year]	array	len(PartialFG)
		FixC	Fix Cost Average	[k€/year]	array	len(PartialFG)
		FuelCost	Fuel Cost	[k€/year]	array	len(PartialFG)
		ElecCost	Electricity Cost	[k€/year]	array	len(PartialFG)

Appendix B

Main program

Code of the different programs written or adapted from Coppieters et al. [49] used in the projec in order to perform the required calculations.

B.1 Constants

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Jan 13 17:42:07 2019

@author: thibaultcoppieters
@modifiedby: Salvador Griñán

"""
#from cantera import Solution as sol 
from MyProperties_Water import MyProperties_Water_2
import numpy as np
# =============================================================================
# # Temperature and Pressure references 
# =============================================================================
T_ref=0+273.15  #[K]
P_ref=101325    #[Pa]
# =============================================================================
# # Constants
# =============================================================================
R = 8.314; # Ideal gas constant [J/K/mol]
# =============================================================================
# # Molar Mass
# =============================================================================
Mm_C=12.0107        #[kg/kmol]
Mm_O=15.999
Mm_N=14.0067
Mm_H=1.00794
Mm_S=32.065
Mm_CO2=Mm_C+2*Mm_O  #[kg/kmol]
Mm_O2=2*Mm_O
Mm_N2=2*Mm_N
Mm_H2O=18.02
Mm_CO=Mm_C+Mm_O
Mm_H2=2*Mm_H
Mm_Air=0.79*Mm_N2+0.21*Mm_O2
Mm_SO2=Mm_S+Mm_O2
# =============================================================================
# # Air of combustion and characteristics 
# =============================================================================
T_air_combustion=20+273.15
T_air_mixer=T_air_combustion
P_air_combustion=P_ref
CO2_db_air=0/100        # Vol %dry
O2_db_air=21/100        # Vol %dry
N2_db_air=79/100        # Vol %dry
compo_db_air=np.array([CO2_db_air, O2_db_air,N2_db_air])   #compo is a vector in vol fraction
#[N2 O2 CO2  H2O SO2 CO H2 CH4 C2H4 C2H6 C3H8 C4H10 H2S] [m3/m3] 
Ratio_H2O_air=0.01*Mm_Air/Mm_H2O
# =============================================================================
# # DH WATER
# =============================================================================
T_cw_OUT=50+273.15      #[K]          # T OUT DH WATER initial value has no influence  
T_cw_IN=90+273.15       #[K]          # T IN DH WATER initial value has no influence   
P_cw_OUT=5*10**5        #[Pa]         # P DH WATER initial value has no influence  
CW_OUT=MyProperties_Water_2(T_cw_OUT,P_cw_OUT)
# =============================================================================
# # FUEL
# =============================================================================
T_Fuel_Ad=15+273.15 #[K]
MCini_ad=40         #[%ad] Wet Basis
MCfin_min_ad=10     #[%ad] Wet Basis
# =============================================================================
# # BOILER
# =============================================================================
P_nominal=18*10**6
T_FG_boiler_100=180+273.15
T_FG_boiler_30=100+273.15
#Temperature curve
a_temp=(T_FG_boiler_100-T_FG_boiler_30)/(1-0.3)
```

```
b_temp=T_FG_boiler_100-a_temp*1
# =============================================================================
# # NG BOILER / ANCILLARY SYSTEM
# =============================================================================
bNG= 28.9       #[€/MWh] # #https://appsso.eurostat.ec.europa.eu/nui/submitViewTableAction.do
effi_NG=0.95
# =============================================================================
# # Mixer
# =============================================================================
T_FG_dryer=T_FG_boiler_100 #Temperature of the drying agent at the inlet of the dryer
# =============================================================================
# #DRYER
# =============================================================================
Coef_losses=5/100       #[-]    #losses in the energy balance
HR_FG_OUT_max=0.9       #[-]    #Relative humidty max at the outlet of the dryer
deltaP_dryer=1000       #[Pa]   #Hipothesis
effiFAN=0.95            #[-]    #Fan efficiency
VFG=0.65                #[m/s]  #FG velicity 
# =============================================================================
# # Economic Parameters
# =============================================================================
DollarToEuro=0.92   #€/$
Nyears=15           #Lifespan
infl=0.0            #Inflation
irate=0.05          #Discount rate
#O&M Cost
k_FixedCost=0.05       #Considering as a percentage of the Capital Cost  #[-] 
be=184.8            # Cost of electricity   #[€/MWh] #25
#be=100            # Cost of electricity   #[€/MWh] #25
bFuel=34            # Fuel Cost             #[€/MWh] #24 #8.3 

# =============================================================================
# # CEPCI INDEX
# =============================================================================
CEPCI2000=394.1
CEPCI2001=394.3
CEPCI2002=395.6
CEPCI2018=615.9
#CEPCI2018=395.6
CEPCI2017=567.5
CEPCIRef=1000   #Index considered in "Rules of Thumb"
# =============================================================================
# # BARE MODULE
# =============================================================================
##RULES OF THUMB
#f_PM=0.2    #[-] # 15-25% of FOB #Taxes,  freight and insurance
#f_BM=0.2    #[-] # 10-45% of LandM #offsites+indirects for home office and field expenses
#f_TM1=0.04  #[-] #3-5% of BM #Contractor fees
#f_TM2=0.2   #[-] #10-30% of BM #Contingency for unexpected delays
#f_FC=0      #[-] # % of TM # Royalties and licenses,Land,Spare parts,Legal fees,Working capital,
#                 # Startup expenses

#RULES OF THUMB
f_PM=0.15   #[-] # 15-25% of FOB #Taxes,  freight and insurance
f_BM=0.2    #[-] # 10-45% of LandM #offsites+indirects for home office and field expenses
f_TM1=0.05  #[-] #3-5% of BM #Contractor fees
f_TM2=0.10  #[-] #10-15% of BM #Contingency for unexpected delays
f_FC=0      #[-] # % of TM # Royalties and licenses,Land,Spare parts,Legal fees,Working capital,
                 # Startup expenses

# =============================================================================
# # Holmber 2004 parameters
# =============================================================================
eps=0.57        #Volume fraction of air in bed 
G=1.6           #Lang Factor
RatioAdAconveyor=1.1

k_conveyor=2700         #Year 2000
exp_conveyor=1
k_AirDuct=3770          #Year 2002
```

```
exp_AirDuct=0.5
k_Fan=0.9*deltaP_dryer  #Year 2002
exp_Fan=0.7
k_Cover=1200            #Year 2002
exp_Cover=0.5
```

B.2 Main code

```python
# -*- coding: utf-8 -*-
"""
Created on Fri Feb 21 10:01:53 2020

@author: salva
"""

import numpy as np
import pandas as pd
from scipy.optimize import brentq
import pickle as pickle

from Constants import Nyears,MCini_ad,P_nominal, P_ref,VFG,bFuel,k_FixedCost
from Constants import T_FG_dryer,T_FG_boiler_100, MCfin_min_ad,compo_db_air

from Boiler2 import Boiler,fuel_properties
from Dryer3 import Dryer, It_FuelMC_InBoiler,It_MassFGTotalDried
from GetProps import Get_Rho_Average

from ResultClass2 import Result
from MyProperties_Gas import MyProperties_Gas_2

BaseCase_file="BaseCase1.pkl"
Results_file="ResultsBest.pkl"

# =============================================================================
"Nomenclature"
# FG   --> Flue Gas
# Fuel --> Fuel
# ini --> Initial conditions || Without Dryer
# nom/nominal --> Condition of Full Load || Nominal rate conditions
# daf,db,ad   --> dry ash free, dry basis and as delivered basis, respectively

# Mass --> Mass Flow, [kg/s]
# MC   --> Moisture Content, [%] for Fuel, [-] for FG
# T    --> Temperature, [K]
# P    --> Power | Demand, [W]
# ETA  --> Efficiency, [-]
# me   --> Evaporation rate, [kg/s]
# Heat_exc --> Heat Exchanged in Dryer, [kW]
# LHV  --> Low Heat Value of Fuel, [J/kgx] where x could be daf,db,ad
# HHV  --> High Heat Value of Fuel, [J/kgx] where x could be daf,db,ad

# Tau --> Running time array, [h]
# Hours --> number of hours that a Load is repeated [h]
# Load --> Curve Demand [W]

# Aconveyor --> Conveyor surface of the Dryer, [m2]
# Rho_Average --> Average density of FG inlet/outlet of the Dryer,[kg/m3].!! CURRENTLY: rhoFGin
    
# NPV --> Net Present Value, [k€]
# RT  --> Payback period, Return time [year]
"Identation"
# Fuel 1 --> Inlet of the Dryer
# Fuel 2 --> Outlet of the Dryer || Inlet of the Boiler
# FG 1   --> Outlet of the Boiler
# FG 2   --> Inlet of the Mixer
# FG 21  --> Oulet of the Mixer || Inlet of the Dryer ||Air + FG
# FG 3   --> Outlet of the Dryer
# =============================================================================

print("P",P_nominal,"MW MCini",MCini_ad,"% FuelC", bFuel,"€/MWh FixC",k_FixedCost)

"Hypothesis"
"Boiler"
P_boiler_nom=P_nominal
T_FG_nom=T_FG_boiler_100
"Dryer"
T_FG_21=T_FG_dryer
MC_Fuel_1_ad=MCini_ad
```

```
# =============================================================================
# Loading LOAD CURVE and Griding
# =============================================================================
"Load Curve"
filename='RealCase5.xlsx'
filePath=filename
df = pd.read_excel (filePath, sheet_name='data',index_col=0,header=0,nrows=8610,skiprows=0)
df = df.apply(pd.to_numeric, errors='coerce')
df=df.dropna()
df.columns=(['h','Load Factor[-]'])
Hours_values=df.loc[:,'h'].values
Hourspervalue=Hours_values[1]
LoadFactor=df.loc[:,'Load Factor[-]'].values
Load=LoadFactor*P_boiler_nom

#GRID
Nsteps=100
Load_step=(Load.max()-Load.min())/Nsteps
Load_grid=np.arange(Load.min(),Load.max()+Load_step,Load_step)
Load_grid_nbr=len(Load_grid)
Weight=np.zeros(Nsteps+1)
Load_new=np.zeros(Nsteps+1)

for l,_ in enumerate(np.arange(0,Load_grid_nbr-1)):
    Load_min=Load_grid[l]
    Load_max=Load_grid[l+1]
    arg=np.argwhere((Load>=Load_min) & (Load<Load_max))

    if np.isnan(len(arg)):
        Weight[l]=0
    else:
        Weight[l]=len(arg)
    if len(arg)<=0:
        Load_new[l]=0
    else:
        Load_new[l]=np.mean(Load[arg])

index=np.argsort(-Load_new)
Load_new=Load_new[index]
Weight=Weight[index]
Hours=Weight*Hourspervalue
Load=Load_new #kW
if Hours[len(Hours)-1]==0:
    Hours=np.delete(Hours,len(Hours)-1)
    Load=np.delete(Load,len(Load)-1)
# =============================================================================
# Initialization of arrays
# =============================================================================
#Partial FG and Tau
StepPartialFG=0.05
TauSteps=50
TauMax=8760

Tau_array=np.append(np.arange(1000,np.ceil(TauMax/TauSteps)*TauSteps,TauSteps),TauMax)
PartialFG_array=np.append(0.00001,np.append(np.arange(StepPartialFG,1,StepPartialFG),1))

#rows=hour   columns=PartialFG
Nrows=len(Hours)
Ncolumns=len(PartialFG_array)
P_NG=np.zeros(Nrows)
P_Boiler=np.zeros(Nrows)

Nhours_NG=0
Nhours_Boiler=0
Q_Boiler_total=0
Q_NG_total=0
# #Full Load arrays
# =============================================================================
MC_Fuel_2_ad_nom_array=np.zeros(Ncolumns)
ETA_Boiler_nom_array=np.zeros(Ncolumns)
```

```
T_FG_3_nom_array=np.zeros(Ncolumns)
T_Fuel_2_nom_array=np.zeros(Ncolumns)

Mass_Fuel_daf_nom_array=np.zeros(Ncolumns) #MassFlow using the dryer
Mass_FG_1_db_nom_array=np.zeros(Ncolumns)
Mass_FG_2_db_nom_array=np.zeros(Ncolumns)
Mass_FG_21_db_nom_array=np.zeros(Ncolumns)
Mass_Air_db_nom_array=np.zeros(Ncolumns)

me_nom_array=np.zeros(Ncolumns)
Heat_exc_nom_array=np.zeros(Ncolumns)

Mass_Fuel_ini_daf_array=np.zeros(Nrows)
ETA_Boiler_ini_array=np.zeros(Nrows)
# =============================================================================
# #Load curve matrix and arrays
# =============================================================================
MC_Fuel_2_ad_matrix=np.zeros((Nrows,Ncolumns))
ETA_Boiler_matrix=np.zeros((Nrows,Ncolumns))

T_FG_3_matrix=np.zeros((Nrows,Ncolumns))
T_Fuel_2_matrix=np.zeros((Nrows,Ncolumns))

Mass_Fuel_daf_matrix=np.zeros((Nrows,Ncolumns)) #MassFlow using the dryer
Mass_FG_1_db_matrix=np.zeros((Nrows,Ncolumns))
Mass_FG_2_db_matrix=np.zeros((Nrows,Ncolumns))
Mass_FG_21_db_matrix=np.zeros((Nrows,Ncolumns))
Mass_Air_db_matrix=np.zeros((Nrows,Ncolumns))

me_matrix=np.zeros((Nrows,Ncolumns))
Heat_exc_matrix=np.zeros((Nrows,Ncolumns))
# =============================================================================
# #Final results array
# =============================================================================
Aconveyor_array=np.zeros(Ncolumns)
Rho_Average_array=np.zeros(Ncolumns)
# =============================================================================
#Calculation of the Fuel Mass Flow and efficiency without Dryer---INITIAL CONDITIONS
# =============================================================================

'Full Load'
#Boiler

[FG,Mass_FG_1_db,Mass_FG_1_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,
 ETA_Boiler]=Boiler(P_boiler_nom,P_boiler_nom,MCini_ad)
#Fuel properties
[MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,HHV_Fuel_daf
 ]=fuel_properties(MCini_ad,"fuel_properties")
#Efficiency and Mass fuel
ETA_Boiler_nom_ini=ETA_Boiler
Mass_Fuel_nom_ini_daf=(1-MCini_ad/100-Ash_Fuel_ad/100) * Fuel_Mass_tot

'Load Curve'
for j in range(Nrows):
    #If the demand is lower than the minimum capacity       --> Ancillary systems only
    if Load[j]<0.3*P_boiler_nom:
        #NG Boiler
        P_NG[j]=Load[j]
        Nhours_NG=Nhours_NG+Hours[j]
        Q_NG_total=Q_NG_total+P_NG[j]*Hours[j]
        #Biomass Boiler
        P_Boiler[j]=0
        Nhours_Boiler=Nhours_Boiler
        Q_Boiler_total=Q_Boiler_total
    #If the demand is between minimum and maximum capacity  --> Biomass boiler only
    elif Load[j]>0.3*P_boiler_nom and Load[j]<P_boiler_nom:
        #NG Boiler
        P_NG[j]=0
        Nhours_NG=Nhours_NG
```

```
        Q_NG_total=Q_NG_total
        #Biomass Boiler
        P_Boiler[j]=Load[j]
        Nhours_Boiler=Nhours_Boiler+Hours[j]
        Q_Boiler_total=Q_Boiler_total+P_Boiler[j]*Hours[j]
    #If the demand is more than maximum capacity        --> Biomass boiler + Ancillary systems
    elif Load[j]>P_boiler_nom:
        #NG Boiler
        P_NG[j]=Load[j]-P_boiler_nom
        Nhours_NG=Nhours_NG+Hours[j]
        Q_NG_total=Q_NG_total+P_NG[j]*Hours[j]
        #Biomass Boiler
        P_Boiler[j]=P_boiler_nom
        Nhours_Boiler=Nhours_Boiler+Hours[j]
        Q_Boiler_total=Q_Boiler_total+P_Boiler[j]*Hours[j]
    
    #When the biomass boiler is running:
    if P_Boiler[j]>0:
        if P_Boiler[j]==P_boiler_nom: #Full Load conditions
            ETA_Boiler_ini_array[j]=ETA_Boiler_nom_ini
            Mass_Fuel_ini_daf_array[j]=Mass_Fuel_nom_ini_daf
        else:
            #Boiler
            [FG,Mass_FG_1_db,Mass_FG_1_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,
             ETA_Boiler]=Boiler(P_Boiler[j],P_boiler_nom,MCini_ad)
            #fuel properties
            [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,
             HHV_Fuel_daf]=fuel_properties(MCini_ad,"fuel_properties")
            #Efficiency and Mass fuelx
            ETA_Boiler_ini_array[j]=ETA_Boiler
            Mass_Fuel_ini_daf_array[j]=(1-MCini_ad/100-Ash_Fuel_ad/100) * Fuel_Mass_tot

# =============================================================================
# #Calculations---DRYING CONDITIONS
# =============================================================================
            
'Calculation of Moisture content of Fuel at the inlet of the...'
'...boiler as a function of the Partial Flue Gas Mass flow in the Dryer'

for i in range(Ncolumns): #Going through the FG share array
    print("i",i)
    'Full Load'
    print("Full Load")
    PartialMass_2_db_max=25
    'Max drying condition in the Full Load case'
    #Boiler
    [FG_1,Mass_FG_1_db,Mass_FG_1_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,
     ETA_Boiler]=Boiler(P_boiler_nom,P_boiler_nom,MCfin_min_ad)
    #fuel properties
    [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,
        HHV_Fuel_daf]=fuel_properties(MCfin_min_ad,"fuel_properties")
    #Function for the iteration of the Humidity of Fuel in the Boiler
    fun1=lambda MC_Fuel_it : It_FuelMC_InBoiler(MC_Fuel_it,MC_Fuel_1_ad,P_boiler_nom,
                                               P_boiler_nom,PartialFG_array[i],Mass_FG_2_db_max)
    #Mass fuel
    Mass_Fuel_daf=(1-MCfin_min_ad/100-Ash_Fuel_ad/100)*Fuel_Mass_tot
    fun2=lambda Mass_FG_2_db_it : It_MassFGTotalDried(Mass_FG_2_db_it,Mass_Fuel_daf,FG_1,
                                                      MC_Fuel_1_ad)
    #PartialFG max:
    Mass_FG_2_db_max=brentq(fun2,0.001*Mass_FG_1_db,10*Mass_FG_1_db,xtol=0.0001)
    PartialMass_2_db_max=Mass_FG_2_db_max/Mass_FG_1_db
    'Thermal properties calculation'
    #If the PartialMass is NOT high enought to dry completely the Fuel:
    if PartialFG_array[i]<PartialMass_2_db_max:
        MC_Fuel_2_ad=MC_Fuel_1_ad
        if PartialFG_array[i] != 0:
            #brentq(funct2solve,lowerBound,upperBound)
            MC_Fuel_2_ad=brentq(fun1,MCfin_min_ad,MC_Fuel_1_ad,xtol=0.0001) 
        MC_Fuel_2_ad_nom_array[i]=MC_Fuel_2_ad
    #If the amount of Partial Mass is enought to dry completely the Fuel
```

```
    else:
        MC_Fuel_2_ad_nom_array[i]=MCfin_min_ad

    #Boiler
    [FG_1,Mass_FG_1_db,Mass_FG_1_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,
     ETA_Boiler]=Boiler(P_boiler_nom,P_boiler_nom,MC_Fuel_2_ad_nom_array[i])
    #fuel properties
    [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,
     HHV_Fuel_daf]=fuel_properties(MC_Fuel_2_ad_nom_array[i],"fuel_properties")
    #FG properties out boiler
    FG_2_nom=MyProperties_Gas_2(FG_1.compo_db,FG_1.y_H2O,FG_1.T,P_ref)
    #Efficiency and Mass fuel
    ETA_Boiler_nom_array[i]=ETA_Boiler
    Mass_Fuel_daf=(1-MC_Fuel_2_ad_nom_array[i]/100-Ash_Fuel_ad/100)*Fuel_Mass_tot
    Mass_Fuel_daf_nom_array[i]=Mass_Fuel_daf
    #Mass FG
    Mass_FG_1_db_nom_array[i]=Mass_FG_1_db
    Mass_FG_2_db_nom_array[i]=min(Mass_FG_1_db*PartialFG_array[i],Mass_FG_2_db_max)
    #if PartialMass_2_db !=0:
    #Dryer
    [FG_3,FG_21,Mass_FG_21_db,T_Fuel_2,me,Heat_exc]=Dryer(FG_2_nom,MC_Fuel_1_ad,
     MC_Fuel_2_ad_nom_array[i],Mass_Fuel_daf,Mass_FG_1_db,PartialFG_array[i],Mass_FG_2_db_nom_array[i],Mass_FG_2_db_max)
    #FG properties out dryer
    FG_3_nom=MyProperties_Gas_2(FG_3.compo_db,FG_3.y_H2O,FG_3.T,P_ref)
    FG_21_nom=MyProperties_Gas_2(FG_21.compo_db,FG_21.y_H2O,FG_21.T,P_ref)
    
    T_FG_3_nom_array[i]=FG_3.T
    Mass_FG_21_db_nom_array[i]=Mass_FG_21_db
    Mass_Air_db_nom_array[i]=Mass_FG_21_db_nom_array[i]-Mass_FG_2_db_nom_array[i]
    #Other termal properties
    me_nom_array[i]=me
    Heat_exc_nom_array[i]=Heat_exc
    T_Fuel_2_nom_array[i]=T_Fuel_2
    
    #Density of FG at inlet instead of average
    Rho_Average_array[i]=Get_Rho_Average(FG_21_nom,FG_21_nom)

    'Load Curve'
    print("Load Curve")
    PartialMass_2_db_max=np.inf
    for j in range(Nrows): #Going through the Hours array
        PartialMass_2_db_max=10
#        print('i',i,'j',j)
        if P_Boiler[j]>0:
            if P_Boiler[j]==P_boiler_nom: #Nominal demand
                #Results = nominal results solved above:
                FG_2=MyProperties_Gas_2(FG_2_nom.compo_db,FG_2_nom.y_H2O,FG_2_nom.T,P_ref)
                
                Mass_FG_1_db_matrix[j,i]=Mass_FG_1_db_nom_array[i]
                Mass_FG_2_db_matrix[j,i]=Mass_FG_1_db_nom_array[i]*PartialFG_array[i]
                T_FG_3_matrix[j,i]=T_FG_3_nom_array[i]
                
                Mass_FG_21_db_matrix[j,i]=Mass_FG_21_db_nom_array[i]
                Mass_Air_db_matrix[j,i]=Mass_Air_db_nom_array[i]
                
                T_Fuel_2_matrix[j,i]=T_Fuel_2_nom_array[i]
                MC_Fuel_2_ad_matrix[j,i]=MC_Fuel_2_ad_nom_array[i]
                Mass_Fuel_daf_matrix[j,i]=Mass_Fuel_daf_nom_array[i]
                ETA_Boiler_matrix[j,i]=ETA_Boiler_nom_array[i]
                
                me_matrix[j,i]=me_nom_array[i]
                Heat_exc_matrix[j,i]=Heat_exc_nom_array[i]
                
            else: #If P_Boiler[j] != P_boiler_nom #Out of nominal rate
                #Function for the iteration of the Humidity of Fuel in the Boiler
                fun1=lambda MC_Fuel_it : It_FuelMC_InBoiler(MC_Fuel_it,MC_Fuel_1_ad,P_Boiler[j],
                                                           P_boiler_nom,PartialFG_array[i],Mass_FG_2_db_max)
                'Max drying condition in the Load curve'
                #Boiler
```

```
                [FG_1,Mass_FG_1_db,Mass_FG_1_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,
                    ETA_Boiler]=Boiler(P_Boiler[j],P_boiler_nom,MCfin_min_ad)
                #fuel properties
                [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,
                    HHV_Fuel_daf]=fuel_properties(MCfin_min_ad,"fuel_properties")

                Mass_Fuel_daf=(1-MCfin_min_ad/100-Ash_Fuel_ad/100)*Fuel_Mass_tot

                fun2=lambda Mass_FG_2_db_it : It_MassFGTotalDried(Mass_FG_2_db_it,Mass_Fuel_daf,
                                                                  FG_1,MC_Fuel_1_ad)
                #PartialFG max:
                Mass_FG_2_db_max=brentq(fun2,0.01*Mass_FG_1_db,10*Mass_FG_1_db,xtol=0.0005)
                PartialMass_2_db_max=Mass_FG_2_db_max/Mass_FG_1_db
                
                'Thermal properties calculation'
                #If the PartialMass is NOT high enought to dry completely the Fuel:
                if PartialFG_array[i]<PartialMass_2_db_max:
                    MC_Fuel_2_ad=MC_Fuel_1_ad
                    if PartialFG_array[i] != 0:
                        #brentq(funct2solve,lowerBound,upperBound)
                        MC_Fuel_2_ad=brentq(fun1, MCfin_min_ad, MC_Fuel_1_ad,xtol=0.0001)
                    MC_Fuel_2_ad_matrix[j,i]=MC_Fuel_2_ad
                #If the amount of Partial Mass is enought to dry completely the Fuel:
                else:
                    MC_Fuel_2_ad_matrix[j,i]=MCfin_min_ad
                #Boiler
                [FG_1,Mass_FG_1_db,Mass_FG_1_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,
                 ETA_Boiler]=Boiler(P_Boiler[j],P_boiler_nom,MC_Fuel_2_ad_matrix[j,i])
                #fuel properties
                [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,
                 HHV_Fuel_daf]=fuel_properties(MC_Fuel_2_ad_matrix[j,i],"fuel_properties")


                [FG_3,FG_21,Mass_FG_21_db,T_Fuel_2,me,Heat_exc]=Dryer(FG_2,MC_Fuel_1_ad,
                 MC_Fuel_2_ad_matrix[j,i],Mass_Fuel_daf,Mass_FG_1_db,PartialFG_array[i],Mass_FG_2_db_nom_arra
y[i],Mass_FG_2_db_max)

                FG_2=MyProperties_Gas_2(FG_1.compo_db,FG_1.y_H2O,FG_1.T,P_ref)
                Mass_FG_1_db_matrix[j,i]=Mass_FG_1_db
                Mass_FG_2_db_matrix[j,i]=Mass_FG_1_db*PartialFG_array[i]
                Mass_FG_21_db_matrix[j,i]=Mass_FG_21_db
                Mass_Air_db_matrix[j,i]=Mass_FG_21_db_matrix[j,i]-Mass_FG_2_db_matrix[j,i]

                Mass_Fuel_daf=(1-MC_Fuel_2_ad_matrix[j,i]/100-Ash_Fuel_ad/100)*Fuel_Mass_tot


                ETA_Boiler_matrix[j,i]=ETA_Boiler
                Mass_Fuel_daf_matrix[j,i]=Mass_Fuel_daf

                T_FG_3_matrix[j,i]=FG_3.T
                me_matrix[j,i]=me
                Heat_exc_matrix[j,i]=Heat_exc
                T_Fuel_2_matrix[j,i]=T_Fuel_2

#Smoothing
print(Mass_FG_21_db_nom_array)
for i in range(Ncolumns-2):
    if Mass_FG_21_db_nom_array[i+1]<Mass_FG_21_db_nom_array[i]:
        #linear interpolation
        A=(MC_Fuel_2_ad_nom_array[i+1]-MC_Fuel_2_ad_nom_array[i])/(
                MC_Fuel_2_ad_nom_array[i+1]-MC_Fuel_2_ad_nom_array[i+2])
        Mass_FG_21_db_nom_array[i+1]=(Mass_FG_21_db_nom_array[i]-Mass_FG_21_db_nom_array[i+2]*A)/(
                1-A)

for i in range(Ncolumns):
    if Mass_FG_2_db_nom_array[i]<0.001:
        Aconveyor_array[i]=0
    else:
        Aconveyor_array[i]=Mass_FG_21_db_nom_array[i]/(VFG*Rho_Average_array[i]) #Conveyor surface
```

```
#BaseCase_file="BaseCase1.pkl"
#Results_file="Results2.pkl"

"__Creating and saving objets__"
Res=Result(P_boiler_nom,MCini_ad,bFuel,k_FixedCost,FG_2_nom.T,FG_21.T,

           Mass_Fuel_nom_ini_daf,ETA_Boiler_nom_ini,
           Mass_Fuel_daf_nom_array,MC_Fuel_2_ad_nom_array,ETA_Boiler_nom_array,
           Mass_FG_1_db_nom_array,me_nom_array,Heat_exc_nom_array,
           T_FG_3_nom_array,
           Mass_FG_21_db_nom_array,Mass_Air_db_nom_array,
           Tau_array,

           Mass_Fuel_ini_daf_array,ETA_Boiler_ini_array,
           Mass_Fuel_daf_matrix,MC_Fuel_2_ad_matrix,ETA_Boiler_matrix,
           Mass_FG_1_db_matrix,me_matrix,Heat_exc_matrix,
           T_FG_3_matrix,
           Mass_FG_21_db_matrix,Mass_Air_db_matrix,
           Load,P_Boiler,Hours,

           Rho_Average_array,Aconveyor_array,PartialFG_array)

if P_boiler_nom==6*10**6 and MCini_ad==40 and bFuel==24 and k_FixedCost==0.1:

    BaseCase=Result(P_boiler_nom,MCini_ad,bFuel,k_FixedCost,FG_2_nom.T,FG_21.T,

                       Mass_Fuel_nom_ini_daf,ETA_Boiler_nom_ini,
                       Mass_Fuel_daf_nom_array,MC_Fuel_2_ad_nom_array,ETA_Boiler_nom_array,
                       Mass_FG_1_db_nom_array,me_nom_array,Heat_exc_nom_array,
                       T_FG_3_nom_array,
                       Mass_FG_21_db_nom_array,Mass_Air_db_nom_array,
                       Tau_array,
            
                       Mass_Fuel_ini_daf_array,ETA_Boiler_ini_array,
                       Mass_Fuel_daf_matrix,MC_Fuel_2_ad_matrix,ETA_Boiler_matrix,
                       Mass_FG_1_db_matrix,me_matrix,Heat_exc_matrix,
                       T_FG_3_matrix,
                       Mass_FG_21_db_matrix,Mass_Air_db_matrix,
                       Load,P_Boiler,Hours,
            
                       Rho_Average_array,Aconveyor_array,PartialFG_array)
            
#Saving or opening BaseCase and Results file
try:
    with open(BaseCase_file, 'rb') as input:
        BaseCase=pickle.load(input)
except:
    if (P_boiler_nom==6*10**6
        and MCini_ad==40 and bFuel==24
        and k_FixedCost==0.1):

        BaseCase=Result(P_boiler_nom,MCini_ad,bFuel,k_FixedCost,FG_2_nom.T,FG_21.T,

                       Mass_Fuel_nom_ini_daf,ETA_Boiler_nom_ini,
                       Mass_Fuel_daf_nom_array,MC_Fuel_2_ad_nom_array,ETA_Boiler_nom_array,
                       Mass_FG_1_db_nom_array,me_nom_array,Heat_exc_nom_array,
                       T_FG_3_nom_array,
                       Mass_FG_21_db_nom_array,Mass_Air_db_nom_array,
                       Tau_array,

                       Mass_Fuel_ini_daf_array,ETA_Boiler_ini_array,
                       Mass_Fuel_daf_matrix,MC_Fuel_2_ad_matrix,ETA_Boiler_matrix,
                       Mass_FG_1_db_matrix,me_matrix,Heat_exc_matrix,
                       T_FG_3_matrix,
                       Mass_FG_21_db_matrix,Mass_Air_db_matrix,
                       Load,P_Boiler,Hours,

                       Rho_Average_array,Aconveyor_array,PartialFG_array)

        with open(BaseCase_file,'wb') as output:
```

```
            pickle.dump(BaseCase,output,pickle.HIGHEST_PROTOCOL)

#Saving and opening Results
            
try:
    with open(Results_file, 'rb') as input:
        print("1")
        ResultsList=pickle.load(input)
        print("2")

        #Checking that these results are not saved yet
        h=False
        print("3")
        for k in range(len(ResultsList)):
            if (ResultsList[k].P_Boiler != Res.P_Boiler or ResultsList[k].MCini_ad!=Res.MCini_ad
                or ResultsList[k].FuelC!=Res.FuelC or ResultsList[k].FixC!=Res.FixC
                or ResultsList[k].TFGinDryer!=Res.TFGinDryer):
                print("No equal to Result",k)
            else:
                print("Equal to result",k)
                h=False

    if h==True:
        print("New Case")
        ResultsList.append(Res)
        with open(Results_file,'wb') as output:
            print("RESULT",ResultsList)
            pickle.dump(ResultsList,output,pickle.HIGHEST_PROTOCOL)

except:
    print("NO")
    ResultsList=[]
    ResultsList.append(Res)
    with open(Results_file,'wb') as output:
        pickle.dump(ResultsList,output,pickle.HIGHEST_PROTOCOL)
```

B.3 Boiler

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Jan 10 11:59:38 2019

@author: thibaultcoppieters
"""
# Variable names convention

# Mm_X: Molar mass [kg/kmol]
# n: number of moles [kmol]

import numpy as np
from Constants import Mm_C,Mm_O,Mm_N,Mm_H,Mm_S,Mm_H2O,Mm_O2,Mm_CO2,Mm_N2,Mm_SO2,Mm_CO
from Constants import Ratio_H2O_air,T_air_combustion,T_ref,P_ref, a_temp, b_temp
#from MyConstants import
from MyProperties_Gas import MyProperties_Gas_2
from CoolProp.CoolProp import PropsSI
    
    
def fuel_properties(MC_fuel,return_option,fuel_type="FUEL_WOOD",wood_type="WOOD_CHIPS_CONTINO_1"):
    
# Options   
##fuel_type="FUEL_WOOD"        # "FUEL_WOOD" or "FUEL_NG"
#wood_type="WOOD_CHIPS_CONTINO_1"    # "WOOD_CHIPS_CONTINO_1"or "WOOD_CHIPS_CONTINO_2" or 
        #"WOOD_CHIPS_HARDWOODS" or "WOOD_CHIPS_SOFTWOODS" or "WOOD_CHIPS_OAK_BARK" or "WOOD_CHIPS_PINE_BARK"
    
    if fuel_type=="FUEL_WOOD":
        ash_fuel_d=2.7 # hyp (from Contino)
        
        if wood_type=="WOOD_CHIPS_CONTINO_1": # Fransesco Contino Composition (cours)   
            C_daf=50
            H_daf=6
            O_daf=44
            N_daf=0
            S_daf=0
            # 35.17*C_daf/100+116.2*H_daf-11.1*O_daf+10.47*S_daf+6.28*N_daf-0.439       
        elif wood_type=="WOOD_CHIPS_CONTINO_2": # Fransesco Contino Composition (cours)   
            C_daf=50.8
            H_daf=6.06
            O_daf=42.51
            N_daf=0.51
            S_daf=0.08    
        elif wood_type=="WOOD_CHIPS_HARDWOODS":
            # Average of 11 hardwoods from http://www.marioloureiro.net/ciencia/ignicao_vegt/ragla91a.pdf
            # Elemental Analysis (CmHnOxNySz) % Weight percentage
            C_daf=50.2
            H_daf=6.2
            O_daf=43.5
            N_daf=0.1 # neglected
            S_daf=0 # neglected   
        elif wood_type=="WOOD_CHIPS_SOFTWOODS": 
            #Average of 9 softwoods from http://www.marioloureiro.net/ciencia/ignicao_vegt/ragla91a.pdf
            # Elemental Analysis (CmHnOxNySz) % Weight percentage
            C_daf=52.7
            H_daf=6.3;
            O_daf=40.8;
            N_daf=0.2 # neglected
            S_daf=0   # neglected
        elif wood_type=="WOOD_CHIPS_OAK_BARK":
            # OAK bark from http://www.marioloureiro.net/ciencia/ignicao_vegt/ragla91a.pdf
            # Elemental Analysis (CmHnOxNySz) % Weight percentage
            C_daf=52.6
            H_daf=5.7
            O_daf=41.5
            N_daf=0.1 # neglected
            S_daf=0.1 # neglected
        elif wood_type=="WOOD_CHIPS_PINE_BARK":
            # Pine bark from http://www.marioloureiro.net/ciencia/ignicao_vegt/ragla91a.pdf
            # Elemental Analysis (CmHnOxNySz) % Weight percentage
```

```
            C_daf=54.9
            H_daf=5.8
            O_daf=39
            N_daf=0.2 # neglected
            S_daf=0.1 # neglected

        C_d=(100-ash_fuel_d)*C_daf/100
        H_d=(100-ash_fuel_d)*H_daf/100
        O_d=(100-ash_fuel_d)*O_daf/100
        N_d=(100-ash_fuel_d)*N_daf/100
        S_d=(100-ash_fuel_d)*S_daf/100


    elif fuel_type=="FUEL_NG":
        # Elemental Analysis (CmHnOxNySz) % Weight percentage
        C_d=75
        H_d=25
        O_d=0
        N_d=0
        S_d=0
        ash_d=0

    Total_dry=C_d+H_d+O_d+N_d+S_d+ash_fuel_d # Should be equal to 100%
    ash_fuel_daf=ash_fuel_d*100/(100-ash_fuel_d)
    Total_daf=C_daf+H_daf+O_daf+N_daf+S_daf # Should be equal to 100%

    m=1
    n=H_daf/C_daf*Mm_C/Mm_H
    x=O_daf/C_daf*Mm_C/Mm_O
    y=N_daf/C_daf*Mm_C/Mm_N
    z=S_daf/C_daf*Mm_C/Mm_S
    Mm_fuel=m*Mm_C+n*Mm_H+x*Mm_O+y*Mm_N+z*Mm_S

    W_MC_fuel_ad=MC_fuel                           # Weight percent 
    W_ash_fuel_ad=ash_fuel_d*(100-W_MC_fuel_ad)/100 # Weight percent
    W_fuel_ad=100-W_MC_fuel_ad-W_ash_fuel_ad          # dry ash free on as delivered Weight percent
    W_MC_fuel_daf=W_MC_fuel_ad/(1-W_MC_fuel_ad/100-W_ash_fuel_ad/100)
    W_ash_fuel_daf=W_ash_fuel_ad/(1-W_MC_fuel_ad/100-W_ash_fuel_ad/100)
    Xw=W_MC_fuel_ad/W_fuel_ad*Mm_fuel/Mm_H2O           # Mole fraction of water in Biomass mol/mol wet

    Lv=2501*10**3                                          # [J/kg]
    Lv=PropsSI('H','P',P_ref,'Q',1,'Water')-PropsSI('H','P',P_ref,'Q',0,'Water')
#    print("Lv ",Lv)
    HHV_Wood_db=(-1.3675+0.3137*C_d+0.7009*H_d+0.0318*(100-C_d-H_d-ash_fuel_d))*10**6   # [J/kg_db] 
    #(from "Estimating the higher heating value of biomass fuels from basic analysis data")
    HHV_Wood_daf=HHV_Wood_db*1/(1-ash_fuel_d/100)  # [J/kg_daf] LHV Wood Dry basis 18.5-21.0 MJ/kg_daf
    #NCV_Wood_daf=HHV_Wood_daf-2.5*10**6*n/2*18/(12+n+16*x)
    NCV_Wood_daf=HHV_Wood_daf-Lv*n/2*18/(12*m+n+16*x)
    NCV_Wood_ad=NCV_Wood_daf*W_fuel_ad/100-Lv*W_MC_fuel_ad/100   

    if return_option=="Fuel_Compositions":
        return m,n,x,y,z,Xw,Mm_fuel
    elif return_option=="fuel_properties":
        return W_MC_fuel_daf,W_ash_fuel_daf,W_MC_fuel_ad,W_ash_fuel_ad,NCV_Wood_ad,NCV_Wood_daf,HHV_Wood_daf

def FG_Composition(load_boiler,MC_fuel,fuel_type="FUEL_WOOD",wood_type="WOOD_CHIPS_CONTINO_1",
          REVERSE_ENGINEERING="INACTIF",O2_d_or_w="INACTIF",O2_fg_target=None,influence_excess_air="INACTIF"
):

    # Options   
#eta_boiler_method="FIXED_ETA_BOILER" # "FIXED_T_BOILER_FG_OUT" or "FIXED_ETA_BOILER"
##fuel_type="FUEL_WOOD"        # "FUEL_WOOD" or "FUEL_NG"
#wood_type="WOOD_CHIPS_CONTINO_1"    # "WOOD_CHIPS_CONTINO_1"or "WOOD_CHIPS_CONTINO_2" or 
#"WOOD_CHIPS_HARDWOODS" or "WOOD_CHIPS_SOFTWOODS" or "WOOD_CHIPS_OAK_BARK" or "WOOD_CHIPS_PINE_BARK"
#REVERSE_ENGINEERING="INACTIF" # "ACTIF"
#O2_d_or_w="INACTIF" # "O2_on_Wet":
#O2_d_or_w="INACTIF"  #"O2_on_Dry":
#influence_excess_air="INACTIF" #ACTIF"
    [m,n,x,y,z,Xw,Mm_fuel]=fuel_properties(MC_fuel,"Fuel_Compositions",fuel_type,wood_type)
```

```
    n_fuel=1/Mm_fuel # kmol/kg_daf
    K=(m+n/4-x/2+z)
    X=0           # Molar fraction of recirculated combustion gases.
    # Air
    Y_02=0.21/(1+Ratio_H2O_air)             # Volume fraction of O2 in Wet Air
    Y_N2=0.79/(1+Ratio_H2O_air)             # Volume fraction of N2 in Wet Air
    Y_H2O=Ratio_H2O_air/(1+Ratio_H2O_air)   # Volume fraction of H2O in Wet Air
    #print("Air composition: ", "O2: ", Y_02," N2: ",Y_N2," H2O: ", Y_H2O)
    Beta=Y_N2/Y_02                          # Ratio between molar fraction of Nitrogen and Oxygen in wet air.
    Gamma=Y_H2O/Y_02                        # Molar fraction between water and oxygen in wet air;
    
    # Excess air
    if fuel_type=="FUEL_NG":
        lambda_air=0
    
    if fuel_type=="FUEL_WOOD":
        x1=1
        x2=0.3
        y1min=6.5/100
        y1max=7.5/100
#        y1min=6/100
#        y1max=7/100
        
        y2min=8/100
        y2max=9/100
#        y2min=8/100
#        y2max=9.5/100
#        y2max=8/100
        MC_low=30
        MC_high=40
        
        if MC_fuel<=MC_low:  #Combustible with moisture content between 0 and 35%
            y1=y1min # Oxygen content at 100%
            y2=y2min   # Oxygen content at 30%
            A_coef=(y1-y2)/(x1-x2)
            B_coef=y1-A_coef*x1
            O2_w_fg_target=A_coef*load_boiler+B_coef
                
        elif MC_fuel>MC_low and MC_fuel<MC_high:
            A=(MC_high-MC_fuel)/(MC_high-MC_low)
            y1=-A*(y1max-y1min)+y1max # Oxygen content at 100%
            y2=-A*(y2max-y2min)+y2max   # Oxygen content at 30%
            A_coef=(y1-y2)/(x1-x2)
            B_coef=y1-A_coef*x1
            O2_w_fg_target=A_coef*load_boiler+B_coef
            
        elif MC_fuel>=MC_high:  #Combustible with moisture content between 35 and 50%
            y1=y1max # Oxygen content at 100%
            y2=y2max   # Oxygen content at 30%
            A_coef=(y1-y2)/(x1-x2)
            B_coef=y1-A_coef*x1
            O2_w_fg_target=A_coef*load_boiler+B_coef
            
        Excess_air=(O2_w_fg_target*(m+(n+2*Xw+2*Gamma*K+y+2*Beta*K)/2+z))/(K-O2_w_fg_target*(Gamma*K+Beta*K
+K))

    if REVERSE_ENGINEERING=="ACTIF":
         
        if O2_d_or_w=="O2_on_Wet":
            Excess_air=(O2_fg_target*(m+(n+2*Xw+2*Gamma*K+y+2*Beta*K)/2+z))/(K-O2_fg_target*(Gamma*K+Beta*K
+K))
        elif O2_d_or_w=="O2_on_Dry":
            Excess_air=O2_fg_target*(m+y/2+Beta*K+z)/(K-O2_fg_target*(Beta*K+K))
    
    if influence_excess_air=="ACTIF":
        Excess_air=Excess_air_influence
    #print("Excess_ air: ", Excess_air)
    ##################################
    # based on Combustion Equation
    #################################
```

```
    # CmHnOxNySz+Xw*H2O+(1+e)*K*(O2+Beta*N2+Gamma*H2O)+ X*[(m-CO_f)*CO2+(n+2*Xw+2*Gamma*K*(1+e))/2 *
        # * H2O+(y+2*Beta*(1+e)*K)/2*N2+(e*K+CO_f/2)*O2+z*SO2+CO_f*CO]
    # -->(1+X)*[(m-CO_f)*CO2+(n+2*Xw+2*Gamma*K*(1+e))/2 * H2O+(y+2*Beta*(1+e)*K)/2*N2+(e*K+CO_f/
2)*O2+z*SO2+CO_f*CO]
    # Total Wet
    n_CO2_fg=n_fuel*(m*(1+X))
    n_H2O_fg=n_fuel*(((n+2*Xw+2*Gamma*K*(1+Excess_air))/2)*(1+X))
    n_N2_fg=n_fuel*(((y+2*Beta*(1+Excess_air)*K)/2)*(1+X))
    n_O2_fg=n_fuel*((Excess_air*K)*(1+X))
    n_SO2_fg=n_fuel*(z*(1+X))
    n_tot_w=(n_CO2_fg+n_H2O_fg+n_N2_fg+n_O2_fg+n_SO2_fg)
    n_tot_d=n_tot_w-n_H2O_fg
    m_fg=n_O2_fg*Mm_O2+n_CO2_fg*Mm_CO2+n_H2O_fg*Mm_H2O+n_N2_fg*Mm_N2+n_SO2_fg*Mm_SO2
    # Volumer Percent (Wet)
    CO2_w_fg=n_CO2_fg/n_tot_w
    O2_w_fg=n_O2_fg/n_tot_w
    N2_w_fg=n_N2_fg/n_tot_w
    SO2_w_fg=n_SO2_fg/n_tot_w
    H2O_w_fg=n_H2O_fg/n_tot_w
    # Volume Percent (Dry)
    CO2_d_fg=n_CO2_fg/n_tot_d
    O2_d_fg=n_O2_fg/n_tot_d
    N2_d_fg=n_N2_fg/n_tot_d
    SO2_d_fg=n_SO2_fg/n_tot_d
    compo_d_fg=np.array([CO2_d_fg,O2_d_fg,N2_d_fg,SO2_d_fg])*100  # compo is a vector in volume fraction
    compo_w_fg=np.array([CO2_w_fg,O2_w_fg,N2_w_fg,SO2_w_fg,H2O_w_fg])*100  # compo is a vector in volume
fraction

    ########################
    # Incomplete Combustion 
    # Rem: there is a small approximation by calculated the CO @O2 measured
    # (can be negligated)
    ########################
    # Measured done at 11% O2

#    Load_Boiler_measured=np.array([30,32.6840561744248,33.4709032423904,35.8314444462871,
#                          36.6182915142527,32.2906326404420,32.7627408812214,41.4180586288427,
#                          48.3423128269399,48.7357363609226,55.4239364386301,57.0763152813578,
#                          59.4368564852546,61.0892353279823,61.4826588619651,62.2695059299306,
#                          62.1908212231341,62.0334518095410,63.2924071182859,64.3939930134377,
#                          64.9447859610136,65.4168942017930,65.9676871493689,68.0134895260794,
#                          70.3740307299761,73.4427342950419,73.6787884154316,75.4885366717524,
#                          78.7932943572078,80.3669884931390,78.6359249436147,77.8490778756492,
#                          78.7932943572078,82.0980520426633,84.7733320737463,87.5272968116258,
#                          90.8320544970812,94.1368121825367,96.7334075068231,99.5660569514992])
    
#    CO_measured=np.array([1099,1091.23490546752,1077.18720921216,1098.25875359520,957.781791041620,
#                 803.257132232687,669.804017806790,494.207814614820,557.422447763930,
#                 416.945485210354,206.230041379991,360.754700188924,213.253889507669,
#                 157.063104486239,128.967711975524,114.920015720167,262.420826401421,
#                 374.802396444281,360.754700188924,150.039256358560,128.967711975524,
#                 213.253889507669,220.277737635348,276.468522656779,367.778548316602,
#                 128.967711975524,206.230041379991,143.015408230881,143.015408230881,
#                 206.230041379991,220.277737635348,220.277737635348,255.396978273742,
#                 241.349282018385,241.349282018385,234.325433890706,241.349282018385,
#                 248.373130146063,241.349282018385,248.373130146063])
    
    #f=fit(Load_Boiler_measured,CO_measured,'poly3');
    p1=-0.005327
    p2=1.418
    p3=-122.4
    p4=3648
    CO=p1*(load_boiler*100)**3 + p2*(load_boiler*100)**2 + p3*(load_boiler*100) + p4
    O2_measured=11/100
    O2_fg=O2_d_fg
    Theta=0.21/(1-21)
    Correction=100+(O2_fg-O2_measured)/(Theta/(Theta+1)-O2_fg/100)
    CO=CO*22.41/(Mm_CO)*100/Correction #% ppm
    n_CO_fg=CO*10**(-6)*n_tot_d
```

```
#   n_CO_fg=0
    
    if fuel_type=="NG":
        n_CO_fg=0;

    #######
    # Gas composition after the small correction due to the incomplete combustion.
    #####
    n_tot_w=(n_CO2_fg+n_H2O_fg+n_N2_fg+n_O2_fg+n_SO2_fg+n_CO_fg)
    n_tot_d=n_tot_w-n_H2O_fg
    m_fg=n_O2_fg*Mm_O2+n_CO2_fg*Mm_CO2+n_H2O_fg*Mm_H2O+n_N2_fg*Mm_N2+n_SO2_fg*Mm_SO2+n_CO_fg*Mm_CO
    m_air=n_fuel*(1+Excess_air)*K*(Mm_O2+Beta*Mm_N2+Gamma*Mm_H2O)
    m_H2O=n_H2O_fg*Mm_H2O
    # Volumer Percent (Wet)
    CO2_w_fg=n_CO2_fg/n_tot_w
    O2_w_fg=n_O2_fg/n_tot_w
    N2_w_fg=n_N2_fg/n_tot_w
    SO2_w_fg=n_SO2_fg/n_tot_w
    CO_w_fg=n_CO_fg/n_tot_w
    H2O_w_fg=n_H2O_fg/n_tot_w
    # Volume Percent (Dry)
    CO2_d_fg=n_CO2_fg/n_tot_d
    O2_d_fg=n_O2_fg/n_tot_d
    N2_d_fg=n_N2_fg/n_tot_d
    SO2_d_fg=n_SO2_fg/n_tot_d
    CO_d_fg=n_CO_fg/n_tot_d
    #compo_d_fg=np.array([CO2_d_fg,O2_d_fg,N2_d_fg,SO2_d_fg,CO_d_fg])*100 # compo is a vector in volume fraction
    #compo_w_fg=np.array([CO2_w_fg,O2_w_fg,N2_w_fg,SO2_w_fg,CO_w_fg,H2O_w_fg])*100 # compo is a vector in volume fraction
    compo_db_fg=np.array([CO2_d_fg,O2_d_fg,N2_d_fg]) # compo is a vector in volume fraction
    compo_wb_fg=np.array([CO2_w_fg,O2_w_fg,N2_w_fg,H2O_w_fg]) # compo is a vector in volume fraction
    
    #[N2 O2 CO2  H2O SO2 CO H2 CH4 C2H4 C2H6 C3H8 C4H10 H2S] [m3/m3] 
    #X_fg=np.array([n_N2_fg,n_O2_fg,n_CO2_fg,n_H2O_fg,n_SO2_fg,n_CO_fg,0,0,0,0,0,0,0])/(n_tot_w)
    #print("CO_d_fg: ", CO_d_fg)
    
    return compo_db_fg,compo_wb_fg,CO_d_fg,m_fg,m_air,m_H2O
    

def Boiler(Qdot_boiler,Qdot_boilers_max,MC_fuel,T_fg_IN_option="slope",eta_boiler_method="FIXED_T_BOILER_FG_OUT",
          fuel_type="FUEL_WOOD",wood_type="WOOD_CHIPS_CONTINO_1",
          REVERSE_ENGINEERING="INACTIF",O2_d_or_w="INACTIF",O2_fg_target=None,influence_excess_air="INACTIF"):

# Options   
#eta_boiler_method="FIXED_ETA_BOILER" # "FIXED_T_BOILER_FG_OUT" or "FIXED_ETA_BOILER"
##fuel_type="FUEL_WOOD"        # "FUEL_WOOD" or "FUEL_NG"
#wood_type="WOOD_CHIPS_CONTINO_1"    # "WOOD_CHIPS_CONTINO_1"or "WOOD_CHIPS_CONTINO_2" or "WOOD_CHIPS_HARDWOODS" or "WOOD_CHIPS_SOFTWOODS" or "WOOD_CHIPS_OAK_BARK" or "WOOD_CHIPS_PINE_BARK"
#REVERSE_ENGINEERING="INACTIF" # "ACTIF"
#O2_d_or_w="INACTIF" # "O2_on_Wet":
#O2_d_or_w="INACTIF"  #"O2_on_Dry":
#influence_excess_air="INACTIF" #ACTIF"
    Lv=2501*10**3
    Lv=PropsSI('H','P',P_ref,'Q',1,'Water')-PropsSI('H','P',P_ref,'Q',0,'Water')
    P_fg=101325

    load_boiler=Qdot_boiler/Qdot_boilers_max
    T_fg=a_temp*load_boiler+b_temp #linear correlation from Mermoud et al.
    T_fg_IN_option=T_fg
    # Flue Gas Composition
#   print("load_boiler",load_boiler)
    [compo_db_fg,compo_wb_fg,CO_db_fg,m_fg,m_air,m_H2O]=FG_Composition(load_boiler,MC_fuel,fuel_type,wood_type,REVERSE_ENGINEERING,O2_d_or_w,O2_fg_target,influence_excess_air)
    [W_MC_fuel_daf,W_ash_fuel_daf,W_MC_fuel_ad,W_ash_fuel_ad,NCV_Wood_ad,NCV_Wood_daf,HHV_Wood_daf]=fuel_properties(MC_fuel,"fuel_properties",fuel_type,wood_type)

    CO2_db_fg=compo_db_fg[0]
```

```
    
    # Flow of Thermal Energy
    # Q thermal loss by radiation, convection and conduction:
    # Qwalls [?] is the thermal loss by radiation, convection and conduction and depends on the temperature of the boiler walls. 
    # Generally a fixed value is assumed since the walls temperature ? and thus the heat losses are almost the same over the entire load range of the boilers. 
    # As an example, Good et al. (Good et al. 2006) consider walls losses as 2 % of the full load.
    Qdot_loss_walls=0.03*Qdot_boilers_max  # voir autre methode qui est decrite dans "data"
    # Q chemical loss due to incomplete combustion
    # Qunburnt [?] is the chemical loss due to incomplete combustion of the wood, including solid and gaseous unburnt products; in the following, the loss due to
    # solid unburnt products is neglected (<0.1 % according to spot ash analyses); the loss due to gaseous unburnt products is estimated following Eq. 3 proposed by Good et al. (Good et al. 2006):
    H_db=MC_fuel/100/(1-MC_fuel/100);
    Qdot_loss_unburnt=CO_db_fg/(CO2_db_fg+CO_db_fg)*11800/(NCV_Wood_daf*10**(-3)/100-0.2442*H_db);  #[-]
    #print("Qdot_loss_unburnt ", Qdot_loss_unburnt)
    # Le taux d?imbr˚lÈs dans les Èchantillons de cendres est faible (<4% de la masse de cendres).
    # Si on considËre que le bois sec prÈsente un taux de cendres de l'ordre de 2%, le taux d'imbr˚lÈs sur bois sec sera infÈrieur ‡ 0.1% : nous l?avons nÈgligÈ dans la suite des calculs.
    # The efficiency ? is calculated following: 
    prop_fg=MyProperties_Gas_2(compo_db_fg,compo_wb_fg[3],T_fg,P_fg)    
    #gas1.cp_mass
    #c=gas1.thermal_conductivity
    #d=gas1.viscosity
    cp_Mean_fg=prop_fg.cp 
    
    cp_air_combustion=PropsSI("Cpmass","T",T_air_combustion,"P",P_ref,"Air")
    
    if eta_boiler_method=="FIXED_ETA_BOILER":
        eta_boiler=0.85 # Hypothese
        mdot_air_ad=m_air/(1+W_MC_fuel_daf/100+W_ash_fuel_daf/100)  # kgair/kg ad
        mdot_fuel_ad=Qdot_boiler/(eta_boiler*(NCV_Wood_ad+mdot_air_ad*cp_air_combustion*20)) ##20??????? kg_fuel_ad/s
        mdot_fuel_daf=mdot_fuel_ad*(1-W_MC_fuel_ad/100-W_ash_fuel_ad/100)
        mdot_tot_fg=m_fg*mdot_fuel_daf
        mdot_MC_tot=m_H2O*mdot_fuel_daf
        Qdot_tot=mdot_fuel_ad*(NCV_Wood_ad+mdot_air_ad*cp_air_combustion*(T_air_combustion-T_ref))
        T_fg_IN=(Qdot_tot-Qdot_loss_unburnt-Qdot_loss_walls-Qdot_boiler)/(cp_Mean_fg*mdot_tot_fg)+T_ref
        #print ("T_fg_IN 1: ",T_fg_IN)
        #cp_w_DH=XSteam('cp_ph',P_w_DH*10^-5,h)*10^3                    # [J/kg/K]cp_cw
        Qdot_tot_HHV=mdot_fuel_daf*(HHV_Wood_daf+m_air*cp_air_combustion*(T_air_combustion-T_ref))
        T_fg_IN=(Qdot_tot_HHV-Qdot_loss_unburnt-Qdot_loss_walls-Qdot_boiler-mdot_MC_tot*Lv)/(cp_Mean_fg*mdot_tot_fg)+T_ref
        #print ("T_fg_IN 2: ",T_fg_IN)
        eta_HHV=Qdot_boiler/Qdot_tot_HHV
        
        prop_fg.update_T_P(T_fg_IN,P_fg)
        
    if eta_boiler_method=="FIXED_T_BOILER_FG_OUT":
        T_fg_IN=T_fg_IN_option
        eta_boiler=0.6
        eta_boiler_min=0.6
        eta_boiler_max=0.99
        T_fg_IN_iter=0;
        while abs(T_fg_IN_iter-T_fg_IN) >0.0001:
            mdot_air_ad=m_air/(1+W_MC_fuel_daf/100+W_ash_fuel_daf/100)  # kgair/kg ad
            mdot_fuel_ad=Qdot_boiler/(eta_boiler*(NCV_Wood_ad+mdot_air_ad*cp_air_combustion*20))
            mdot_fuel_daf=mdot_fuel_ad*(1-W_MC_fuel_ad/100-W_ash_fuel_ad/100)
            mdot_tot_fg=m_fg*mdot_fuel_daf           
            mdot_tot_fg=mdot_fuel_ad+mdot_fuel_ad*mdot_air_ad            
            mdot_H2O_tot=m_H2O*mdot_fuel_daf
            Qdot_tot=mdot_fuel_ad*(NCV_Wood_ad+mdot_air_ad*cp_air_combustion*(T_air_combustion-T_ref))
            T_fg_IN_iter=(Qdot_tot-Qdot_loss_unburnt-Qdot_loss_walls-Qdot_boiler)/(cp_Mean_fg*mdot_tot_fg)+T_ref
            Qdot_tot_HHV=mdot_fuel_daf*(HHV_Wood_daf+m_air*cp_air_combustion*(T_air_combustion-T_ref))
            T_fg_IN_iter=(Qdot_tot_HHV-Qdot_loss_unburnt-Qdot_loss_walls-Qdot_boiler-mdot_H2O_tot*Lv)/(cp_Mean_fg*mdot_tot_fg)+T_ref
            Eta_HHV=Qdot_boiler/Qdot_tot_HHV
            if T_fg_IN_iter>T_fg_IN:
```

```
                eta_boiler_min=eta_boiler
                eta_boiler=(eta_boiler_min+eta_boiler_max)/2
            else:
                eta_boiler_max=eta_boiler
                eta_boiler=(eta_boiler_min+eta_boiler_max)/2
        prop_fg.update_T_P(T_fg_IN,P_fg)
    mdot_nc=(1-prop_fg.X_H2O)*mdot_tot_fg
#    print("eta_boiler",eta_boiler)        
#    prop_fg.update_mdot_nc(mdot_nc)
#    print("Qdot_tot",Qdot_tot,"Qdot_loss_unburnt",Qdot_loss_unburnt,"Qdot_loss_walls",Qdot_loss_walls)
    return prop_fg,mdot_nc,mdot_tot_fg,mdot_fuel_ad,Qdot_tot,Qdot_tot_HHV,eta_boiler,



def Boilers(Qdot_boiler_max_1,Qdot_boiler_max_2,Qdot_boilers,MC_fuel,eta_boiler_method="FIXED_T_BOILER_FG_OUT",T_fg_IN_option=180+273.15,
          fuel_type="FUEL_WOOD",wood_type="WOOD_CHIPS_CONTINO_1",
          REVERSE_ENGINEERING="INACTIF",O2_d_or_w="INACTIF",O2_fg_target=None,influence_excess_air="INACTIF"):

# Options   
#eta_boiler_method="FIXED_ETA_BOILER" # "FIXED_T_BOILER_FG_OUT" or "FIXED_ETA_BOILER"
##fuel_type="FUEL_WOOD"        # "FUEL_WOOD" or "FUEL_NG"
#wood_type="WOOD_CHIPS_CONTINO_1"    # "WOOD_CHIPS_CONTINO_1"or "WOOD_CHIPS_CONTINO_2" or "WOOD_CHIPS_HARDWOODS" or "WOOD_CHIPS_SOFTWOODS" or "WOOD_CHIPS_OAK_BARK" or "WOOD_CHIPS_PINE_BARK"
#REVERSE_ENGINEERING="INACTIF" # "ACTIF"
#O2_d_or_w="INACTIF" # "O2_on_Wet":
#O2_d_or_w="INACTIF"  #"O2_on_Dry":
#influence_excess_air="INACTIF" #ACTIF"
    
    T_fg=T_fg_IN_option
    P_fg=101325

    Qdot_boilers_min=0.3*min(Qdot_boiler_max_1,Qdot_boiler_max_2)
    Qdot_boilers_max=Qdot_boiler_max_1+Qdot_boiler_max_2
    Qdot_boiler_smallest_max=min(Qdot_boiler_max_1,Qdot_boiler_max_2)
    Qdot_boiler_biggest_max=max(Qdot_boiler_max_1,Qdot_boiler_max_2)
    if Qdot_boilers < Qdot_boilers_min:
        [prop_fg,mdot_nc_fg,mdot_tot_fg,mdot_fuel_ad,Qdot_tot,Qdot_tot_HHV]=Boiler(Qdot_boilers,Qdot_boiler_smallest_max,MC_fuel,eta_boiler_method=eta_boiler_method,T_fg_IN_option=T_fg_IN_option,
          fuel_type=fuel_type,wood_type=wood_type,REVERSE_ENGINEERING=REVERSE_ENGINEERING,O2_d_or_w=O2_d_or_w,O2_fg_target=None,influence_excess_air=influence_excess_air)
    
    elif Qdot_boilers >= Qdot_boilers_min and Qdot_boilers < Qdot_boilers_min/0.3:
        [prop_fg,mdot_nc_fg,mdot_tot_fg,mdot_fuel_ad,Qdot_tot,Qdot_tot_HHV]=Boiler(Qdot_boilers,Qdot_boiler_smallest_max,MC_fuel,eta_boiler_method=eta_boiler_method,T_fg_IN_option=T_fg_IN_option,
          fuel_type=fuel_type,wood_type=wood_type,REVERSE_ENGINEERING=REVERSE_ENGINEERING,O2_d_or_w=O2_d_or_w,O2_fg_target=None,influence_excess_air=influence_excess_air)
    
    elif Qdot_boilers >= Qdot_boiler_smallest_max and Qdot_boilers < Qdot_boiler_biggest_max:
        [prop_fg,mdot_nc_fg,mdot_tot_fg,mdot_fuel_ad,Qdot_tot,Qdot_tot_HHV]=Boiler(Qdot_boilers,Qdot_boiler_biggest_max,MC_fuel,eta_boiler_method=eta_boiler_method,T_fg_IN_option=T_fg_IN_option,
          fuel_type=fuel_type,wood_type=wood_type,REVERSE_ENGINEERING=REVERSE_ENGINEERING,O2_d_or_w=O2_d_or_w,O2_fg_target=None,influence_excess_air=influence_excess_air)
    
    elif Qdot_boilers >= Qdot_boiler_biggest_max and Qdot_boilers < 0.7*Qdot_boilers_max:
        Qdot_boiler_biggest=Qdot_boiler_biggest_max*0.7
        Qdot_boiler_smallest=Qdot_boilers-Qdot_boiler_biggest
        
        [prop_fg_biggest,mdot_nc_biggest,mdot_tot_fg_biggest,mdot_fuel_ad_biggest,Qdot_tot_biggest,Qdot_tot_HHV_biggest]=Boiler(Qdot_boiler_biggest,Qdot_boiler_biggest_max,MC_fuel,eta_boiler_method=eta_boiler_method,T_fg_IN_option=T_fg_IN_option,
          fuel_type=fuel_type,wood_type=wood_type,REVERSE_ENGINEERING=REVERSE_ENGINEERING,O2_d_or_w=O2_d_or_w,O2_fg_target=None,influence_excess_air=influence_excess_air)
        [prop_fg_smallest,mdot_nc_smallest,mdot_tot_fg_smallest,mdot_fuel_ad_smallest,Qdot_tot_smallest,Qdot_tot_HHV_smallest]=Boiler(Qdot_boiler_smallest,Qdot_boiler_smallest_max,MC_fuel,eta_boiler_method=eta_boiler_method,T_fg_IN_option=T_fg_IN_option,
          fuel_type=fuel_type,wood_type=wood_type,REVERSE_ENGINEERING=REVERSE_ENGINEERING,O2_d_or_w=O2_d_or_w,O2_fg_target=None,influence_excess_air=influence_excess_air)
```

```python
        mdot_nc_fg=mdot_nc_biggest+mdot_nc_smallest
        mdot_tot_fg=mdot_tot_fg_biggest+mdot_tot_fg_smallest
        T_fg_IN=(prop_fg_biggest.T*mdot_tot_fg_biggest+prop_fg_smallest.T*mdot_tot_fg_smallest)/mdot_tot_fg
        
        compo_db_fg=(prop_fg_biggest.compo_db*mdot_tot_fg_biggest +prop_fg_smallest.compo_db*mdot_tot_fg_smallest)/mdot_tot_fg
        compo_wb_fg=(prop_fg_biggest.compo_wb*mdot_tot_fg_biggest +prop_fg_smallest.compo_wb*mdot_tot_fg_smallest)/mdot_tot_fg
        prop_fg=MyProperties_Gas_2(compo_db_fg,compo_wb_fg[3],T_fg_IN,P_fg)
        mdot_fuel_ad=mdot_fuel_ad_smallest+ mdot_fuel_ad_biggest
        Qdot_tot=Qdot_tot_smallest+Qdot_tot_biggest
        Qdot_tot_HHV=Qdot_tot_HHV_smallest+Qdot_tot_HHV_biggest
    
    elif Qdot_boilers >= 0.7*Qdot_boilers_max and Qdot_boilers <= Qdot_boilers_max*1.4:
        Qdot_boiler_smallest=Qdot_boilers/Qdot_boilers_max*Qdot_boilers_min/0.3
        Qdot_boiler_biggest=Qdot_boilers/Qdot_boilers_max*(Qdot_boilers_max-Qdot_boilers_min/0.3)
        
        [prop_fg_biggest,mdot_nc_biggest,mdot_tot_fg_biggest,mdot_fuel_ad_biggest,Qdot_tot_biggest,Qdot_tot_HHV_biggest]=Boiler(Qdot_boiler_biggest,Qdot_boiler_biggest_max,MC_fuel,eta_boiler_method=eta_boiler_method,T_fg_IN_option=T_fg_IN_option,
           fuel_type=fuel_type,wood_type=wood_type,REVERSE_ENGINEERING=REVERSE_ENGINEERING,O2_d_or_w=O2_d_or_w,O2_fg_target=None,influence_excess_air=influence_excess_air)
        [prop_fg_smallest,mdot_nc_smallest,mdot_tot_fg_smallest,mdot_fuel_ad_smallest,Qdot_tot_smallest,Qdot_tot_HHV_smallest]=Boiler(Qdot_boiler_smallest,Qdot_boiler_smallest_max,MC_fuel,eta_boiler_method=eta_boiler_method,T_fg_IN_option=T_fg_IN_option,
           fuel_type=fuel_type,wood_type=wood_type,REVERSE_ENGINEERING=REVERSE_ENGINEERING,O2_d_or_w=O2_d_or_w,O2_fg_target=None,influence_excess_air=influence_excess_air)
        mdot_nc_fg=mdot_nc_biggest+mdot_nc_smallest
        mdot_tot_fg=mdot_tot_fg_biggest+mdot_tot_fg_smallest
        T_fg_IN=(prop_fg_biggest.T*mdot_tot_fg_biggest+prop_fg_smallest.T*mdot_tot_fg_smallest)/mdot_tot_fg
        
        compo_db_fg=(prop_fg_biggest.compo_db*mdot_tot_fg_biggest +prop_fg_smallest.compo_db*mdot_tot_fg_smallest)/mdot_tot_fg
        compo_wb_fg=(prop_fg_biggest.compo_wb*mdot_tot_fg_biggest +prop_fg_smallest.compo_wb*mdot_tot_fg_smallest)/mdot_tot_fg
        prop_fg=MyProperties_Gas_2(compo_db_fg,compo_wb_fg[3],T_fg_IN,P_fg)
        mdot_fuel_ad=mdot_fuel_ad_smallest+ mdot_fuel_ad_biggest
        Qdot_tot=Qdot_tot_smallest+Qdot_tot_biggest
        Qdot_tot_HHV=Qdot_tot_HHV_smallest+Qdot_tot_HHV_biggest

    elif Qdot_boilers > Qdot_boilers_max*1.4:
        print ('Q_load > Qdot_boilers_max*1.4')
        

    return prop_fg,mdot_nc_fg,mdot_tot_fg,mdot_fuel_ad,Qdot_tot,Qdot_tot_HHV
```

B.4 Dryer

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 11:12:51 2020

@author: salva
"""
import numpy as np
from MyProperties_Gas import MyProperties_Gas_2
from Constants import T_Fuel_Ad,P_ref,Coef_losses,compo_db_air,Ratio_H2O_air,T_air_mixer
from Constants import T_FG_dryer,MCfin_min_ad
from Boiler2 import Boiler,fuel_properties
from scipy.optimize import brentq

from Mixer import It_Mass_Air,Mix

from GetProps import Get_MC_FG,Get_hFuel,Get_hFG_IN,Get_TandMC_FG_OUT,Get_MC_Fuel
# =============================================================================
"Nomenclature"
# FG   --> Flue Gas
# Fuel --> Fuel
# ini --> Initial conditions || Without Dryer
# nom/nominal --> Condition of Full Load || Nominal rate conditions
# daf,db,ad   --> dry ash free, dry basis and as delivered basis, respectively

# Mass --> Mass Flow, [kg/s]
# MC   --> Moisture Content, [%] for Fuel, [-] for FG
# T    --> Temperature, [K]
# P    --> Power | Demand, [W]
# ETA  --> Efficiency, [-]
# me   --> Evaporation rate, [kg/s]
# Heat_exc --> Heat Exchanged in Dryer, [kW]
# LHV  --> Low Heat Value of Fuel, [J/kgx] where x could be daf,db,ad
# HHV  --> High Heat Value of Fuel, [J/kgx] where x could be daf,db,ad

# Tau --> Running time array, [h]
# Hours --> number of hours that a Load is repeated, [h]
# Load --> Curve Demand, [W]

# Aconveyor --> Conveyor surface of the Dryer, [m2]
# Rho_Average --> Average density of FG inlet/outlet of the Dryer,[kg/m3].!! CURRENTLY: rhoFGin

# NPV --> Net Present Value, [k€]
# RT  --> Payback period, Return time [year]

"Identation"
# Fuel 1 --> Inlet of the Dryer
# Fuel 2 --> Outlet of the Dryer || Inlet of the Boiler
# FG 1   --> Outlet of the Boiler
# FG 2   --> Inlet of the Dryer
# FG 3   --> Outlet of the Dryer
# =============================================================================

def It_FuelMC_InBoiler(MC_Fuel_2_ad,MC_Fuel_1_ad,P_Boiler,P_Boiler_nominal,
                       PartialMass_2_db,Mass_FG_2_db_max):

    "BOILER"
    #Total Dried Fuel:
    [FG_2,Mass_FG_1_db_nom,FG_OutBoiler_Mass_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,ETA_boiler
     ]=Boiler(P_Boiler_nominal,P_Boiler_nominal,0)

    [FG_2,Mass_FG_1_db,FG_OutBoiler_Mass_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,ETA_boiler
     ]=Boiler(P_Boiler,P_Boiler_nominal,0)

#    Mass_FG_1_db_TotalDried=Mass_FG_1_db
#    Mass_FG_2_db_TotalDried=Mass_FG_1_db_TotalDried*PartialMass_2_db

    #MC_Fuel_2_ad
    [FG_2,Mass_FG_1_db_nom,FG_OutBoiler_Mass_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,ETA_boiler
     ]=Boiler(P_Boiler_nominal,P_Boiler_nominal,MC_Fuel_2_ad)

    [FG_2,Mass_FG_1_db,FG_OutBoiler_Mass_tot,Fuel_Mass_tot,Qdot_tot,Qdot_tot_HHV,ETA_boiler
     ]=Boiler(P_Boiler,P_Boiler_nominal,MC_Fuel_2_ad)

    #fuel properties
    [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,
     HHV_Fuel_daf]=fuel_properties(MC_Fuel_2_ad,"fuel_properties")
    Mass_Fuel_daf=Fuel_Mass_tot*(1-MC_Fuel_2_ad/100-Ash_Fuel_ad/100)

    Load=P_Boiler/P_Boiler_nominal
    Mass_FG_2_db_nom=PartialMass_2_db*Mass_FG_1_db_nom

```

```
        
    if Load ==1:
        Mass_FG_2_db=min(Mass_FG_2_db_nom,Mass_FG_2_db_max)
    else:
        
        if Mass_FG_1_db>=Mass_FG_2_db_nom:
            
            Mass_FG_2_db=min(Mass_FG_2_db_nom,Mass_FG_2_db_max)
        else:
            Mass_FG_2_db=min(Mass_FG_1_db,Mass_FG_2_db_max)
                

#    "MIXER"
#    T_FG_2=T_FG_dryer
#    Air=MyProperties_Gas_2(compo_db_air,Ratio_H2O_air/(1+Ratio_H2O_air),T_air_mixer,P_ref)
#    fun= lambda Mass_Air_db_it : It_Mass_Air(Mass_Air_db_it,Mass_FG_2_db,Air,FG_2,T_FG_21)
#    Mass_Air_db=0
#    if T_FG_21== T_FG_boiler:
#        FG_21=MyProperties_Gas_2(FG_2.compo_db,FG_2.y_H2O,FG_2.T,P_ref)
#        Mass_FG_21_db=Mass_FG_2_db
#    else:
#        Mass_Air_db=brentq(fun,0,10*Mass_FG_2_db)
#        [FG_21,Mass_FG_21_db]=Mix(Mass_Air_db,Mass_FG_2_db,Air,FG_2,T_FG_21)
    
    "DRYER"
    "____FG____"
    MC_FG_2_db=Get_MC_FG(FG_2)
    "____FUEL IN____"
    MC_Fuel_1_daf=Get_MC_Fuel(MC_Fuel_1_ad) #[%daf]
    T_Fuel_1=T_Fuel_Ad
    htot_Fuel_1_db=Get_hFuel(T_Fuel_1,MC_Fuel_1_daf)
    "____FG IN____"
    htot_FG_2_db=Get_hFG_IN(FG_2) #kJ/kgdaf
    "____FUEL OUT____"
    MC_Fuel_2_daf=Get_MC_Fuel(MC_Fuel_2_ad) #[%daf]
    #From linear correlation between wood temperature and T_FG_2:
    T_Fuel_2=(MC_Fuel_2_daf/MC_Fuel_1_daf)*(T_Fuel_1-FG_2.T)+FG_2.T
    htot_Fuel_2_db=Get_hFuel(T_Fuel_2,MC_Fuel_2_daf)
    "____FG OUT____"
    #Energy Balance
    #if PartialFG=0
    MC_FG_3_db=MC_FG_2_db
    htot_FG_3_db=htot_FG_2_db
    if PartialMass_2_db != 0:
        htot_FG_3_db=htot_FG_2_db-(Mass_Fuel_daf*(htot_Fuel_2_db-htot_Fuel_1_db)/
                                    (Mass_FG_2_db*(1-Coef_losses)))
#        print("htot_FG_3_db_2",htot_FG_3_db)

        htot_FG_3_db=max(20,htot_FG_3_db)
#        print("htot_FG_3_db",htot_FG_3_db)
        [T_FG_3,MC_FG_3_db,Y_H2O_OUT]=Get_TandMC_FG_OUT(FG_2,htot_FG_3_db)
    #Mass Balance
    #if PartialFG=0
    MC_Fuel_2_daf_2=MC_Fuel_1_daf
    MC_Fuel_2_ad_2=MC_Fuel_2_ad
    if PartialMass_2_db != 0:
        MC_Fuel_2_daf_2=(MC_Fuel_1_daf/100-Mass_FG_2_db*(MC_FG_3_db-MC_FG_2_db)/
                         Mass_Fuel_daf)*100
                        
        MC_Fuel_2_ad_2=MC_Fuel_2_daf_2 / (1+MC_Fuel_2_daf_2/100+Ash_Fuel_daf/100)
    res=MC_Fuel_2_ad - MC_Fuel_2_ad_2

    return res

def Dryer(FG_2,MC_Fuel_1_ad,MC_Fuel_2_ad,Mass_Fuel_daf,Mass_FG_1_db,PartialMass_2_db, Mass_FG_2_db_nom,Mass_FG_2_db_max):
    
    if Mass_FG_1_db>=Mass_FG_2_db_nom:
        Mass_FG_2_db=min(Mass_FG_2_db_nom,Mass_FG_2_db_max)
        Mass_FG_2_db=Mass_FG_2_db_nom
    else:
        Mass_FG_2_db=min(Mass_FG_1_db,Mass_FG_2_db_max)

#    Mass_FG_2_db=PartialMass_2_db*Mass_FG_1_db
#    "MIXER"
#    T_FG_21=T_FG_dryer
#    Air=MyProperties_Gas_2(compo_db_air,Ratio_H2O_air/(1+Ratio_H2O_air),T_air_mixer,P_ref)
#    fun= lambda Mass_Air_db_it : It_Mass_Air(Mass_Air_db_it,Mass_FG_2_db,Air,FG_2,T_FG_21)
#    Mass_Air_db=0
#    if T_FG_21== T_FG_boiler:
#        FG_21=MyProperties_Gas_2(FG_2.compo_db,FG_2.y_H2O,FG_2.T,P_ref)
#        Mass_FG_21_db=Mass_FG_2_db
```

```
#   else:
#       Mass_Air_db=brentq(fun,0,10*Mass_FG_2_db)
#       [FG_21,Mass_FG_21_db]=Mix(Mass_Air_db,Mass_FG_2_db,Air,FG_2,T_FG_21)

    "____FG____"
    MC_FG_2_db=Get_MC_FG(FG_2)
    "____FUEL IN____"
    MC_Fuel_1_daf=Get_MC_Fuel(MC_Fuel_1_ad) #[%daf]
    T_Fuel_1=T_Fuel_Ad
    htot_Fuel_1_db=Get_hFuel(T_Fuel_1,MC_Fuel_1_daf) #kJ/kg
    "____FG IN____"
    htot_FG_2_db=Get_hFG_IN(FG_2) #kJ/kg
    "____FUEL OUT____"
    MC_Fuel_2_daf=Get_MC_Fuel(MC_Fuel_2_ad) #[%daf]
    #From linear correlation between wood temperature and T_FG_2:
    T_Fuel_2=(MC_Fuel_2_daf/MC_Fuel_1_daf)*(T_Fuel_1-FG_2.T)+FG_2.T
    htot_Fuel_2_db=Get_hFuel(T_Fuel_2,MC_Fuel_2_daf) #kW
    "____FG OUT____"
    #Energy Balance
    #if PartialFG=0
    MC_FG_3_db=MC_FG_2_db
    htot_FG_3_db=htot_FG_2_db
    FG_3=MyProperties_Gas_2(FG_2.compo_db,FG_2.y_H2O,FG_2.T,P_ref)
    if PartialMass_2_db != 0:
        htot_FG_3_db=htot_FG_2_db-(Mass_Fuel_daf*(htot_Fuel_2_db-htot_Fuel_1_db)/
                                    (Mass_FG_2_db*(1-Coef_losses)))
        [T_FG_3,MC_FG_3_db,Y_H2O_OUT]=Get_TandMC_FG_OUT(FG_2,htot_FG_3_db)
        FG_3=MyProperties_Gas_2(FG_2.compo_db,Y_H2O_OUT,T_FG_3,P_ref)
    Heat_exchanged=Mass_FG_2_db*(htot_FG_2_db-htot_FG_3_db) #kW
    me=(MC_FG_3_db-MC_FG_2_db)*Mass_FG_2_db #Evaporation rate kgH2O/s
    
    return FG_3,FG_2,Mass_FG_2_db,T_Fuel_2,me,Heat_exchanged

def It_MassFGTotalDried(Mass_FG_2_db,Mass_Fuel_daf,FG_2,MC_Fuel_1_ad):
    MC_Fuel_2_ad=MCfin_min_ad
    "Total Dried conditions"
#   "MIXER"
#   T_FG_21=T_FG_dryer
#   Air=MyProperties_Gas_2(compo_db_air,Ratio_H2O_air/(1+Ratio_H2O_air),T_air_mixer,P_ref)
#   fun= lambda Mass_Air_db_it : It_Mass_Air(Mass_Air_db_it,Mass_FG_2_db,Air,FG_2,T_FG_21)
#   Mass_Air_db=0
#   if T_FG_21== T_FG_boiler:
#       FG_21=MyProperties_Gas_2(FG_2.compo_db,FG_2.y_H2O,FG_2.T,P_ref)
#       Mass_FG_21_db=Mass_FG_2_db
#   else:
#       Mass_Air_db=brentq(fun,0,10*Mass_FG_2_db)
#       [FG_21,Mass_FG_21_db]=Mix(Mass_Air_db,Mass_FG_2_db,Air,FG_2,T_FG_21)

    "____FG____"
    MC_FG_2_db=Get_MC_FG(FG_2)
    "____FUEL IN____"
    MC_Fuel_1_daf=Get_MC_Fuel(MC_Fuel_1_ad) #[%daf]
    T_Fuel_1=T_Fuel_Ad
    htot_Fuel_1_db=Get_hFuel(T_Fuel_1,MC_Fuel_1_daf)
    "____FG IN____"
    htot_FG_2_db=Get_hFG_IN(FG_2)
    "____FUEL OUT____"
    MC_Fuel_2_daf=Get_MC_Fuel(MC_Fuel_2_ad) #[%daf]
    #From linear correlation between wood temperature and T_FG_2:
    T_Fuel_2=(MC_Fuel_2_daf/MC_Fuel_1_daf)*(T_Fuel_1-FG_2.T)+FG_2.T 
    htot_Fuel_2_db=Get_hFuel(T_Fuel_2,MC_Fuel_2_daf)
    "____FG OUT____"
    MC_FG_3_db=MC_FG_2_db
    htot_FG_3_db=htot_FG_2_db 
    #Energy Balance
    htot_FG_3_db=htot_FG_2_db-(Mass_Fuel_daf*(htot_Fuel_2_db-htot_Fuel_1_db)/(
            Mass_FG_2_db*(1-Coef_losses)))
#   print("htot_FG_3_db",htot_FG_3_db)
#
    htot_FG_3_db=max(10,htot_FG_3_db)

    [T_FG_3,MC_FG_3_db,Y_H2O_OUT]=Get_TandMC_FG_OUT(FG_2,htot_FG_3_db)
    #Mass Balance
    Mass_FG_2_db_2=Mass_Fuel_daf*(MC_Fuel_1_daf/100-MC_Fuel_2_daf/100)/(MC_FG_3_db-MC_FG_2_db)
#   Mass_FG_2_db_2=Mass_FG_2_db_2-Mass_Air_db
    
    res=Mass_FG_2_db-Mass_FG_2_db_2
    
    MC_Fuel_2_daf_2=100*(MC_Fuel_1_daf/100-Mass_FG_2_db*(MC_FG_3_db-MC_FG_2_db)/Mass_Fuel_daf)

    res=MC_Fuel_2_daf-MC_Fuel_2_daf_2
```

```
    return res
```

B.5 Thermal calculations

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 26 11:19:58 2020

@author: salva
"""
import numpy as np

#sys.path.append('C:/Users/marta/Desktop/Python/MartaCastaneda_MasterThesis/Properties')
from CoolProp.CoolProp import PropsSI
from Constants import Mm_CO2,Mm_N2,Mm_O2,Mm_H2O,T_ref,P_ref,R,HR_FG_OUT_max, compo_db_air
from scipy.optimize import fminbound,brentq,brenth,bisect,ridder,brent
from Boiler2 import fuel_properties
from MyProperties_Gas import MyProperties_Gas_2
# =============================================================================
# # Water vapour content in the FG
# Retur of the absolute humidity
# =============================================================================
def Get_MC_FG(FG):
    #Molar fraction wb
    Y_H2O_IN=FG.y_H2O
    #Molar fraction db
    X_CO2=FG.compo_db[0]              #[mol CO2/mol dry fg]
    X_N2=FG.compo_db[2]               #[mol N2/mol dry fg]    
    X_O2=FG.compo_db[1]               #[mol O2/mol dry fg] Fire hazards if higher than 10%  
    X_H2O_IN=Y_H2O_IN/(1-Y_H2O_IN)    # mol H20/mol dry fg
    #Molar mass db and wb
    Mm_FG_dry=X_CO2*(Mm_CO2)+X_N2*(Mm_N2)+X_O2*(Mm_O2)      #[kg FG/kmolFGdaf]
#    Mm_FG_wet_in=Y_CO2_IN*(Mm_CO2)+Y_N2_IN*(Mm_N2)+Y_O2_IN*(Mm_O2)+Y_H2O_IN*(Mm_H2O)
    #Weight Moisture contents FG
    MC_FG_2_db=X_H2O_IN*Mm_H2O/Mm_FG_dry                    #[kg H2O/kg FG db]
    #Weight FG content in total FG
    return MC_FG_2_db
# =============================================================================
# # Fuel Moisture contents
# Return of the fuel humidity in [%_daf]
# =============================================================================
def Get_MC_Fuel(MC_Fuel_ad):
    #fuel properties
    [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,
     HHV_Fuel_daf]=fuel_properties(MC_Fuel_ad,"fuel_properties")
    MC_Fuel_daf=MC_Fuel_ad/(1-MC_Fuel_ad/100-Ash_Fuel_ad/100) #Wood moisture daf
    return MC_Fuel_daf
# =============================================================================
# # Fuel Enthalpy
# Return the enthalpy of the fuel in [kJ/kg Fuel daf]
# =============================================================================
def Get_hFuel(T,MC_Fuel_daf):
    #Properties of Water in the Fuel
    Cp_WaterFuel_IN=PropsSI("CPMASS","T",T,"Q",0,"Water")/1000 #kJ/kgK
    #Properties of Fuel without water
#    Cp_Fuel_IN=266*10**-3 + 1.16*10**-3*(T-273.15) # Basu, from De Fusco
    Cp_Fuel_IN1= 5.46*T-524.77 #Softwood Gupta et al 2003
    Cp_Fuel_IN2= 3.69*T+231.06 #Sfotwood bark Gupta et al 2003
    Cp_Fuel_IN=(Cp_Fuel_IN1+Cp_Fuel_IN2)/2 /1000 #kJ/kgFueldaf
    h_Fuel_IN=Cp_Fuel_IN*(T-T_ref)
    #Entalphy of the whole fuel
    htot_Fuel_IN_db=h_Fuel_IN+(MC_Fuel_daf/100)*Cp_WaterFuel_IN*(T - T_ref)    #[kJ/kg Fuel daf]
    return htot_Fuel_IN_db
# =============================================================================
# # FG enthalpy at the inlet of the dryer
# Return the enthlpy of the Fg in in [kJ/kg Fuel daf]
# =============================================================================
def Get_hFG_IN(FG_2):
    T=FG_2.T
    MC_FG_2_db=Get_MC_FG(FG_2)
    #Water vapor on FG
    P_SatVapFG_IN=PropsSI("P", "T", T, "Q", 0,"Water")
    P_v=FG_2.y_H2O*P_SatVapFG_IN
    Cp_VapourFG_IN=PropsSI('CPMASS','P',P_v,'T',T,'Water')/1000
    H_V=PropsSI('H','P',max(P_v,611.655),'Q',1,'Water')/1000  
    H_L=PropsSI('H','P',max(P_v,611.655),'Q',0,'Water')/1000
    #611.655 is the min value acceptable by CoolProps
    Lv_vap_IN=H_V - H_L                                 #Water latent value [kJ/kg]
    h_vap_fg_in=(Lv_vap_IN+Cp_VapourFG_IN*(T-T_ref))    #[kJ/kg H2O]
    #Cp just FG dried
    Cp_FG_IN=FG_2.cp_db/1000                            #[kJ/kg K]
    #Entalphy of just FG (without water) db
    h_FG_IN=Cp_FG_IN*(T-T_ref)                          #[kJ/kg FG db]
    #Total Entalphy of the whole FG
    htot_FG_IN_db=h_FG_IN+MC_FG_2_db*h_vap_fg_in        #[kJ/kg FG db]
```

```
#   _____
#    htot_FG_IN_db=1*(T-T_ref)+MC_FG_2_db*(2501+1.805*(T-T_ref))
#   _____
    return htot_FG_IN_db

def Get_TandMC_FG_OUT(FG,h_tot):
    compo_db_FG=FG.compo_db
    T_FG_IN=FG.T
    #Molar fraction db
    X_CO2=compo_db_FG[0]   #[mol CO2/mol dry fg]
    X_N2=compo_db_FG[2]    #[mol N2/mol dry fg]
    X_O2=compo_db_FG[1]    #[mol O2/mol dry fg] Fire hazards if higher than 10%
    Mm_FG_db=X_CO2*(Mm_CO2)+X_N2*(Mm_N2)+X_O2*(Mm_O2)
    T_min=1+273.153
    T_max=99+273.153
    fun=lambda T : Iteration_TandM(T,h_tot,compo_db_FG)
#    T=fminbound(fun, T_min, T_max, xtol=1e-03, maxfun=30)
    T=brentq(fun,T_min,T_max)
    #Water vapor on FG
    P_SatVapFG_OUT=PropsSI("P", "T", T, "Q", 0,"Water")
    Y_H2O_sat=P_SatVapFG_OUT / P_ref
    Y_H2O=HR_FG_OUT_max*Y_H2O_sat
    X_H2O=Y_H2O/(1-Y_H2O)
    MC_FG_Out_db=X_H2O*Mm_H2O/Mm_FG_db
    return T,MC_FG_Out_db,Y_H2O

def Iteration_TandM(T,h_tot,compo_db_FG):
    #Molar fraction db
    X_CO2=compo_db_FG[0]       #[mol CO2/mol dry fg]
    X_N2=compo_db_FG[2]        #[mol N2/mol dry fg]
    X_O2=compo_db_FG[1]        #[mol O2/mol dry fg] Fire hazards if higher than 10%
    Mm_FG_db=X_CO2*(Mm_CO2)+X_N2*(Mm_N2)+X_O2*(Mm_O2)
    #Water vapor on FG
    P_SatVapFG_OUT=PropsSI("P", "T", T, "Q", 0,"Water")
    Y_H2O_sat=P_SatVapFG_OUT / P_ref
    Y_H2O=HR_FG_OUT_max*Y_H2O_sat
    P_v=Y_H2O*P_SatVapFG_OUT
    H_V=PropsSI('H','P',max(P_v,611.655),'Q',1,'Water')/1000
    H_L=PropsSI('H','P',max(P_v,611.655),'Q',0,'Water')/1000
    Lv_vap_OUT=H_V - H_L                     #Water latent value [kJ/kg]
    Cp_VapourFG_OUT=PropsSI('CPMASS','P',P_v,'T',T,'Water')/1000
    h_vap_fg_OUT=(Lv_vap_OUT+Cp_VapourFG_OUT*(T-T_ref))
    X_H2O=Y_H2O/(1-Y_H2O) #Volumen fraction in db
    MC_FG_Out_db=X_H2O*Mm_H2O/Mm_FG_db
    #Cp just FG dried
    Cp_FG_OUT=Get_CpFG_db(compo_db_FG,T)/1000
    #Entalphy of just FG (without water) db
    h_FG_OUT=Cp_FG_OUT*(T-T_ref)                       #[kJ/kg FG db]
    #Total Entalphy of the whole FG
    h_tot_iter=h_FG_OUT+MC_FG_Out_db*h_vap_fg_OUT       #[kJ/kg FG db]
    MC_FG_Out_db_2=(h_tot-h_FG_OUT)/h_vap_fg_OUT
    T_2=(h_tot-MC_FG_Out_db*h_vap_fg_OUT)/Cp_FG_OUT+T_ref
#   _____
#    h_tot_iter=1*(T-T_ref)+MC_FG_Out_db*(2501+1.805*(T-T_ref))
#   _____
    res=h_tot-h_tot_iter
#    print("h_tot",h_tot,"h_tot_iter",h_tot_iter)
#    print("MC_FG_Out_db",MC_FG_Out_db,"MC_FG_Out_db_2",MC_FG_Out_db_2)
#    res=MC_FG_Out_db_2-MC_FG_Out_db
#    print("T",T,"T_2",T_2)
    res=T-T_2
    
    return res

def Get_Rho_Average(FG_IN,FG_OUT):
    Rho_FG_IN=FG_IN.rho_db
    Rho_FG_OUT=FG_OUT.rho_db
    Rho_FG_average=(Rho_FG_OUT+Rho_FG_IN)/2
    return Rho_FG_average

# cp
a_cp=np.array([2.2103, -3.4255000, 4.9436000, -3.94790 ,-5.3108, 1.48900 ,4.07830 ,-17.6680 ,-11.636 ,-18.620 ,-24.331
,-31.7590 ,0.95438])
b_cp=np.array([-3.8184, 4.8470000 ,-6.264100 ,5.755700 ,9.09030 ,-2.9223 ,-8.0091 ,27.86180 ,25.5480 ,34.0620 ,46.5630 ,
61.76330 ,-0.6875])
c_cp=np.array([6.08270, 1.1190000 ,5.3017000 ,0.931800 ,-2.3569 ,5.72450 ,8.21470 ,-12.0257 ,-16.097 ,-19.517 ,-29.394
,-38.9150 ,4.0549])
d_cp=np.array([-8.5309, 4.2939000 ,2.5038000 ,7.222700 ,22.0440 ,-8.1762 ,-12.697 ,39.17620 ,66.2570 ,75.6580 ,118.890 ,
158.4600 ,-0.3014])
e_cp=np.array([1.38464, -0.068363 ,-0.021273 ,-0.73425 ,-2.5107 ,1.45690 ,1.75360 ,-3.61900 ,-7.8851 ,-8.2042 ,-13.763
,-18.6000 ,0.3768])
```

```
f_cp=np.array([-9.6258, -2.023300 ,-0.790000 ,4.955000 ,14.4630 ,-10.877 ,-12.028 ,20.26850 ,51.2520 ,50.6110 ,88.1480 ,
119.9700 ,-2.2393])
g_cp=np.array([2.51970, 1.0390000 ,0.2849600 ,-1.33690 ,-3.3690 ,3.02800 ,3.36810 ,-4.97670 ,-13.703 ,-13.193 ,-23.429
,-32.0167 ,0.308685])
# Mm
M_Mm=np.array([28.01, 31.9988, 44.01, 18.02, 64.06, 28.011, 2.016, 16.04, 28.054, 30.7, 44.097, 58.12, 34.076])

def Get_CpFG_db(compo_db,T):
    X=np.array([compo_db[2],compo_db[1],compo_db[0],0,0,0,0,0,0,0,0,0,0])
    if T < 273.15:
        T=273.15
    if T > 2500:    
        T=2500
    cpT=(a_cp*10**4*T**(-2)+b_cp*10**(2)*T**(-1)+c_cp+d_cp*10**(-3)*T+e_cp*10**-5*T**2+f_cp*10**(-9)*T**(3)
+g_cp*10**(-12)*T**4)*X*R 
    out=sum(cpT)/MM_db(compo_db)*1000
    return out
def MM_db(compo_db):
    X=np.array([compo_db[2],compo_db[1],compo_db[0],0,0,0,0,0,0,0,0,0,0])
    Mc=M_Mm*X/1000.0
    out=sum(Mc)*1000
    return out

def Get_Mix(Mass_Air_db,Mass_FG_db,Air,FG,T):
    
    nTotal_db=np.sum(Mass_FG_db/FG.Mm_db)+np.sum(Mass_Air_db/Air.Mm_db)
    n=np.multiply(Air.compo_db,Mass_Air_db)/Air.Mm_db+np.multiply(FG.compo_db,Mass_FG_db)/Air.Mm_db
    compo_db=n/nTotal_db
    Air_x_H2O=Air.y_H2O/(1-Air.y_H2O)
    FG_x_H2O=FG.y_H2O/(1-FG.y_H2O)
    n_H2O=Air_x_H2O*Mass_Air_db/Air.Mm_db+FG_x_H2O*Mass_FG_db/FG.Mm_db
    xfin=n_H2O/nTotal_db
    yfin=xfin/(1+xfin)
#    print("yfin",yfin)
    FGmix=MyProperties_Gas_2(compo_db,yfin,T,P_ref)
    MassFG21=Mass_Air_db+Mass_FG_db
    return FGmix,MassFG21
    
    
    
    
    
    
    
    
    
```

B.6 Economic calculations

```
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 17 23:21:02 2020

@author: salva
"""

import numpy as np
import math
from Boiler2 import fuel_properties
from Constants import Nyears,infl,irate,eps,G,be,deltaP_dryer,effiFAN,VFG,RatioAdAconveyor
from Constants import MCini_ad,k_conveyor,exp_conveyor,k_AirDuct,exp_AirDuct,k_Fan,exp_Fan
from Constants import CEPCI2000,CEPCI2002,CEPCI2018,CEPCIRef,DollarToEuro
from Constants import k_Cover,exp_Cover,k_FixedCost,bFuel
from Constants import f_PM, f_BM, f_TM1, f_TM2, f_FC

#ALL RETURNS IN k€

def NetPresentValue(Hours,RhoFGAverage,Aconveyor,MassFG_db,MassFuel_daf,MassFuel_ini_daf):
    
    FuelCostIni=OPEX_initial(Hours,MassFuel_ini_daf)  
    
    NPV_Holmberg=np.zeros(Nyears+1)
    NPV_Brammer=np.zeros(Nyears+1)
    NPV_BeltBand=np.zeros(Nyears+1)
    NPV_SolidBand=np.zeros(Nyears+1)

    NPV_BeltBand_max=np.zeros(Nyears+1)
    NPV_SolidBand_max=np.zeros(Nyears+1)

    NPV_BeltBand_min=np.zeros(Nyears+1)
    NPV_SolidBand_min=np.zeros(Nyears+1)

    NPV_Holmberg_RT=np.zeros(50)
    NPV_Brammer_RT=np.zeros(50)
    NPV_BeltBand_RT=np.zeros(50)
    NPV_SolidBand_RT=np.zeros(50)
    NPV_RT=np.zeros(50)
    
#    NPV_Myllymaa=np.zeros(Nyears+1)
    
    [CAPEX_IndexArray,CAPEX_array]=CAPEX(Aconveyor,MassFG_db)
    [OPEX_array,ElecCost,FuelCost,FixedCost]=OPEX(Hours,MassFuel_daf,
     MassFG_db,RhoFGAverage,CAPEX_array)
    #CAPEX=130%
    [OPEX_max_array,ElecCost1,FuelCost1,FixedCost_min]=OPEX(Hours,MassFuel_daf,
     MassFG_db,RhoFGAverage,CAPEX_array*1.3)
    #CAPEX=70%
    [OPEX_min_array,ElecCost2,FuelCost2,FixedCost_max]=OPEX(Hours,MassFuel_daf,
     MassFG_db,RhoFGAverage,CAPEX_array*0.7)

    AnnualRevenue=(FuelCostIni-OPEX_array)
    AnnualRevenue_max=(FuelCostIni-OPEX_min_array)
    AnnualRevenue_min=(FuelCostIni-OPEX_max_array)

    NPV_Holmberg[0]=-CAPEX_array[0] #Year0
    NPV_Brammer[0]=-CAPEX_array[1]
    NPV_BeltBand[0]=-CAPEX_array[2]
    NPV_SolidBand[0]=-CAPEX_array[3]
#    NPV_Myllymaa[0]=-CAPEX_array[4]
    
    #NPV CAPEX=70%
    NPV_BeltBand_max[0]=-CAPEX_array[2]*0.7
    NPV_SolidBand_max[0]=-CAPEX_array[3]*0.7
    #NPV CAPEX=130%
    NPV_BeltBand_min[0]=-CAPEX_array[2]*1.3
    NPV_SolidBand_min[0]=-CAPEX_array[3]*1.3
    yearfraction=0.01
    d=np.append(np.arange(0,1,yearfraction),1)
#    d=np.arange(0,1,yearfraction)

    RT_Brammer=50
    RT_BeltBand=50
    RT_SolidBand=50
    RT_Holmberg=50
    RT=50
    for i in range(Nyears):
        i=i+1
        if i<=Nyears:
            NPV_Holmberg[i]=NPV_Holmberg[i-1] + AnnualRevenue[0]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
```

```
            NPV_Brammer[i]=NPV_Brammer[i-1] + AnnualRevenue[1]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
            NPV_BeltBand[i]=NPV_BeltBand[i-1] + AnnualRevenue[2]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
            NPV_SolidBand[i]=NPV_SolidBand[i-1] + AnnualRevenue[3]/(
                    ((1+infl)**(i)*(1+irate)**(i)))

            NPV_BeltBand_max[i]=NPV_BeltBand_max[i-1] + AnnualRevenue_max[2]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
            NPV_SolidBand_max[i]=NPV_SolidBand_max[i-1] + AnnualRevenue_max[3]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
        
            NPV_BeltBand_min[i]=NPV_BeltBand_min[i-1] + AnnualRevenue_min[2]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
            NPV_SolidBand_min[i]=NPV_SolidBand_min[i-1] + AnnualRevenue_min[3]/(
                    ((1+infl)**(i)*(1+irate)**(i)))

    a1=False
    a2=False
    a3=False
    a4=False
    a5=False
    
    NPVlim=0.5
    
    NPV_Holmberg_RT[0]=-CAPEX_array[0]
    NPV_Brammer_RT[0]=-CAPEX_array[1]
    NPV_BeltBand_RT[0]=-CAPEX_array[2]
    NPV_SolidBand_RT[0]=-CAPEX_array[3]
    AnnRev=(AnnualRevenue[2]+AnnualRevenue[3])/2
    NPV_RT[0]=-(CAPEX_array[2]+CAPEX_array[3])/2
    for i in range(51):
        i=i+1
        if i<50:
            NPV_Brammer_RT[i]=NPV_Brammer_RT[i-1] + AnnualRevenue[1]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
            NPV_BeltBand_RT[i]=NPV_BeltBand_RT[i-1] + AnnualRevenue[2]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
            NPV_SolidBand_RT[i]=NPV_SolidBand_RT[i-1] + AnnualRevenue[3]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
            NPV_Holmberg_RT[i]=NPV_Holmberg_RT[i-1] + AnnualRevenue[0]/(
                    ((1+infl)**(i)*(1+irate)**(i)))
            NPV_RT[i]=NPV_RT[i-1] + AnnRev/(
                    ((1+infl)**(i)*(1+irate)**(i)))
    
            if NPV_Brammer_RT[i-1]<=0 and NPV_Brammer_RT[i]>=0 and a1==False:
                if NPV_Brammer_RT[i]<NPVlim:
                    RT_Brammer=i
                else:
                    for n in range(len(d)):
                        NPV_Brammer1=NPV_Brammer_RT[i-1]+AnnualRevenue[1]*d[n]/(
                                ((1+infl)**(i+d[n])*(1+irate)**(i+d[n])))
                        if NPV_Brammer1>=0 :
                            RT_Brammer=i-1+d[n]
                            a1=True
                            break
                    
            if NPV_BeltBand_RT[i-1]<=0 and NPV_BeltBand_RT[i]>=0 and a2==False:
                if NPV_BeltBand_RT[i]<NPVlim:
                    RT_BeltBand=i
                else:
                    for n in range(len(d)):
                        NPV_BeltBand1=NPV_BeltBand_RT[i-1]+AnnualRevenue[2]*d[n]/(
                                ((1+infl)**(i+d[n])*(1+irate)**(i+d[n])))
                        if NPV_BeltBand1>=0 :
                            RT_BeltBand=i-1+d[n]
                            a2=True
                            break
                    
            if NPV_SolidBand_RT[i-1]<=0 and NPV_SolidBand_RT[i]>=0 and a3==False:
                if NPV_SolidBand_RT[i]<NPVlim:
                    RT_SolidBand=i
                else:
                    for n in range(len(d)):
                        NPV_SolidBand1=NPV_SolidBand_RT[i-1]+AnnualRevenue[3]*d[n]/(
                                ((1+infl)**(i+d[n])*(1+irate)**(i+d[n])))
                        if NPV_SolidBand1>=0 :
                            RT_SolidBand=i-1+d[n]
                            a3=True
                            break
                    
```

```
            if NPV_Holmberg_RT[i-1]<=0 and NPV_Holmberg_RT[i]>=0 and a4==False:
                if NPV_Holmberg_RT[i]<NPVlim:
                    RT_Holmberg=i
                else:
                    for n in range(len(d)):
                        NPV_Holmberg1=NPV_Holmberg_RT[i-1]+AnnualRevenue[0]*d[n]/(
                                ((1+infl)**(i+d[n])*(1+irate)**(i+d[n])))
                        if NPV_Holmberg1>=0 :
                            RT_Holmberg=i-1+d[n]
                            a4=True
                            break
                    
            if NPV_RT[i-1]<=0 and NPV_RT[i]>=0 and a5==False:
                if NPV_RT[i]<NPVlim:
                    RT=i
                else:
                    for n in range(len(d)):
                        NPV_RT1=NPV_RT[i-1]+AnnRev*d[n]/(
                                ((1+infl)**(i+d[n])*(1+irate)**(i+d[n])))
                        if NPV_RT1>=0 :
                            RT=i-1+d[n]
                            a5=True
                            break
        
    return (NPV_Holmberg,NPV_Brammer,NPV_SolidBand,NPV_BeltBand,
            NPV_SolidBand_max,NPV_BeltBand_max,NPV_SolidBand_min,NPV_BeltBand_min,
            RT_Holmberg,RT_Brammer,RT_SolidBand,RT_BeltBand,RT,
            AnnualRevenue,FuelCostIni,CAPEX_IndexArray,CAPEX_array,
            OPEX_array,ElecCost,FuelCost, FixedCost)

def CAPEX(Aconveyor,MassFG_db):
    
    CAPEX_IndexArray=["Holmberg","Brammer","BeltBand","SolidBand","Myllymaa"]
    CAPEX_array=np.zeros(len(CAPEX_IndexArray))

    if Aconveyor !=0:
        CAPEX_array[0]=Get_CAPEXHolmberg(Aconveyor,np.max(MassFG_db)) #k€
        CAPEX_array[1]=Get_CAPEXBrammer(Aconveyor) #k€
        CAPEX_array[2]=Get_CAPEXBeltBand(Aconveyor) #k€
        CAPEX_array[3]=Get_CAPEXSolidBand(Aconveyor) #k€
        CAPEX_array[4]=Get_CAPEXMyllymaa(Aconveyor) #k€
    
    return CAPEX_IndexArray,CAPEX_array

def OPEX(Hours,MassFuel_daf,MassFG_db,RhoFGAverage, CAPEX_array):
     
    OPEX_array=np.zeros(len(CAPEX_array))
    ElecCost= (be/10**6/3600)*(np.sum(MassFG_db*Hours*3600)*deltaP_dryer/(effiFAN*RhoFGAverage))/1000 #k€
    #be is transformed to €/Wh for unit unification in SI UNITS
    [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,
     HHV_Fuel_daf]=fuel_properties(MCini_ad,"fuel_properties")
    
    FuelCost = (bFuel/10**6/3600)*(np.sum(MassFuel_daf*Hours)*3600/(1-MCini_ad/100-Ash_Fuel_ad/100)*LHV_Fuel_ad)/1000 
    #k€ # (€/MWh/10**6 * Kg/s * J/kg * h /1000) 
    FixedCost=k_FixedCost*CAPEX_array #k€
    OPEX_array=ElecCost+FuelCost+FixedCost
    
    return OPEX_array, ElecCost, FuelCost, FixedCost

def OPEX_initial(Hours,MassFuel_ini_daf):
    [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,LHV_Fuel_daf,
     HHV_Fuel_daf]=fuel_properties(MCini_ad,"fuel_properties")
    
    FuelCost=(bFuel/10**6/3600)*(np.sum(MassFuel_ini_daf*Hours*3600)/(1-MCini_ad/100-Ash_Fuel_ad/100)*LHV_Fuel_ad)/1000 
    #k€ # (€/MWh/10**6Wh/3600s * kg/year * J/kg * h /1000) 
    return FuelCost

def Get_CAPEXHolmberg(Aconveyor,MassFG_db):
    Acover=RatioAdAconveyor*Aconveyor
    Cost_conveyor=k_conveyor*Aconveyor**(exp_conveyor) * CEPCI2018/CEPCI2000
    Cost_AirDuct=k_AirDuct*MassFG_db**(exp_AirDuct) * CEPCI2018/CEPCI2002
    Cost_Fan=k_Fan*MassFG_db**(exp_Fan) * CEPCI2018/CEPCI2002
    Cost_Cover=k_Cover*Acover**exp_Cover * CEPCI2018/CEPCI2002
    CAPEX= G  * (Cost_conveyor + Cost_AirDuct + Cost_Fan + Cost_Cover) / 1000 #k€
    return CAPEX

def Get_CAPEXBrammer(Aconveyor):
    CAPEX= 7820*(2.79*Aconveyor+52.2)**0.863 / 1000 * CEPCI2018/CEPCI2000 #k€
    return CAPEX
```

```
def Get_CAPEXBeltBand(Aconveyor):
#from Rules of Thumb:
#Belt/band with flow through circulation (convection) c/s including fan, motor,
#feeders. FOB cost = $350 000 at drying area = 10 m2 with n = 1.0 for the range
#1–10; FOB cost = $450 000 at drying area = 20 m2 for the range 20–40. L+M*
#= 2.4. L/M = 0.38. Alloy factor: c/s, x 1.00; 304 s/s, x 1.45; 316 s/s, q 2.3.
    if Aconveyor<=10: #1-10m2
        FOB_CEPCIRef=350000 #$
        Aref=10
        n=1
    elif Aconveyor>=20: #20-40m2
        FOB_CEPCIRef=450000
        Aref=20
        n=1
    elif 10<Aconveyor<20: #In between
        FOB_CEPCIRef=350000
        Aref=10
        n=math.log(350000/450000)/math.log(10/20)
        
        
    FOB=FOB_CEPCIRef*CEPCI2018/CEPCIRef *DollarToEuro #€
    LandM=2.4
    CostFOB=FOB*(Aconveyor/Aref)**(n) #€
    CostLandM=CostFOB*LandM
    CostPM=CostLandM+CostFOB*f_PM
    CostBM=CostPM+f_BM*CostLandM
    CostTM=CostBM*(1+f_TM1+f_TM2)
    CAPEX=CostTM*(1+f_FC) /1000 #k€
#    CAPEX=91648*Aconveyor / 1000 #k€
    return CAPEX

def Get_CAPEXSolidBand(Aconveyor):
#from Rules of Thumb:
#Solid band (convection), c/s including fans, motors, direct firing, conveyor feeder
#for bands > 6 m long. Band width 1.2, 1.5, 2 and 2.8 m. For intermediate temperatures,
#260 hC. FOB cost = $220 000 at band area = 10 m2 with n = 1.04 for
#the range 1.5–25. L+M* = 2.4. L/M = 0.38. Factors temperature 260 hC, x 1.00;
#> 540 ºC, x 1.9; low temperature, x 0.6. Alloy factor: c/s, q 1.00; 304 s/s,
#x 1.45; 316 s/s, x 2.3.
    #1.5-25 m2
    FOB_CEPCIRef=220000 #$
    FOB=FOB_CEPCIRef*CEPCI2018/CEPCIRef *DollarToEuro #€
    Aref=10
    n=1.04
    LandM=2.4
    CostFOB=FOB*(Aconveyor/Aref)**(n) #€
    CostLandM=CostFOB*LandM
    CostPM=CostLandM+CostFOB*f_PM
    CostBM=CostPM+f_BM*CostLandM
    CostTM=CostBM*(1+f_TM1+f_TM2)
    CAPEX=CostTM*(1+f_FC) /1000 #k€
#    CAPEX=(108.41*Aconveyor**2+57456*Aconveyor-8689.2) / 1000 #k€
    return CAPEX

def Get_CAPEXMyllymaa(Aconveyor):
    CAPEX=0
    if Aconveyor !=0:
        CAPEX=(-3095*math.log(Aconveyor/480)+5838)*Aconveyor /1000 #k€
    return CAPEX
```

B.7 Gas properties

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Feb  6 17:57:15 2019

@author: thibaultcoppieters
"""

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""

Created on Fri Jan 18 11:50:04 2019

@author: thibaultcoppieters
"""
from pyXSteam.XSteam import XSteam
#from cantera import Solution as sol  # see if ThermoPhase is not better
from Constants import Mm_H2O
from CoolProp.CoolProp import PropsSI
import numpy as np 


 
class MyProperties_Gas_2():

    TK=273.15
    R=8.314472 # Universal gas constant [J.mol^-1.K^-1]
    
    # Lambda
    a_lambda=np.array([0.3918, -0.32720, -7.21390, 17.5000, -8.0847, 0.50660, 168.0, -1.86860, -17.6013, -31.6103, 1.858000,
1.858000, 17.5])
    b_lambda=np.array([0.9814, 0.996500, 0.801400, 0.65864, 0.63430, 0.91230, 5.680, 0.872500, 1.199500, 2.201400, 0.047000,
0.047000, 0.65864])
    c_lambda=np.array([-5.066, -3.74260 ,0.547600, -3.4412, -1.3816, -3.5236, 2.354, 11.78570, 3.333900, -19.2300, 21.76300,
21.76300, -3.4412])
    d_lambda=np.array([1.5034, 0.973012, -1.05256, 100.910, 0.23027, 0.83560, 0.000, -3.61362, -1.36573, 16.63834, -8.40709,
-8.40709, 100.91])
        
    # mu 
    mu1=np.array([17.499, 20.194, 14.568, 14.2,  12.626, 16.6,  8.78,   10.745, 10.152, 9.181, 8.181, 7.359, 12.674])
    T1_mu =np.array([20.0,   20.0,   20.0,   20.0,  20.0,   0.0,   20.0,   20.0,   20.0,   20.0,  28.0,  20.0,  33.0])
    Tc_mu =np.array([-196.0, -183.0, -78.5,  99.63, -10.0, -191.5, -253.0, -258.7, -103.7, -88.0, -42.1, -0.5,  -60.0])
    
    # cp
    a_cp=np.array([2.2103, -3.4255000, 4.9436000, -3.94790 ,-5.3108, 1.48900 ,4.07830 ,-17.6680 ,-11.636 ,-18.620 ,-24.331
,-31.7590 ,0.95438])
    b_cp=np.array([-3.8184, 4.8470000 ,-6.264100 ,5.755700 ,9.09030 ,-2.9223 ,-8.0091 ,27.86180 ,25.5480 ,34.0620 ,46.5630 ,
61.76330 ,-0.6875])
    c_cp=np.array([6.08270, 1.1190000 ,5.3017000 ,0.931800 ,-2.3569 ,5.72450 ,8.21470 ,-12.0257 ,-16.097 ,-19.517 ,-29.394
,-38.9150 ,4.0549])
    d_cp=np.array([-8.5309, 4.2939000 ,2.5038000 ,7.222700 ,22.0440 ,-8.1762 ,-12.697 ,39.17620 ,66.2570 ,75.6580 ,118.890 ,
158.4600 ,-0.3014])
    e_cp=np.array([1.38464, -0.068363 ,-0.021273 ,-0.73425 ,-2.5107 ,1.45690 ,1.75360 ,-3.61900 ,-7.8851 ,-8.2042 ,-13.763
,-18.6000 ,0.3768])
    f_cp=np.array([-9.6258, -2.023300 ,-0.790000 ,4.955000 ,14.4630 ,-10.877 ,-12.028 ,20.26850 ,51.2520 ,50.6110 ,88.1480 ,
119.9700 ,-2.2393])
    g_cp=np.array([2.51970, 1.0390000 ,0.2849600 ,-1.33690 ,-3.3690 ,3.02800 ,3.36810 ,-4.97670 ,-13.703 ,-13.193 ,-23.429
,-32.0167 ,0.308685])

    # Mm
    M_Mm=np.array([28.01, 31.9988, 44.01, 18.02, 64.06, 28.011, 2.016, 16.04, 28.054, 30.7, 44.097, 58.12, 34.076])
   
    # Rho
    ro0=np.array([1.165, 1.331, 1.842, 0.804, 2.279, 1.165, 0.0866, 0.668, 1.260 ,1.264, 1.882, 2.489, 1.434])
    
    
    
    def __init__(self,compo_db_fg,y_H2O,T_fg,P_fg): #,mdot_nc):
        
        self.T=T_fg
        self.P=P_fg       
        self.compo_db=compo_db_fg
        self.y_H2O=y_H2O
        self.compo_wb=compo_db_fg*(1-self.y_H2O)
        self.compo_wb=np.append(self.compo_wb,self.y_H2O)    
        self.X=np.array([self.compo_wb[2],self.compo_wb[1],self.compo_wb[0],self.compo_wb[3],0,0,0,0,0,0,0,0,0])
        self.Mm=self.MM(self.compo_wb)
        self.Mm_db=self.MM_db(self.compo_db)
        self.rho=self.Rho(self.compo_wb,self.T)
        self.rho_db=self.Rho_db(self.compo_db,self.T)
```

```python
        self.k=self.Lambda(self.compo_wb,self.T) # lambda as defined in the matlab code
        self.cp=self.Cp(self.compo_wb,self.T)
        self.cp_db=self.Cp_db(self.compo_db,self.T)
        self.mu=self.Mu(self.compo_wb,self.T) #as defined in the matlab code
        self.alpha=self.Alpha(self.compo_wb,self.T,self.k,self.rho,self.cp) 
        self.D_H2O=(7.65*10**-5*self.T**(11/6))/10**5
        self.Pr=self.mu*self.cp/self.k
        self.xsteam=XSteam(XSteam.UNIT_SYSTEM_MKS)     
        if self.y_H2O==0:
            a=1
            self.X_H2O=0
            self.X_H2O_dry=0
        else:       
            self.T_dew=self.xsteam.tsat_p(self.y_H2O*self.P*10**-5)+273.15        
            self.X_H2O=(Mm_H2O*self.y_H2O)/self.Mm
#            self.X_H2O_dry=(Mm_H2O)*(self.y_H2O/(1-self.y_H2O))/self.Mm_dry
#        (Mm_H2O*self.y_H2O)/self.Mm_dry
        # Enthalpie gas: 
        self.P_v=self.y_H2O*PropsSI("P","T",self.T,"Q",1,"Water")
        self.H_V=PropsSI("H","P",max(self.P_v,611.656),"Q",1,"Water") #J/kg
        self.H_L=PropsSI("H","P",max(self.P_v,611.656),"Q",0,"Water") #J/kg
        #611.655 is the min value acceptable by CoolProps
        self.Lv=self.H_V-self.H_L #J/kg
#        self.h=self.cp*(self.T-273.15)+self.X_H2O*(1.86*(self.T-273.15)+2501)*10**3 #J/kg
        self.h=self.cp*(self.T-273.15)+self.X_H2O*(1.86*(self.T-273.15)+self.Lv/1000)*10**3 #J/kg
        self.Cp_v=PropsSI("CPMASS","P",max(self.P_v,611.657),"T",self.T,"Water")
        self.h_db=self.cp_db*(self.T-273.15)+(self.X_H2O/(1-self.X_H2O))*(self.Cp_v/1000*(self.T-273.15)+self.Lv/1000)*10**3
#        self.string='HEOS::Nitrogen'+'['+str(self.compo_wb[2])+']'+'&Oxygen'+'['+(
#                str(self.compo_wb[1])+']'+'&CarbonDioxide'+'['+str(self.compo_wb[0])+']'+'&Water'+'['+str(self.compo_wb[3])
+']')
#        self.cp=PropsSI('Cp0mass','T',self.T,'P',self.P,self.string)
        
            
    def update_T_P(self,T_new,P_new):
        self.T=T_new
        self.P=P_new          
        self.Mm=self.MM(self.compo_wb)
        self.rho=self.Rho(self.compo_wb,self.T)
        self.k=self.Lambda(self.compo_wb,self.T) # lambda as defined in the matlab code
        self.cp=self.Cp(self.compo_wb,self.T)
        self.mu=self.Mu(self.compo_wb,self.T) #as defined in the matlab code
        self.alpha=self.Alpha(self.compo_wb,self.T,self.k,self.rho,self.cp) 
        self.D_H2O=(7.65*10**-5*self.T**(11/6))/10**5  
        self.Pr=self.mu*self.cp/self.k
        
        if self.y_H2O==0:
            a=1
            self.X_H2O=0
        else:       
            self.T_dew=self.xsteam.tsat_p(self.y_H2O*self.P*10**-5)+273.15  
            
        # Enthalpie gas: 
#        self.h=self.cp*(self.T-273.15)+self.X_H2O*(1.86*(self.T-273.15)+2501)*10**3
        self.P_v=self.y_H2O*PropsSI("P","T",self.T,"Q",1,"Water")
        self.H_V=PropsSI("H","P",self.P_v,"Q",1,"Water") #J/kg
        self.H_L=PropsSI("H","P",self.P_v,"Q",0,"Water") #J/kg
        self.Lv=self.H_V-self.H_L #J/kg
#        self.h=self.cp*(self.T-273.15)+self.X_H2O*(1.86*(self.T-273.15)+2501)*10**3 #J/kg
        self.h=self.cp*(self.T-273.15)+self.X_H2O*(1.86*(self.T-273.15)+self.Lv/1000)*10**3 #J/kg
        self.Cp_v=PropsSI("CPMASS","P",self.P_v,"T",self.T,"Water")
        self.h_db=self.cp_db*(self.T-273.15)+(self.X_H2O/(1-self.X_H2O))*(self.Cp_v/1000*(self.T-273.15)+self.Lv/1000)*10**3
      
    def update_T_P_mod(self,T_new,P_new):
        
        if self.y_H2O==0:
            a=1
            self.T_dew=0
            self.X_H2O=0
        else:       
            self.T_dew=self.xsteam.tsat_p(self.y_H2O*self.P*10**-5)+273.15 
        
        if T_new<=self.T_dew:
            self.y_H2O=self.xsteam.psat_t(T_new-273.15)/(self.P*10**-5)
            self.compo_wb=self.compo_db*(1-self.y_H2O)
            self.compo_wb=np.append(self.compo_wb,self.y_H2O) 
            self.T_dew=self.xsteam.tsat_p(self.y_H2O*self.P*10**-5)+273.15
            
        self.T=T_new
        self.P=P_new          
        self.Mm=self.MM(self.compo_wb)
        self.rho=self.Rho(self.compo_wb,self.T)
```

```
        self.k=self.Lambda(self.compo_wb,self.T) # lambda as defined in the matlab code
        self.cp=self.Cp(self.compo_wb,self.T)
        self.mu=self.Mu(self.compo_wb,self.T) #as defined in the matlab code
        self.alpha=self.Alpha(self.compo_wb,self.T,self.k,self.rho,self.cp) 
        self.D_H2O=(7.65*10**-5*self.T**(11/6))/10**5  
        self.Pr=self.mu*self.cp/self.k
        self.X_H2O_old=self.X_H2O
        self.X_H2O=(Mm_H2O*self.y_H2O)/self.Mm
        
        # Enthalpie gas: 
        self.h=self.cp*(self.T-273.15)+self.X_H2O*(1.86*(self.T-273.15)+2501)*10**3
        self.h_cw=self.xsteam.Cp_pt(1,self.T-273.15)*10**3*(self.X_H2O_old-self.X_H2O)*(self.T-273.15)

    def update_y_H2O_T_P(self,y_H2O_new,T_new,P_new):

        self.T=T_new
        self.P=P_new       
        self.y_H2O=y_H2O_new
        self.compo_wb=self.compo_db*(1-self.y_H2O)
        self.compo_wb=np.append(self.compo_wb,self.y_H2O)    
        self.X=np.array([self.compo_wb[2],self.compo_wb[1],self.compo_wb[0],self.compo_wb[3],0,0,0,0,0,0,0,0])
        self.Mm=self.MM(self.compo_wb)
        self.rho=self.Rho(self.compo_wb,self.T)
        self.k=self.Lambda(self.compo_wb,self.T) # lambda as defined in the matlab code
        self.cp=self.Cp(self.compo_wb,self.T)
        self.mu=self.Mu(self.compo_wb,self.T) #as defined in the matlab code
        self.alpha=self.Alpha(self.compo_wb,self.T,self.k,self.rho,self.cp) 
        self.D_H2O=(7.65*10**-5*self.T**(11/6))/10**5  
        self.Pr=self.mu*self.cp/self.k
        if self.y_H2O==0:
            a=1
            self.X_H2O=0
        else: 
            self.T_dew=self.xsteam.tsat_p(self.y_H2O*self.P*10**-5)+273.15       
            self.X_H2O=(Mm_H2O*self.y_H2O)/self.Mm
        
        # Enthalpie gas: 
        self.h=self.cp*(self.T-273.15)+self.X_H2O*(1.86*(self.T-273.15)+2501)*10**3
        
    def get(self,nbr,*arg):
        
        RETURN=np.zeros(nbr)
        
        i=0
        for n in arg:
            if n=="y_H2O":
                RETURN[i]=self.y_H2O
            elif n=="compo_db":
                RETURN[i]=self.compo_db
            elif n=="compo_wb":
                RETURN[i]=self.compo_wb
            elif n=="T":
                RETURN[i]=self.T
            elif n=="P":
                RETURN[i]=self.P
            elif n=="Mm":
                RETURN[i]=self.Mm
            elif n=="rho":
                RETURN[i]=self.rho
            elif n=="cp":
                RETURN[i]=self.cp
            elif n=="k":
                RETURN[i]=self.k
            elif n=="mu":
                RETURN[i]=self.mu
            elif n=="alpha":
                RETURN[i]=self.alpha
            elif n=="D_H2O":
                RETURN[i]=self.D_H2O
            elif n=="T_dew":
                RETURN[i]=self.T_dew  
            elif n=="X_H2O":
                RETURN[i]=self.X_H2O
            elif n=="mdot_tot":
                RETURN[i]=self.mdot_tot
            elif n=="mdot_H2O":
                RETURN[i]=self.mdot_H2O
            elif n=="rho_cp_k_mu":
                RETURN[i:i+4]=[self.rho,self.cp,self.k,self.mu]
                i+=3
#           elif n=="mdot_cell":
```

```
#               RETURN[i]=self.mdot_cell
#           elif n=="mdot_H2O_cell":
#               RETURN[i]=self.mdot_H2O_cell
            elif n=="hfg":
                RETURN[i]=self.hfg
            else:
                raise Exception
            i+=1
            
        return RETURN

    def print_element(self):
        
        print("Composition db: "+str(self.compo_db))
        print("Water mole fraction: "+str(self.y_H2O))
        print("Temperature T: "+str(self.T))
        print("Pressure P: "+str(self.P))
        print("Molecular weight Mm: "+str(self.Mm))
        print("Density_mass rho: "+str(self.rho))
        print("thermal_conductivity k: "+str(self.k))
        print("cp_mass cp :"+str(self.cp))
        print("viscosity :"+str(self.mu))
        print("Dew point T_dew: "+str(self.T_dew))
#        print("mdot_nc: "+str(self.mdot_nc))
#        print("mdot_tot: "+str(self.mdot_tot))
#        print("mdot_H2O: "+str(self.mdot_H2O))
        print("X_H2O "+str(self.X_H2O))
        


    def Lambda(self,compo_wb,T):
    
        #
        ## Description
        #
        # Function for thermal conductivity calculation  
        # of gas with composition X and temperature T.
        #
        # lambda = (a.10^(-3) + b.10^(-4).T + c.10^(-8).T^2 + d.10^(-11).T^3).X
        #
        ## Inputs 
        #
        # X - vector of volume fractions for next componemt of gas mixture:
        #   X(1)  - volume fraction of N2                              [m^3.m^-3]
        #   X(2)  - volume fraction of O2                              [m^3.m^-3]
        #   X(3)  - volume fraction of CO2                             [m^3.m^-3]
        #   X(4)  - volume fraction of H2O                             [m^3.m^-3]
        #   X(5)  - volume fraction of SO2                             [m^3.m^-3]
        #   X(6)  - volume fraction of CO                              [m^3.m^-3]
        #   X(7)  - volume fraction of H2                              [m^3.m^-3]
        #   X(8)  - volume fraction of CH4                             [m^3.m^-3]
        #   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
        #   X(10) - volume fraction of C2H6                            [m^3.m^-3]
        #   X(11) - volume fraction of C3H8                            [m^3.m^-3]
        #   X(12) - volume fraction of C4H10                           [m^3.m^-3]
        #   X(13) - volume fraction of H2S                             [m^3.m^-3]
        #
        # T - Temperature of mixture                                   [K]
        #
        ## Used coeficients and functions
        #
        # a, b, c, d - parameters for thermal conductivity calculation
        #
        ## Output
        #
        # lambda - thermal conductivity of gas mixture                [W.m^-1.K^-1]
        #          defined for temperature T =<273,15;2500>.
        # 
        ## Copyright (C) 2011
        # Authors      : Kukurugya Jan, Terpak Jan
        # Organization: Technical University of Kosice
        # e-mail       : jan.kukurugya@tuke.sk, jan.terpak@tuke.sk
        # Revision     : 28.12.2011
        #
        #   1      2        3        4       5       6       7     8        9        10       11       12       13 
        #   N2     O2       CO2      H2O     SO2     CO      H2    CH4      C2H4     C2H6     C3H8     C4H10    H2S   
        #a=[0.3918 -0.32720 -7.21390 17.5000 -8.0847 0.50660 168.0 -1.86860 -17.6013 -31.6103 1.858000 1.858000 17.5];
        #b=[0.9814 0.996500 0.801400 0.65864 0.63430 0.91230 5.680 0.872500 1.199500 2.201400 0.047000 0.047000 0.65864];
        #c=[-5.066 -3.74260 0.547600 -3.4412 -1.3816 -3.5236 2.354 11.78570 3.333900 -19.2300 21.76300 21.76300 -3.4412];
        #d=[1.5034 0.973012 -1.05256 100.910 0.23027 0.83560 0.000 -3.61362 -1.36573 16.63834 -8.40709 -8.40709 100.91];
```

```
    
        X=np.array([compo_wb[2],compo_wb[1],compo_wb[0],compo_wb[3],0,0,0,0,0,0,0,0,0])
        
        
          
        if T < 273.15:
            T=273.15
        if T > 2500:    
            T=2500
         
        #la=np.zeros(len(X))
#        for i in range (0,13):
#            la[i]=(a[i]*10**(-3) + b[i]*10**(-4)*T + c[i]*10**(-8)*T**2 + d[i]*10**(-11)*T**3)*X[i]
#        
        la=(self.a_lambda*10**(-3) + self.b_lambda*10**(-4)*T + self.c_lambda*10**(-8)*T**2 + self.d_lambda*10**(-11)*T**3)*X
        
        out=sum(la)
    
        return out


    def Mu(self,compo_wb,T):
    #
    ## Description
    #
    # Function for dynamic viscozity calculation 
    # of gas with composition X and temperature T.
    #
    #       mu1 . (T1 + 1,47.Tc)        T^1,5
    # mu = ---------------------- . ---------- . 10^(-6) . X
    #          T + 1,47.Tc           T1^1,5
    #
    ## Inputs 
    #
    # X - vector of volume fractions for next componemt of gas mixture:
    #   X(1)  - volume fraction of N2                              [m^3.m^-3]
    #   X(2)  - volume fraction of O2                              [m^3.m^-3]
    #   X(3)  - volume fraction of CO2                             [m^3.m^-3]
    #   X(4)  - volume fraction of H2O                             [m^3.m^-3]
    #   X(5)  - volume fraction of SO2                             [m^3.m^-3]
    #   X(6)  - volume fraction of CO                              [m^3.m^-3]
    #   X(7)  - volume fraction of H2                              [m^3.m^-3]
    #   X(8)  - volume fraction of CH4                             [m^3.m^-3]
    #   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
    #   X(10) - volume fraction of C2H6                            [m^3.m^-3]
    #   X(11) - volume fraction of C3H8                            [m^3.m^-3]
    #   X(12) - volume fraction of C4H10                           [m^3.m^-3]
    #   X(13) - volume fraction of H2S                             [m^3.m^-3]
    #
    # T - Temperature of mixture                                   [K]
    #
    ## Used coeficients and functions
    #
    # self.TK - constant for conversion Celsius degrees to Kelvins      [K]
    # mu1(1).. mu1(13) - dynamic viscosity of gas component 
    #                    for temperature T1(1)...T(13)             [Pa.s]
    # Tc(1) .. Tc(13)  - boiling temperature of gas component      [K]
    #
    ## Output
    #
    # mu - Dynamic viscosity of gas mixture                        [Pa.s]
    #      defined for temperature T =<273,15;2500>.
    #
    ## Copyright (C) 2011
    # Authors     : Kukurugya Jan, Terpak Jan
    # Organization: Technical University of Kosice
    # e-mail      : jan.kukurugya@tuke.sk, jan.terpak@tuke.sk
    # Revision    : 28.12.2011
    #
    #     1      2      3      4     5      6     7      8      9      10    11    12    13
    #     N2     O2     CO2    H2O   SO2    CO    H2     CH4    C2H4   C2H6  C3H8  C4H10 H2S  
    #mu1=[17.499 20.194 14.568 14.2  12.626 16.6  8.78   10.745 10.152 9.181 8.181 7.359 12.674];
    #T1 =[20.0   20.0   20.0   20.0  20.0   0.0   20.0   20.0   20.0   20.0  28.0  20.0  33.0];
    #Tc =[-196.0 -183.0 -78.5  99.63 -10.0 -191.5 -253.0 -258.7 -103.7 -88.0 -42.1 -0.5  -60.0];
        X=np.array([compo_wb[2],compo_wb[1],compo_wb[0],compo_wb[3],0,0,0,0,0,0,0,0,0])

        T1_mu=self.T1_mu+273.15    
        Tc_mu=self.Tc_mu+273.15    
            
        if T < 273.15:
            T=273.15
```

```
        if T > 2500:    
            T=2500;
    
#       muT=np.zeros(len(X))    
#       for i in range (0,13):
#           muT[i]=(mu1[i]*10**(-6)*X[i]*(T1[i] + 1.47*Tc[i])*T**1.5)/((T + 1.47*Tc[i])*T1[i]**1.5)
#        
        muT=(self.mu1*10**(-6)*X*(T1_mu+1.47*Tc_mu)*T**1.5)/((T + 1.47*Tc_mu)*T1_mu**1.5)
            
        out=sum(muT)
         
        return out

    def Alpha(self,compo_wb,T,k=None,rho=None,cp=None):
        
        if k==None:
            k=self.Lambda(compo_wb,T)
        if rho==None:
            rho=self.rho(compo_wb,T)
        if cp==None:
            cp=self.rho(compo_wb,T)
    
        out=k/(rho*cp)
        
        return out




    def Cp(self,compo_wb,T):
#
## Description
#
# Function for specific heat capacity calculation 
# of gas with composition X and temperature T.
#
# cp=( a.10^(4)*T^(-2) + b.10^(2).T^(-1) + c +d.10^(-3).T 
#    + e.10^(-5).T^(2) + f.10^(-9).T^(3) + g.10^(-12).T^(4)
#    ).X.R
#
## Inputs 
#
# X - vector of volume fractions for next componemt of gas mixture:
#   X(1)  - volume fraction of N2                              [m^3.m^-3]
#   X(2)  - volume fraction of O2                              [m^3.m^-3]
#   X(3)  - volume fraction of CO2                             [m^3.m^-3]
#   X(4)  - volume fraction of H2O                             [m^3.m^-3]
#   X(5)  - volume fraction of SO2                             [m^3.m^-3]
#   X(6)  - volume fraction of CO                              [m^3.m^-3]
#   X(7)  - volume fraction of H2                              [m^3.m^-3]
#   X(8)  - volume fraction of CH4                             [m^3.m^-3]
#   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
#   X(10) - volume fraction of C2H6                            [m^3.m^-3]
#   X(11) - volume fraction of C3H8                            [m^3.m^-3]
#   X(12) - volume fraction of C4H10                           [m^3.m^-3]
#   X(13) - volume fraction of H2S                             [m^3.m^-3]
#
# T - Temperature of mixture                                   [K]
#
## Used coeficients and functions
#
# a, b, c, d, e, f, g  - parameters for heat capacity calculation
# R - univerzal gas constant                                [J.mol^-1.K^-1]
#
## Output
#
# cp - Specific heat capacity of gas mixture                [J.mol^-1.K^-1]
#      defined for temperature T =<273,15;2500>.
#
## Copyright (C) 2011
# Authors     : Kukurugya Jan, Terpak Jan
# Organization: Technical University of Kosice
# e-mail      : jan.kukurugya@tuke.sk, jan.terpak@tuke.sk
# Revision    : 28.12.2011
#
#   1      2          3         4        5       6       7       8        9       10      11      12       13 
#   N2     O2         CO2       H2O      SO2     CO      H2      CH4      C2H4    C2H6    C3H8    C4H10    H2S   
#a=[2.2103 -3.4255000 4.9436000 -3.94790 -5.3108 1.48900 4.07830 -17.6680 -11.636 -18.620 -24.331 -31.7590 0.95438];
#b=[-3.8184 4.8470000 -6.264100 5.755700 9.09030 -2.9223 -8.0091 27.86180 25.5480 34.0620 46.5630 61.76330 -0.6875];
#c=[6.08270 1.1190000 5.3017000 0.931800 -2.3569 5.72450 8.21470 -12.0257 -16.097 -19.517 -29.394 -38.9150 4.0549];
#d=[-8.5309 4.2939000 2.5038000 7.222700 22.0440 -8.1762 -12.697 39.17620 66.2570 75.6580 118.890 158.4600 -0.3014];
```

```
#e=[1.38464 -0.068363 -0.021273 -0.73425 -2.5107 1.45690 1.75360 -3.61900 -7.8851 -8.2042 -13.763 -18.6000 0.3768];
#f=[-9.6258 -2.023300 -0.790000 4.955000 14.4630 -10.877 -12.028 20.26850 51.2520 50.6110 88.1480 119.9700 -2.2393];
#g=[2.51970 1.0390000 0.2849600 -1.33690 -3.3690 3.02800 3.36810 -4.97670 -13.703 -13.193 -23.429 -32.0167 0.308685];
        X=np.array([compo_wb[2],compo_wb[1],compo_wb[0],compo_wb[3],0,0,0,0,0,0,0,0,0])

        if T < self.TK:
            T=self.TK

        if T > 2500:    
            T=2500


#       cpT=np.zeros(len(X))  
#       
#       for i in range (0,13):
#           cpT[i]=a[i]*10**4*T**(-2)+b[i]*10**(2)*T**(-1) + c[i] + d[i]*10**(-3)*T
#           cpT[i]=cpT[i] + e[i]*10**-5*T**2 + f[i]*10**(-9)*T**(3) + g[i]*10**(-12)*T**4
#           cpT[i]=cpT[i]*X[i]*self.R

        cpT=(self.a_cp*10**4*T**(-2)+self.b_cp*10**(2)*T**(-1)+self.c_cp+self.d_cp*10**(-3)*T
+self.e_cp*10**-5*T**2+self.f_cp*10**(-9)*T**(3)+self.g_cp*10**(-12)*T**4)*X*self.R 

        out=sum(cpT)/self.MM(compo_wb)*1000

        return out

    def Cp_db(self,compo_db,T):
#
## Description
#
# Function for specific heat capacity calculation 
# of gas with composition X and temperature T.
#
# cp=( a.10^(4)*T^(-2) + b.10^(2).T^(-1) + c +d.10^(-3).T 
#    + e.10^(-5).T^(2) + f.10^(-9).T^(3) + g.10^(-12).T^(4)
#    ).X.R
#
## Inputs 
#
# X - vector of volume fractions for next componemt of gas mixture:
#   X(1)  - volume fraction of N2                              [m^3.m^-3]
#   X(2)  - volume fraction of O2                              [m^3.m^-3]
#   X(3)  - volume fraction of CO2                             [m^3.m^-3]
#   X(4)  - volume fraction of H2O                             [m^3.m^-3]
#   X(5)  - volume fraction of SO2                             [m^3.m^-3]
#   X(6)  - volume fraction of CO                              [m^3.m^-3]
#   X(7)  - volume fraction of H2                              [m^3.m^-3]
#   X(8)  - volume fraction of CH4                             [m^3.m^-3]
#   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
#   X(10) - volume fraction of C2H6                            [m^3.m^-3]
#   X(11) - volume fraction of C3H8                            [m^3.m^-3]
#   X(12) - volume fraction of C4H10                           [m^3.m^-3]
#   X(13) - volume fraction of H2S                             [m^3.m^-3]
#
# T - Temperature of mixture                                   [K]
#
## Used coeficients and functions
#
# a, b, c, d, e, f, g  - parameters for heat capacity calculation
# R - univerzal gas constant                                [J.mol^-1.K^-1]
#
## Output
#
# cp - Specific heat capacity of gas mixture                [J.mol^-1.K^-1]
#      defined for temperature T =<273,15;2500>.
#
## Copyright (C) 2011
# Authors     : Kukurugya Jan, Terpak Jan
# Organization: Technical University of Kosice
# e-mail      : jan.kukurugya@tuke.sk, jan.terpak@tuke.sk
# Revision    : 28.12.2011
#
#   1      2          3         4        5       6       7       8        9       10      11      12       13 
#   N2     O2         CO2       H2O      SO2     CO      H2      CH4      C2H4    C2H6    C3H8    C4H10    H2S   
#a=[2.2103 -3.4255000 4.9436000 -3.94790 -5.3108 1.48900 4.07830 -17.6680 -11.636 -18.620 -24.331 -31.7590 0.95438];
#b=[-3.8184 4.8470000 -6.264100 5.755700 9.09030 -2.9223 -8.0091 27.86180 25.5480 34.0620 46.5630 61.76330 -0.6875];
#c=[6.08270 1.1190000 5.3017000 0.931800 -2.3569 5.72450 8.21470 -12.0257 -16.097 -19.517 -29.394 -38.9150 4.0549];
#d=[-8.5309 4.2939000 2.5038000 7.222700 22.0440 -8.1762 -12.697 39.17620 66.2570 75.6580 118.890 158.4600 -0.3014];
#e=[1.38464 -0.068363 -0.021273 -0.73425 -2.5107 1.45690 1.75360 -3.61900 -7.8851 -8.2042 -13.763 -18.6000 0.3768];
#f=[-9.6258 -2.023300 -0.790000 4.955000 14.4630 -10.877 -12.028 20.26850 51.2520 50.6110 88.1480 119.9700 -2.2393];
#g=[2.51970 1.0390000 0.2849600 -1.33690 -3.3690 3.02800 3.36810 -4.97670 -13.703 -13.193 -23.429 -32.0167 0.308685];
```

```
        X=np.array([compo_db[2],compo_db[1],compo_db[0],0,0,0,0,0,0,0,0,0,0])

        if T < self.TK:
            T=self.TK

        if T > 2500:
            T=2500


#        cpT=np.zeros(len(X))
#
#        for i in range (0,13):
#            cpT[i]=a[i]*10**4*T**(-2)+b[i]*10**(2)*T**(-1) + c[i] + d[i]*10**(-3)*T
#            cpT[i]=cpT[i] + e[i]*10**-5*T**2 + f[i]*10**(-9)*T**(3) + g[i]*10**(-12)*T**4
#            cpT[i]=cpT[i]*X[i]*self.R

        cpT=(self.a_cp*10**4*T**(-2)+self.b_cp*10**(2)*T**(-1)+self.c_cp+self.d_cp*10**(-3)*T
+self.e_cp*10**-5*T**2+self.f_cp*10**(-9)*T**(3)+self.g_cp*10**(-12)*T**4)*X*self.R

        out=sum(cpT)/self.MM_db(compo_db)*1000

        return out



    def MM(self,compo_wb):
#
## Description
#
# Function for mollar mass calculation of gas with composition X.
#
# M = sum( M[i].X[i] )
#
## Inputs
#
# X - vector of volume fractions for next componemt of gas mixture:
#   X(1)  - volume fraction of N2                              [m^3.m^-3]
#   X(2)  - volume fraction of O2                              [m^3.m^-3]
#   X(3)  - volume fraction of CO2                             [m^3.m^-3]
#   X(4)  - volume fraction of H2O                             [m^3.m^-3]
#   X(5)  - volume fraction of SO2                             [m^3.m^-3]
#   X(6)  - volume fraction of CO                              [m^3.m^-3]
#   X(7)  - volume fraction of H2                              [m^3.m^-3]
#   X(8)  - volume fraction of CH4                             [m^3.m^-3]
#   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
#   X(10) - volume fraction of C2H6                            [m^3.m^-3]
#   X(11) - volume fraction of C3H8                            [m^3.m^-3]
#   X(12) - volume fraction of C4H10                           [m^3.m^-3]
#   X(13) - volume fraction of H2S                             [m^3.m^-3]
#
## Output
#
# Mm - Mollar mass of gas mixture                              [kg.mol-1]
#
## Copyright (C) 2011
# Authors     : Kukurugya Jan, Terpak Jan
# Organization: Technical University of Kosice
# e-mail      : jan.kukurugya@tuke.sk, jan.terpak@tuke.sk
# Revision    : 28.12.2011
#
#   1     2       3     4     5     6      7     8     9      10   11     12    13
#   N2    O2      CO2   H2O   SO2   CO     H2    CH4   C2H4   C2H6 C3H8   C4H10 H2S
#M=[28.01 31.9988 44.01 18.02 64.06 28.011 2.016 16.04 28.054 30.7 44.097 58.12 34.076];
        X=np.array([compo_wb[2],compo_wb[1],compo_wb[0],compo_wb[3],0,0,0,0,0,0,0,0,0])

#        Mc=np.zeros(len(X))
#
#        for i in range(0,13):
#            Mc[i]=M[i]*X[i]/1000.0

        Mc=self.M_Mm*X/1000.0
        out=sum(Mc)*1000

        return out

    def MM_db(self,compo_db):
#
## Description
#
# Function for mollar mass calculation of gas with composition X.
#
```

```
# M = sum( M[i].X[i] )
#
## Inputs
#
# X - vector of volume fractions for next componemt of gas mixture:
#   X(1)  - volume fraction of N2                              [m^3.m^-3]
#   X(2)  - volume fraction of O2                              [m^3.m^-3]
#   X(3)  - volume fraction of CO2                             [m^3.m^-3]
#   X(4)  - volume fraction of H2O                             [m^3.m^-3]
#   X(5)  - volume fraction of SO2                             [m^3.m^-3]
#   X(6)  - volume fraction of CO                              [m^3.m^-3]
#   X(7)  - volume fraction of H2                              [m^3.m^-3]
#   X(8)  - volume fraction of CH4                             [m^3.m^-3]
#   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
#   X(10) - volume fraction of C2H6                            [m^3.m^-3]
#   X(11) - volume fraction of C3H8                            [m^3.m^-3]
#   X(12) - volume fraction of C4H10                           [m^3.m^-3]
#   X(13) - volume fraction of H2S                             [m^3.m^-3]
#
## Output
#
# Mm - Mollar mass of gas mixture                              [kg.mol-1]
#
## Copyright (C) 2011
# Authors     : Kukurugya Jan, Terpak Jan
# Organization: Technical University of Kosice
# e-mail      : jan.kukurugya@tuke.sk, jan.terpak@tuke.sk
# Revision    : 28.12.2011
#
#   1     2       3     4     5     6      7     8     9      10   11     12    13
#   N2    O2      CO2   H2O   SO2   CO     H2    CH4   C2H4   C2H6 C3H8   C4H10 H2S
#M=[28.01 31.9988 44.01 18.02 64.06 28.011 2.016 16.04 28.054 30.7 44.097 58.12 34.076];
        X=np.array([compo_db[2],compo_db[1],compo_db[0],0,0,0,0,0,0,0,0,0,0])


#        Mc=np.zeros(len(X))
#
#        for i in range(0,13):
#            Mc[i]=M[i]*X[i]/1000.0

        Mc=self.M_Mm*X/1000.0
        out=sum(Mc)*1000

        return out

    def Rho(self,compo_wb,T):
#
# Function for density calculation of gas
# with composition X and temperature T.
#
#           ro0 . T0
# ro = ------------------
#               T
#
## Inputs
# X - vector of volume fractions for next componemt of gas mixture:
#   X(1)  - volume fraction of N2                              [m^3.m^-3]
#   X(2)  - volume fraction of O2                              [m^3.m^-3]
#   X(3)  - volume fraction of CO2                             [m^3.m^-3]
#   X(4)  - volume fraction of H2O                             [m^3.m^-3]
#   X(5)  - volume fraction of SO2                             [m^3.m^-3]
#   X(6)  - volume fraction of CO                              [m^3.m^-3]
#   X(7)  - volume fraction of H2                              [m^3.m^-3]
#   X(8)  - volume fraction of CH4                             [m^3.m^-3]
#   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
#   X(10) - volume fraction of C2H6                            [m^3.m^-3]
#   X(11) - volume fraction of C3H8                            [m^3.m^-3]
#   X(12) - volume fraction of C4H10                           [m^3.m^-3]
#   X(13) - volume fraction of H2S                             [m^3.m^-3]
#
# T - Temperature of mixture                                   [K]
#
## Used coeficients and functions
#
# ro0 - vector of components density for 20 oC:
#   ro0(1)  - density of N2                                    [kg.m^-3]
#   ro0(2)  - density of O2                                    [kg.m^-3]
#   ro0(3)  - density of CO2                                   [kg.m^-3]
#   ro0(4)  - density of H2O                                   [kg.m^-3]
#   ro0(5)  - density of SO2                                   [kg.m^-3]
#   ro0(6)  - density of CO                                    [kg.m^-3]
```

```
#   ro0(7)  - density of H2                                  [kg.m^-3]
#   ro0(8)  - density of CH4                                 [kg.m^-3]
#   ro0(9)  - density of C2H4                                [kg.m^-3]
#   ro0(10) - density of C2H6                                [kg.m^-3]
#   ro0(11) - density of C3H8                                [kg.m^-3]
#   ro0(12) - density of C4H10                               [kg.m^-3]
#   ro0(13) - density of H2S                                 [kg.m^-3]
#
## Output
#
# rho - density of gas mixture                               [kg.m^-3]
#       defined for temperature T =<273,15;2500>.
#
## Copyright (C) 2011
# Authors     : Kukurugya Jan, Terpak Jan
# Organization: Technical University of Kosice
# e-mail      : jan.kukurugya@tuke.sk, jan.terpak@tuke.sk
# Revision    : 28.12.2011
#
#     1     2     3     4     5     6     7      8     9     10    11    12    13
#     N2    O2    CO2   H2O   SO2   CO    H2     CH4   C2H4  C2H6  C3H8  C4H10 H2S
#ro0=[1.165 1.331 1.842 0.804 2.279 1.165 0.0866 0.668 1.260 1.264 1.882 2.489 1.434];
        X=np.array([compo_wb[2],compo_wb[1],compo_wb[0],compo_wb[3],0,0,0,0,0,0,0,0,0])

        if T < self.TK:
            T=self.TK
        if T > 2500:
            T=2500

#        ro=np.zeros(len(X))
#        for i in range(0,13):
#            ro[i]=ro0[i]*X[i]*(20.0 + self.TK)/T

        ro=self.ro0*X*(20.0 + self.TK)/T
        out=sum(ro)

        return out


    def Rho_db(self,compo_db,T):

        X=np.array([compo_db[2],compo_db[1],compo_db[0],0,0,0,0,0,0,0,0,0,0])

        if T < self.TK:
            T=self.TK
        if T > 2500:
            T=2500

#        ro=np.zeros(len(X))
#        for i in range(0,13):
#            ro[i]=ro0[i]*X[i]*(20.0 + self.TK)/T

        ro=self.ro0*X*(20.0 + self.TK)/T
        out=sum(ro)

        return out


#def Grashof(X,T,l,dT):
##
### Description
##
## Function for Grashof number (Grashof) calculation of gas mixture with composition X and temperature TG.
##
##           l^3 . g . beta . dT
## Grashoff = -------------------
##                 nu^2
##
### Inputs
##
## X - vector of volume fractions for next componemt of gas mixture:
##   X(1)  - volume fraction of N2                             [m^3.m^-3]
##   X(2)  - volume fraction of O2                             [m^3.m^-3]
##   X(3)  - volume fraction of CO2                            [m^3.m^-3]
##   X(4)  - volume fraction of H2O                            [m^3.m^-3]
##   X(5)  - volume fraction of SO2                            [m^3.m^-3]
##   X(6)  - volume fraction of CO                             [m^3.m^-3]
##   X(7)  - volume fraction of H2                             [m^3.m^-3]
##   X(8)  - volume fraction of CH4                            [m^3.m^-3]
##   X(9)  - volume fraction of C2H4                           [m^3.m^-3]
##   X(10) - volume fraction of C2H6                           [m^3.m^-3]
```

```
##   X(11) - volume fraction of C3H8                            [m^3.m^-3]
##   X(12) - volume fraction of C4H10                           [m^3.m^-3]
##   X(13) - volume fraction of H2S                             [m^3.m^-3]
##
## T  - temperature                                             [K]
##
## l  - characteristic length                                   [m]
##
## dT - temperature difference                                  [K]
##
### Used coeficients and functions
##
## g       - gravity acceleration                               [m.s^-2]
## beta(T) - thermal expansivity of gas                         [K^-1] 
## nu(X,T) - kinematic viscosity of gas                         [m^2.s^-1]
##
### Output
##
## Grashof - grashof number for gas mixture with composition X and temperature TG [1];
##
### Copyright (C) 2011
## Authors     : Terpak Jan, Kukurugya Jan
## Organization: Technical University of Kosice
## e-mail      : jan.terpak@tuke.sk, jan.kukurugya@tuke.sk
## Revision    : 28.12.2011
##
#    out=(l^3*g*beta(T)*dT)/(nu(X,T)^2)
#    return out
#
#
#
#def Reynolds(self,X,T,l,w):
##
### Description
##
## Function for Reynolds number calculation of gas 
## mixture with composition X and temperature T.
##
##              l . w
## Reynolds = ----------
##               nu
##
### Inputs
##
## X - vector of volume fractions for next componemt of gas mixture:
##   X(1)  - volume fraction of N2                              [m^3.m^-3]
##   X(2)  - volume fraction of O2                              [m^3.m^-3]
##   X(3)  - volume fraction of CO2                             [m^3.m^-3]
##   X(4)  - volume fraction of H2O                             [m^3.m^-3]
##   X(5)  - volume fraction of SO2                             [m^3.m^-3]
##   X(6)  - volume fraction of CO                              [m^3.m^-3]
##   X(7)  - volume fraction of H2                              [m^3.m^-3]
##   X(8)  - volume fraction of CH4                             [m^3.m^-3]
##   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
##   X(10) - volume fraction of C2H6                            [m^3.m^-3]
##   X(11) - volume fraction of C3H8                            [m^3.m^-3]
##   X(12) - volume fraction of C4H10                           [m^3.m^-3]
##   X(13) - volume fraction of H2S                             [m^3.m^-3]
##
## T - temperature                                              [K]
##
## w - speed of gas flow                                        [m.s^-1]
##
## l - characteristic length                                    [m]
##
### Used coeficients and functions
##
## nu(X,T) - kinematic viscosity of gas mixture                 [m^2.s^-1]
##
### Output
##
## Reynolds - Reynolds number                                   [1]
##
### Copyright (C) 2011
## Authors     : Terpak Jan, Kukurugya Jan
## Organization: Technical University of Kosice
## e-mail      : jan.terpak@tuke.sk, jan.kukurugya@tuke.sk
## Revision    : 28.12.2011
##
#    out=(l*w)/nu(X,T)
#    return out
```

```
#
#
#
#def Nusselt(self,X,T,l,k):
##
### Description
##
## Function for Nusselt number calculation of gas
## mixture with composition X and temperature T.
##
##             k.l
## Nusselt = ------------
##            lambda
##
### Inputs
##
## X - vector of volume fractions for next componemt of gas mixture:
##   X(1)  - volume fraction of N2                             [m^3.m^-3]
##   X(2)  - volume fraction of O2                             [m^3.m^-3]
##   X(3)  - volume fraction of CO2                            [m^3.m^-3]
##   X(4)  - volume fraction of H2O                            [m^3.m^-3]
##   X(5)  - volume fraction of SO2                            [m^3.m^-3]
##   X(6)  - volume fraction of CO                             [m^3.m^-3]
##   X(7)  - volume fraction of H2                             [m^3.m^-3]
##   X(8)  - volume fraction of CH4                            [m^3.m^-3]
##   X(9)  - volume fraction of C2H4                           [m^3.m^-3]
##   X(10) - volume fraction of C2H6                           [m^3.m^-3]
##   X(11) - volume fraction of C3H8                           [m^3.m^-3]
##   X(12) - volume fraction of C4H10                          [m^3.m^-3]
##   X(13) - volume fraction of H2S                            [m^3.m^-3]
##
## T - Temperature                                             [K]
##
## l - characteristic length                                   [m]
##
## k - coefficient of heat transfer by convection              [W.m^-2.K^-1]
##
### Used coeficients and functions
##
## lambda  (X,T)         - thermal conductivity                [W.m^-1.K^-1]
##
### Output
##
## Nusselt - Nusselt number                                    [1]
##
### Copyright (C) 2011
## Authors     : Terpak Jan, Kukurugya Jan
## Organization: Technical University of Kosice
## e-mail      : jan.terpak@tuke.sk, jan.kukurugya@tuke.sk
## Revision    : 28.12.2011
##
#    out = (k*l)/Lambda(X,T)
#    return out
#
#
#
#def Prandtl(self,X,T):
#    #
#    ## Description
#    #
#    # Function for Prandtl number calculation of gas
#    # mixture with composition X and temperature T.
#    #
#    #             nu
#    # Prandtl = -----
#    #             a
#    #
#    ## Inputs
#    #
#    # X - vector of volume fractions for next componemt of gas mixture:
#    #   X(1)  - volume fraction of N2                              [m^3.m^-3]
#    #   X(2)  - volume fraction of O2                              [m^3.m^-3]
#    #   X(3)  - volume fraction of CO2                             [m^3.m^-3]
#    #   X(4)  - volume fraction of H2O                             [m^3.m^-3]
#    #   X(5)  - volume fraction of SO2                             [m^3.m^-3]
#    #   X(6)  - volume fraction of CO                              [m^3.m^-3]
#    #   X(7)  - volume fraction of H2                              [m^3.m^-3]
#    #   X(8)  - volume fraction of CH4                             [m^3.m^-3]
#    #   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
#    #   X(10) - volume fraction of C2H6                            [m^3.m^-3]
#    #   X(11) - volume fraction of C3H8                            [m^3.m^-3]
```

```
#    #   X(12) - volume fraction of C4H10                           [m^3.m^-3]
#    #   X(13) - volume fraction of H2S                             [m^3.m^-3]
#    #
#    # T - temperature                                              [K]
#    #
#    ## Used coeficients and functions
#    #
#    # a(X,T)  - thermal diffusivity                                [m^2.s^-1]
#    # nu(X,T) - kinematic viscosity                                [m^2.s^-1]
#    #
#    ## Output
#    #
#    # Prandtl - Prandtl number                                     [1]
#    #
#    ## Copyright (C) 2011
#    # Authors     : Terpak Jan, Kukurugya Jan
#    # Organization: Technical University of Kosice
#    # e-mail      : jan.terpak@tuke.sk, jan.kukurugya@tuke.sk
#    # Revision    : 28.12.2011
#    #
#    out=nu(X,T)/a(X,T)
#
#    return out
#
#
#


#def alpha(X,Tg,Tm,type,param,coef):
#    #
#    #% Description
#    #
#    # Function for calculation of coefficient of heat transfer by convection
#    # of gas mixture with composition X and temperature Tg.
#    #
#    #                              Pr(Tg)^a5         lambda
#    # alpha = a1.Gr^a2.Pr^a3.Re^a4.---------.eps^a6.--------
#    #                              Pr(Tm)^a5            l
#    #
#    #% Inputs
#    #
#    # X - vector of volume fractions for next componemt of gas mixture:
#    #   X(1)  - volume fraction of N2                              [m^3.m^-3]
#    #   X(2)  - volume fraction of O2                              [m^3.m^-3]
#    #   X(3)  - volume fraction of CO2                             [m^3.m^-3]
#    #   X(4)  - volume fraction of H2O                             [m^3.m^-3]
#    #   X(5)  - volume fraction of SO2                             [m^3.m^-3]
#    #   X(6)  - volume fraction of CO                              [m^3.m^-3]
#    #   X(7)  - volume fraction of H2                              [m^3.m^-3]
#    #   X(8)  - volume fraction of CH4                             [m^3.m^-3]
#    #   X(9)  - volume fraction of C2H4                            [m^3.m^-3]
#    #   X(10) - volume fraction of C2H6                            [m^3.m^-3]
#    #   X(11) - volume fraction of C3H8                            [m^3.m^-3]
#    #   X(12) - volume fraction of C4H10                           [m^3.m^-3]
#    #   X(13) - volume fraction of H2S                             [m^3.m^-3]
#    #
#    # Tg - temperature of gas mixture                              [K]
#    #
#    # Tm - temperature of material or wall                         [K]
#    #
#    # type - definition of convecton type:
#    #   0 - user set the coefficients in vector 'coef';
#    #   1 - free convection in unlimited space,      set up param(1);
#    #   2 - free convection in   limited space,      set up param(1);
#    #   3 - forced convection along the plain board, set up param(1),(2);
#    #   4 - forced convection in tubes and canals,   set up param(1),(2);
#    #   5 - forced convection across the tube,       set up param(1),(2),(3);
#    #
#    # param - vector of parameters:
#    #         param(1) - specific length parameter       (l)       [m]
#    #         param(2) - speed of flowing gas            (w)       [m.s^-1]
#    #         param(3) - angle of flowing of gas in tube (fi)      [o]
#    #
#    # coef - vector of coefficient for alpha calculation for type=0:
#    #        coef(1) is a(1),
#    #        coef(2) is a(2),
#    #        coef(3) is a(3),
#    #        coef(4) is a(4),
#    #        coef(5) is a(5),
#    #        coef(6) is a(6).
#    #
```

```
#    #% Used coeficients and functions
#    #
#    # lambda  (X,T)         - thermal conductivity [W.m^-1.K^-1]
#    # Prandtl (X,T)         - Prandtl number;
#    # Grashof (X,T,l,dT)    - Grashof number;
#    # Reynolds(X,T,w,l)     - Reynolds number;
#    #
#    #% Output
#    #
#    # alpha - coefficient of heat transfer by convection         [W.m^-2.K^-1]
#    #
#    #% Copyright (C) 2011
#    # Authors     : Terpak Jan, Kukurugya Jan
#    # Organization: Technical University of Kosice
#    # e-mail      : jan.terpak@tuke.sk, jan.kukurugya@tuke.sk
#    # Revision    : 28.12.2011
#    #
#    l = param(1)
#    w = param(2)
#    fi = param(3)
#
#    if fi < 30:
#        fi = 30
#
#    if fi > 90:
#        fi = 90
#
#    eps = 1.0 - 0.52 * np.cos(fi)**3
#    dT = abs(Tg - Tm)
#    Ta = (Tg + Tm)/2
#
#    Gr = Grashof(X, Ta, l, dT)
#    Pr = Prandtl(X, Ta)
#    Re = Reynolds(X, Ta, l, w)
#    Prg = Prandtl(X, Tg)
#    Prm = Prandtl(X, Tm)
#    la = Lambda(X, Ta)
#
#    for i in range(0,6):
#        a[i]= 0
#
#    switch type
#
#    case 0
#    for i in range (0,6):
#        a[i] = coef[i]
#
#    case 1
#    GrPr=Gr*Pr
#
#    if GrPr < 1*10**(-3)):
#        a[1]=0.450
#
#    if GrPr >= 1*10**(-3) and GrPr < 5*10**2:
#        a[1]=1.180
#        a[2]=1/8
#        a[3]=1/8
#
#    if GrPr >= (5*10^(2)) && GrPr < 2*10^7:
#        a[1]=0.540
#        a[2]=1/4
#        a[3]=1/4
#
#    if GrPr >= (2*10^(7))
#        a[1]=0.135
#        a[2]=1/3
#        a[3]=1/3
#
#    case 2
#    GrPr= Gr*Pr
#
#    if GrPr < 1*10**3:
#        a[1]=1
#
#    if GrPr >= 1*10**3 and GrPr < 1*10**6:
#        a[1]=0.105
#        a[2]=0.3
#        a[3]=0.3
#
#    if GrPr >= 1*10**6:
#        a[1]=0.400
```

```
#        a[2]=0.2
#        a[3]=0.2
#
#    case 3
#    if Re <= 2300:
#        a[1]=0.570
#        a[4]=0.5;
#    
#    if Re > 2300 and Re < 10**4
#        a[1]=0.301
#        a[4]=0.65
#    
#    if Re >= 10**4
#        a[1]=0.032
#        a[4]=0.8
#
#    case 4
#    if Re <= 2300:
#        a[1]=0.15
#        a[2]=0.1
#        a[3]=0.43
#        a[4]=0.33
#        a[5]=0.25
#
#    if Re > 2300 and Re < 10**4:
#        a[1]=0.1
#        a[3]=0.4
#        a[4]=0.5
#        a[5]=0.25
#        
#    if Re >= 10**4:
#        a[1]=0.023
#        a[3]=0.4
#        a[4]=0.8
#        a[5]=0.25
#  
#    case 5
#      a(6)=1
#    if Re <= 5:
#        Re = 5
#        
#    if Re > 5 and Re < 1000:
#        a[1]=0.5
#        a[3]=0.38
#        a[4]=0.5
#        a[5]=0.25
#
#    if Re >=1000 and Re < 2*10**5:
#        a[1]=0.25
#        a[3]=0.38
#        a[4]=0.6
#        a[5]=0.25
#        
#    if Re >= 2*10**5
#        a[1]=0.023
#        a[3]=0.37
#        a[4]=0.8
#        a[5]=0.25
#
#    out = a[1]*Gr**a[2]*Pr**a[3]*Re**a[4]*(Prg/Prm)**a[5]*np.eps**a[6]*(la/l)
#
#    return out
```

B.8 Result class

```
# -*- coding: utf-8 -*-
"""
Created on Sat May 30 11:44:00 2020

@author: Salva
"""

import numpy as np
from Constants import Nyears,P_ref,T_FG_dryer,compo_db_air,Ratio_H2O_air
from Constants import T_air_mixer,a_temp,b_temp
from GetEconomics import NetPresentValue
from Boiler2 import fuel_properties, FG_Composition
from CoolProp.CoolProp import PropsSI
from ExcessAir import Exc_air
from MyProperties_Gas import MyProperties_Gas_2
from Mixer import Mix

from GetProps import Get_hFG_IN

Air=MyProperties_Gas_2(compo_db_air,Ratio_H2O_air/(1+Ratio_H2O_air),T_air_mixer,P_ref)
class Result():
    
    def __init__(self,Pn,MCiad,FuelC,FixC,TFGoutBoilernom,TFGinDryernom,
                 
                 MassFuelininom,Effininom,
                 MassFuelnom_array,MCfinadnom_array,Effinom_array,
                 MassFG1nom,menom,HeatExcnom,
                 TFGoutnom_array,
                 MassMixnom, MassAirnom,
                 Tau_array,
                 
                 MassFuelini_array,Effini_array,
                 MassFuel_matrix,MCfinad_matrix,Effi_matrix,
                 MassFG1_matrix,me_matrix,HeatExc_matrix,
                 TFGout_matrix,
                 MassMix_matrix,MassAir_matrix,
                 Load,P_BoilerCurve,Hours,
                 
                 RhoFG_array,Ac_array,PartialFG_array):
        
#__Main Parameters
        self.P_Boiler=Pn
        self.MCini_ad=MCiad
        self.FuelC=FuelC
        self.FixC=FixC
    #Temperatures
        self.T_FG_boiler=TFGoutBoilernom
        self.T_FG_dryer=TFGinDryernom
    #Main arrays for calculations
        self.PartialFG=np.round(PartialFG_array,2)
        #Full load: running hours
        self.Tau=Tau_array
        #Load curve load: Load curve and boiler curve
        self.Hours=Hours
        self.Load=Load
        self.P_BoilerCurve=P_BoilerCurve
        self.P_BoilerFactor=P_BoilerCurve/self.P_Boiler
    #Main result: Conveyor surface
        self.RhoFG=RhoFG_array
        self.Ac=Ac_array

# =============================================================================
# #Full Load results
# =============================================================================
        self.FullLoad=FullLoad(self.P_Boiler,self.MCini_ad,
                               MassFuelininom,Effininom,
                               MassFuelnom_array,MCfinadnom_array,Effinom_array,MassFG1nom,menom,
                               HeatExcnom,TFGoutnom_array,
                               MassMixnom,MassAirnom,
                               self.Tau,
```

```
                               self.PartialFG,self.RhoFG,self.Ac)

# ============================================================================
# #Load curve results
# ============================================================================
        self.LoadCurve=LoadCurve(self.P_Boiler,self.MCini_ad,
                                 MassFuelini_array,Effini_array,
                                 MassFuel_matrix,MCfinad_matrix,Effi_matrix,MassFG1_matrix,me_matrix,
                                 HeatExc_matrix,TFGout_matrix,
                                 MassMix_matrix,MassAir_matrix,
                                 self.Load,self.P_BoilerFactor,self.Hours,
                                 self.PartialFG,self.RhoFG,self.Ac)

class FullLoad():
    def __init__ (self,Pn,MCini_ad,
                  MassFuelininom,Effininom,
                  MassFuelnom_array,MCfinadnom_array,Effinom_array,MassFG1nom,menom,
                  HeatExcnom,TFGoutnom_array,
                  MassMixnom,MassAirnom,
                  Tau_array,
                  PartialFG_array,RhoFG,Ac):

        self.PartialFG=np.round(PartialFG_array,2)
        self.Tau=Tau_array
        self.T_FG_boiler=a_temp*1+b_temp

# ============================================================================
# #__Initial situation / Without dryer
# ============================================================================
    #Fuel properties
        [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,
         LHV_Fuel_daf,HHV_Fuel_daf]=fuel_properties(MCini_ad,"fuel_properties")
    #FG properties
        [compo_db_fg,compo_wb_fg,CO_d_fg,m_fg,m_air,m_H2O]=FG_Composition(1,MCini_ad)
        FG=MyProperties_Gas_2(compo_db_fg,compo_wb_fg[3],self.T_FG_boiler,P_ref)
    #Results
        #Mass Fuel and efficiency
        self.MassFuelini_daf=MassFuelininom    #kg Fuel daf/s
        self.Ashini_ad=Ash_Fuel_ad             #kg Ash/kg Fuel ad
        self.MassFuelini_ad=self.MassFuelini_daf/(1-MCini_ad/100-self.Ashini_ad/100)
        self.Effini=Effininom
        #LHV
        self.LHVini_daf=LHV_Fuel_daf           #J/kg Fuel daf
        self.LHVini_ad=LHV_Fuel_ad             #J/kg Fuel ad
        #Especific mass air in the boiler
        self.m_air_esp_daf_ini=m_air           #kg air /kg Fuel daf
        self.m_air_esp_ad_ini=m_air/(1+MC_Fuel_daf/100+Ash_Fuel_daf/100) #kgAir/kg Fuel ad
        #Water Vapor content in the FG
        self.MC_FGini_wb=FG.X_H2O              #kgH2O/kg FG total
        self.MC_FGini_db=FG.X_H2O/(1-FG.X_H2O)  #kgH2O/kg FG db

# ============================================================================
# #__Energetic results / with dryer
# ============================================================================
    #Amount of FG and fuel in the dryer
        self.MassFuel_daf=MassFuelnom_array
        self.MassFuel_ad=self.MassFuel_daf/(1-MCini_ad/100-self.Ashini_ad/100)
        #FG in dryer = Recovered gas from the boiler + Air
        self.MassFG_Boiler=MassFG1nom  #FG produced in the boiler
        self.MassFG_Mixer=np.multiply(self.PartialFG,self.MassFG_Boiler)
        self.MassAir=MassAirnom        #Air added to the FG from boiler
        self.MassFG_Dryer=MassMixnom    #Mixing of FG + Air in the dryer
    #Final fuel humidity in the boiler and efficiency
        self.MCfin_ad=MCfinadnom_array
        self.Effi=Effinom_array

    #Initialization of ecnergy variables arrays
        #Fuel properties as delivered
        self.MassFuel_Boiler_ad=np.zeros(len(self.PartialFG))   #kg Fuel ad/s
        self.Ash_Boiler_ad=np.zeros(len(self.PartialFG))        #kg Ash/kg Fuel ad
```

```
            #LHV
        self.LHV_daf=np.zeros(len(self.PartialFG))              #J/kg Fuel daf
        self.LHV_ad=np.zeros(len(self.PartialFG))               #J/kg Fuel ad

        #Especific mass air in the boiler
        self.m_air_esp_daf=np.zeros(len(self.PartialFG))        #kg air /kg Fuel daf
        self.m_air_esp_ad=np.zeros(len(self.PartialFG))         #kg air/ kg Fuel ad
        self.ExcessAir=np.zeros(len(self.PartialFG))
        #Water Vapor content in the FG
            #Outlet of the boiler
        self.MC_FG_boiler_wb=np.zeros(len(self.PartialFG))      #kgH2O/kg FG total
        self.MC_FG_boiler_db=np.zeros(len(self.PartialFG))      #kgH2O/kg FG db
            #Inlet of the dryer
        self.MC_FG_dryer_wb=np.zeros(len(self.PartialFG))       #kgH2O/kg FG total
        self.MC_FG_dryer_db=np.zeros(len(self.PartialFG))       #kgH2O/kg FG db
        #Enthalpy and energy of the FG
        self.h_FG_wet=np.zeros(len(self.PartialFG))             #kJ/kg FG db
        self.E_FG_wet_ava=np.zeros(len(self.PartialFG))         #kW FG
        self.E_FG_wet_used=np.zeros(len(self.PartialFG))        #kW FG
        #Global efficiency system Boiler + Dryer
        self.EffiGlobal=np.zeros(len(self.PartialFG))

#Calculation of energy parameters
        for i in range(len(self.PartialFG)):
            [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,
             LHV_Fuel_daf,HHV_Fuel_daf]=fuel_properties(self.MCfin_ad[i],"fuel_properties")
            [compo_db_fg,compo_wb_fg,CO_d_fg,m_fg,m_air,
             m_H2O]=FG_Composition(1,self.MCfin_ad[i])

            self.Ash_Boiler_ad[i]=Ash_Fuel_ad
            self.MassFuel_Boiler_ad[i]=self.MassFuel_daf[i]/(1-self.MCfin_ad[i]/100-self.Ash_Boiler_ad[i]/
100)

            self.LHV_daf[i]=LHV_Fuel_daf
            self.LHV_ad[i]=LHV_Fuel_ad

            self.m_air_esp_daf[i]=m_air
            self.m_air_esp_ad[i]=m_air/(1+MC_Fuel_daf/100+Ash_Fuel_daf/100)

            FG=MyProperties_Gas_2(compo_db_fg,compo_wb_fg[3],self.T_FG_boiler,P_ref)
            self.MC_FG_boiler_wb[i]=FG.X_H2O
            self.MC_FG_boiler_db[i]=FG.X_H2O/(1-FG.X_H2O)

#            if T_FG_dryer!=T_FG_boiler:
#                FGmix,MassFG21=Mix(self.MassAir[i],self.MassFG_Mixer[i],Air,FG,T_FG_dryer)
#                self.MC_FG_dryer_wb[i]=FGmix.X_H2O
#                self.MC_FG_dryer_db[i]=FGmix.X_H2O/(1-FGmix.X_H2O)
#            else:
#                self.MC_FG_dryer_wb[i]=self.MC_FG_boiler_wb[i]
#                self.MC_FG_dryer_db[i]=self.MC_FG_boiler_db[i]

            self.MC_FG_dryer_wb[i]=self.MC_FG_boiler_wb[i]
            self.MC_FG_dryer_db[i]=self.MC_FG_boiler_db[i]

            self.h_FG_wet[i]=Get_hFG_IN(FG)
            self.E_FG_wet_ava[i]=self.h_FG_wet[i]*self.MassFG_Boiler[i]
            self.E_FG_wet_used[i]=self.h_FG_wet[i]*self.MassFG_Dryer[i]
            self.EffiGlobal[i]=Get_EffiGlobal(Pn,self.LHVini_ad,self.MassFuel_ad[i],self.m_air_esp_ad[i],1)

        for mc in range(len(self.PartialFG)):
            self.ExcessAir[mc]=Exc_air(self.MCfin_ad[mc],1)

        self.me=menom                           #kg H2O/s
        self.HeatExc=HeatExcnom                 #kW
        self.Drying_Agent=self.MassFG_Dryer/self.MassFuel_daf      #kg Mix FG db/kg Fuel daf
        self.Drying_Agent_FG=self.MassFG_Mixer/self.MassFuel_daf   #kg FG db/kg Fuel daf
        self.Heat_Consum=self.HeatExc/self.me   #kJ/kgH2O
        self.Heat_Consum_e=self.HeatExc/self.MassFuel_daf          #kJ/kg Fuel daf
        self.TFGout=TFGoutnom_array

```

```
#Demand Covered and Load Factor
        self.LF=np.zeros(len(self.Tau))
        for t in range(len(self.Tau)):
            self.LF[t]=LoadFactor(Pn,self.Tau[t],Pn)
            
        self.DemCov=1
        
#Initialization of economic arrays and matrixs
        #Al monetary values in [k€]
        #All time values in [years]
        self.Years=np.arange(0,Nyears+1,1)
        
        self.PartialFG15max=np.zeros(len(self.Tau))
        self.NPV15max=np.zeros(len(self.Tau))
        
        self.PartialFG15max_max=np.zeros(len(self.Tau))
        self.NPV15max_max=np.zeros(len(self.Tau))

        self.PartialFG15max_min=np.zeros(len(self.Tau))
        self.NPV15max_min=np.zeros(len(self.Tau))
        
        self.PartialFG5max=np.zeros(len(self.Tau))
        self.NPV5max=np.zeros(len(self.Tau))
        
        self.PartialFG5max_max=np.zeros(len(self.Tau))
        self.NPV5max_max=np.zeros(len(self.Tau))

        self.PartialFG5max_min=np.zeros(len(self.Tau))
        self.NPV5max_min=np.zeros(len(self.Tau))

        
        self.MCmin_opt_eco=np.zeros(len(self.Tau))
        self.Effimax_opt_eco=np.zeros(len(self.Tau))

        #NPV[Tau_index,year,FGindex]
        self.NPVHolmberg=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        self.NPVBrammer=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        self.NPVBelt=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        self.NPVSolid=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        
        self.NPVSolid_max=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        self.NPVBelt_max=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        self.NPVSolid_min=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        self.NPVBelt_min=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        
        self.NPV=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        self.NPV_max=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        self.NPV_min=np.zeros((len(self.Tau),Nyears+1,len(self.PartialFG)))
        
        #Payback period
        self.RTHolmberg=np.zeros((len(self.Tau),len(self.PartialFG)))
        self.RTBrammer=np.zeros((len(self.Tau),len(self.PartialFG)))
        self.RTBelt=np.zeros((len(self.Tau),len(self.PartialFG)))
        self.RTSolid=np.zeros((len(self.Tau),len(self.PartialFG)))
        self.RT=np.zeros((len(self.Tau),len(self.PartialFG)))
        #Revenues
        self.AnnualRev_Belt=np.zeros((len(self.Tau),len(self.PartialFG)))
        self.AnnualRev_Solid=np.zeros((len(self.Tau),len(self.PartialFG)))
        self.AnnuaRev=np.zeros((len(self.Tau),len(self.PartialFG)))
        #O&M Cost
        self.Opex_Belt=np.zeros((len(self.Tau),len(self.PartialFG)))
        self.Opex_Solid=np.zeros((len(self.Tau),len(self.PartialFG)))
        self.Opex=np.zeros((len(self.Tau),len(self.PartialFG)))
        
        self.FuelCost=np.zeros((len(self.Tau),len(self.PartialFG)))
        self.ElecCost=np.zeros((len(self.Tau),len(self.PartialFG)))
        
        self.FixC_Belt=np.zeros(len(self.PartialFG))
        self.FixC_Solid=np.zeros(len(self.PartialFG))
        self.FixC=np.zeros(len(self.PartialFG))
```

```
        #Investment cost
        self.Capex_Belt=np.zeros(len(self.PartialFG))
        self.Capex_Solid=np.zeros(len(self.PartialFG))
        self.Capex=np.zeros(len(self.PartialFG))
        
#Net present value loops
        for i in range(len(self.PartialFG)):
            for j in range(len(self.Tau)):
                [NPV_Holmberg,NPV_Brammer,NPV_SolidBand,NPV_BeltBand, 
                 NPV_SolidBand_max,NPV_BeltBand_max,NPV_SolidBand_min,NPV_BeltBand_min,
                 RT_Holmberg,RT_Brammer,RT_SolidBand,RT_BeltBand,RT,
                 AnnualRevenue, FuelCostIni, CapitalCost_index,
                 CapitalCost_array,OPEX_array,ElecCost,FuelCost, FixedCost
                 ]=NetPresentValue(self.Tau[j],RhoFG[i],Ac[i],
                 self.MassFG_Dryer[i],self.MassFuel_daf[i], self.MassFuelini_daf)
        
                self.NPVHolmberg[j,:,i]=NPV_Holmberg
                self.NPVBrammer[j,:,i]=NPV_Brammer
                self.NPVSolid[j,:,i]=NPV_SolidBand
                self.NPVBelt[j,:,i]=NPV_BeltBand
                self.NPV[j,:,i]=(self.NPVBelt[j,:,i]+self.NPVSolid[j,:,i])/2
                
                self.NPVSolid_max[j,:,i]=NPV_SolidBand_max
                self.NPVBelt_max[j,:,i]=NPV_BeltBand_max
                self.NPV_max[j,:,i]=(self.NPVBelt_max[j,:,i]+self.NPVSolid_max[j,:,i])/2
                
                self.NPVSolid_min[j,:,i]=NPV_SolidBand_min
                self.NPVBelt_min[j,:,i]=NPV_BeltBand_min
                self.NPV_min[j,:,i]=(self.NPVBelt_min[j,:,i]+self.NPVSolid_min[j,:,i])/2
                
                self.RTHolmberg[j,i]=RT_Holmberg
                self.RTBrammer[j,i]=RT_Brammer
                self.RTBelt[j,i]=RT_BeltBand
                self.RTSolid[j,i]=RT_SolidBand
                self.RT[j,i]=RT
                
                self.AnnualRev_Belt[j][i]=AnnualRevenue[2]
                self.AnnualRev_Solid[j][i]=AnnualRevenue[3]
                
                self.Opex_Belt[j][i]=OPEX_array[2]
                self.Opex_Solid[j][i]=OPEX_array[3]
                
                self.FuelCost[j][i]=FuelCost
                self.ElecCost[j][i]=ElecCost
                
            self.FixC_Belt[i]=FixedCost[2]
            self.FixC_Solid[i]=FixedCost[3]

            self.Capex_Belt[i]=CapitalCost_array[2]
            self.Capex_Solid[i]=CapitalCost_array[3]
        #smoothing RT
        for i in range(len(self.PartialFG)):
            for j in range(len(self.Tau)-2):
                if self.RT[j+1,i]>1.1*self.RT[j,i]:
                    A=(self.Tau[j+2]-self.Tau[j+1])/(self.Tau[j+2]-self.Tau[j])
                    self.RT[j+1,i]=-A*(self.RT[j+2,i]-self.RT[j,i])+self.RT[j+2,i]
#                    self.RT[j+1,i]=(self.RT[j,i]+self.RT[j+2,i])/2
            
        self.FixC=(self.FixC_Belt+self.FixC_Solid)/2
        self.Capex=(self.Capex_Belt+self.Capex_Solid)/2
        self.Opex=(self.Opex_Belt+self.Opex_Solid)/2
        self.AnnualRev=(self.AnnualRev_Belt+self.AnnualRev_Solid)/2

        for i in range(len(self.Tau)):
            #Year 15
            ArgMax=np.argmax(self.NPV[i,Nyears,:])
            self.NPV15max[i]=self.NPV[i,Nyears,ArgMax]
            self.PartialFG15max[i]=self.PartialFG[ArgMax]
            self.MCmin_opt_eco[i]=self.MCfin_ad[ArgMax]
            self.Effimax_opt_eco[i]=self.Effi[ArgMax]
```

```
#            self.AnnualRevmax[i]=self.AnnualRev[ArgMaxRev]
            #CAPEX=70%
            ArgMax1=np.argmax(self.NPV_max[i,Nyears,:])
            self.NPV15max_max[i]=self.NPV_max[i,Nyears,ArgMax1]
            self.PartialFG15max_max[i]=self.PartialFG[ArgMax1]
            #CAPEX=130%
            ArgMax2=np.argmax(self.NPV_min[i,Nyears,:])
            self.NPV15max_min[i]=self.NPV_min[i,Nyears,ArgMax2]
            self.PartialFG15max_min[i]=self.PartialFG[ArgMax2]
            
            #Year5
            ArgMax=np.argmax(self.NPV[i,4,:])
            self.NPV5max[i]=self.NPV[i,4,ArgMax]
            self.PartialFG5max[i]=self.PartialFG[ArgMax]
#            self.AnnualRevmax[i]=self.AnnualRev[ArgMaxRev]
            #CAPEX=70%
            ArgMax1=np.argmax(self.NPV_max[i,4,:])
            self.NPV5max_max[i]=self.NPV_max[i,4,ArgMax1]
            self.PartialFG5max_max[i]=self.PartialFG[ArgMax1]
            #CAPEX=130%
            ArgMax2=np.argmax(self.NPV_min[i,4,:])
            self.NPV5max_min[i]=self.NPV_min[i,4,ArgMax2]
            self.PartialFG5max_min[i]=self.PartialFG[ArgMax2]

            
class LoadCurve():
    def __init__(self,Pn,MCini_ad,
                 MassFuelini_array,Effini_array,
                 MassFuel_matrix,MCfinad_matrix,Effi_matrix,MassFG1_matrix,me_matrix,
                 HeatExc_matrix,TFGout_matrix,
                 MassMix_matrix,MassAir_matrix,
                 LoadCurve,P_BoilerFactor,Hours,
                 PartialFG_array,RhoFG,Ac):
        
        self.PartialFG=np.round(PartialFG_array,2)
        self.Load=LoadCurve
        self.P_BoilerFactor=P_BoilerFactor
        self.Hours=Hours
        self.Hours_cum=np.cumsum(self.Hours)
        
        self.TFGout=TFGout_matrix

#__Initial situation / Without dryer  
        [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,
         LHV_Fuel_daf,HHV_Fuel_daf]=fuel_properties(MCini_ad,"fuel_properties")
        
        self.MassFuelini_daf=MassFuelini_array #kgdaf/s
        self.Ashini_ad=Ash_Fuel_ad #kgAsh/kgFuelad
        self.MassFuelini_ad=self.MassFuelini_daf/(1-MCini_ad/100-self.Ashini_ad/100)
        self.Effini=Effini_array
        self.LHVini_daf=LHV_Fuel_daf #J/kg Fuel daf
        self.LHVini_ad=LHV_Fuel_ad #J/kg Fuel ad
        
        self.m_air_esp_daf_ini=np.zeros(len(self.Hours))
        self.m_air_esp_ad_ini=np.zeros(len(self.Hours))
        self.T_FG_boiler=np.zeros(len(self.Hours))
        for h in range(len(self.Hours)):
            if self.P_BoilerFactor[h]>0:
                self.T_FG_boiler[h]=a_temp*self.P_BoilerFactor[h]+b_temp
        
        for l in range(len(self.P_BoilerFactor)):
            if self.P_BoilerFactor[l]>0:
                [compo_db_fg,compo_wb_fg,CO_d_fg,m_fg,m_air,m_H2O]=FG_Composition(
                        self.P_BoilerFactor[l],MCini_ad)
                self.m_air_esp_daf_ini[l]=m_air #kg air /kg Fuel daf
                self.m_air_esp_ad_ini[l]=m_air/(1+MC_Fuel_daf/100+Ash_Fuel_daf/100) #kgAir/kgFuelad
#__Energetic results    
        self.MassFG_Boiler=MassFG1_matrix #db

        self.MassFuel_daf=MassFuel_matrix #db
```

```
        self.MassFuel_ad=self.MassFuel_daf/(1-MCini_ad/100-self.Ashini_ad/100) #Inlet of the system D+B kg/s
        self.MassFG_Mixer=np.multiply(self.PartialFG,self.MassFG_Boiler)
        self.MassFG_Dryer=MassMix_matrix
        self.MassAir=MassAir_matrix
        self.MCfin_ad=MCfinad_matrix
        self.Effi=Effi_matrix
#Initialization of ecnergy variables arrays and matrix
        self.MassFuel_Boiler_ad=np.zeros((len(self.Hours),len(self.PartialFG)))
        
        self.Ash_Boiler_ad=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.LHV_daf=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.LHV_ad=np.zeros((len(self.Hours),len(self.PartialFG)))
        
        self.h_FG_dry_boiler=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.h_FG_dry_out=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.h_FG_boiler=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.h_FG_out=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.E_FG_dry_ava=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.E_FG_dry_used=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.E_ava=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.E_used=np.zeros((len(self.Hours),len(self.PartialFG)))
        
        self.MC_FG_boiler_wb=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.MC_FG_boiler_db=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.MC_FG_dryer_wb=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.MC_FG_dryer_db=np.zeros((len(self.Hours),len(self.PartialFG)))

        self.m_air_esp_daf=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.m_air_esp_ad=np.zeros((len(self.Hours),len(self.PartialFG)))
        self.ExcessAir=np.zeros((len(self.Hours),len(self.PartialFG)))
        
        self.EffiGlobal=np.zeros(len(self.PartialFG))
        
        for i in range(len(self.PartialFG)):

            for h in range(len(Hours)):
                [MC_Fuel_daf,Ash_Fuel_daf,MC_Fuel_ad,Ash_Fuel_ad,LHV_Fuel_ad,
                 LHV_Fuel_daf,HHV_Fuel_daf]=fuel_properties(self.MCfin_ad[h,i],"fuel_properties")
                if P_BoilerFactor[h]>0:
                    T_FG_boiler=a_temp*P_BoilerFactor[h]+b_temp
                if P_BoilerFactor[h]>0:
                    [compo_db_fg,compo_wb_fg,CO_d_fg,m_fg,m_air,
                     m_H2O]=FG_Composition(self.P_BoilerFactor[h],self.MCfin_ad[h,i])
                    self.Ash_Boiler_ad[h,i]=Ash_Fuel_ad
                    self.MassFuel_Boiler_ad[h,i]=self.MassFuel_daf[h,i]/((1-self.MCfin_ad[h,i]/100
                                                   -self.Ash_Boiler_ad[h,i]/100))
                    self.LHV_daf[h,i]=LHV_Fuel_daf #J/kg Fuel daf
                    self.LHV_ad[h,i]=LHV_Fuel_ad #J/kg Fuel ad
                    self.m_air_esp_daf[h,i]=m_air #kg air /kg Fuel daf
                    self.m_air_esp_ad[h,i]=m_air/(1+MC_Fuel_daf/100+Ash_Fuel_daf/100) #kg air/ kg Fuel ad
                       
                    FG=MyProperties_Gas_2(compo_db_fg,compo_wb_fg[3],T_FG_boiler,P_ref)
                    
                    self.h_FG_boiler[h,i]=Get_hFG_IN(FG)
                    self.h_FG_dry_boiler[h,i]=FG.cp_db*(T_FG_boiler-273.15)/1000
                    self.h_FG_dry_out[h,i]=self.TFGout[h,i]
                    self.E_FG_dry_ava[h,i]=self.h_FG_dry_boiler[h,i]*self.MassFG_Boiler[h,i]
                    self.E_FG_dry_used[h,i]=self.h_FG_dry_boiler[h,i]*self.MassFG_Dryer[h,i]
                    self.E_ava[h,i]=self.h_FG_boiler[h,i]*self.MassFG_Boiler[h,i]
                    self.E_used[h,i]=self.h_FG_boiler[h,i]*self.MassFG_Dryer[h,i]
                    self.MC_FG_boiler_wb[h,i]=FG.X_H2O
                    self.MC_FG_boiler_db[h,i]=FG.X_H2O/(1-FG.X_H2O)
                    
#                    if T_FG_dryer!=T_FG_boiler:
#                        FG=MyProperties_Gas_2(compo_db_fg,compo_wb_fg[3],T_FG_boiler,P_ref)
#                        FGmix,MassFG21=Mix(self.MassAir[h,i],self.MassFG_Mixer[h,i],Air,FG,T_FG_dryer)
#                        self.MC_FG_dryer_wb[h,i]=FGmix.X_H2O
#                        self.MC_FG_dryer_db[h,i]=FGmix.X_H2O/(1-FGmix.X_H2O)
#                    else:
#                        self.MC_FG_dryer_wb[h,i]=self.MC_FG_boiler_wb[h,i]
```

```
#                        self.MC_FG_dryer_db[h,i]=self.MC_FG_boiler_db[h,i]

                    self.MC_FG_dryer_wb[h,i]=self.MC_FG_boiler_wb[h,i]
                    self.MC_FG_dryer_db[h,i]=self.MC_FG_boiler_db[h,i]


            self.EffiGlobal[i]=Get_EffiGlobal(Pn*self.P_BoilerFactor,self.LHVini_ad,
                                                self.MassFuel_ad[:,i],self.m_air_esp_ad[:,i],
                                                self.Hours)
        for h in range(len(self.Hours)):
            for p in range(len(self.PartialFG)):
                self.ExcessAir[h,p]=Exc_air(self.MCfin_ad[h,p],self.P_BoilerFactor[h])
        
        self.me=me_matrix #kg/s
        self.HeatExc=HeatExc_matrix #kW
        self.Drying_Agent=self.MassFG_Dryer/(self.MassFuel_daf+1*10**-6) #kgFG/kgFueldb
        self.Drying_Agent_FG=self.MassFG_Mixer/(self.MassFuel_daf+1*10**-6)
        self.Heat_Consum=self.HeatExc/(self.me+1*10**-6) #kJ/kgH2O
        self.Heat_Consum_e=self.HeatExc/(self.MassFuel_daf+1*10**-6) #kJ/kgFuel
#Demand Covered and Load Factor
        
        self.LF=LoadFactor(Pn,self.Hours,Pn*self.P_BoilerFactor)
        self.DemCov=DemandCov(Pn*self.P_BoilerFactor,self.Hours,self.Load)
        
#Initialization of economic arrays and matrixs
        self.Years=np.arange(0,Nyears+1,1)

        #NPV[Tau_index,year,FGindex]
        self.NPVHolmberg=np.zeros((Nyears+1,len(self.PartialFG)))
        self.NPVBrammer=np.zeros((Nyears+1,len(self.PartialFG)))
        self.NPVBelt=np.zeros((Nyears+1,len(self.PartialFG)))
        self.NPVSolid=np.zeros((Nyears+1,len(self.PartialFG)))
        
        self.NPVSolid_max=np.zeros((Nyears+1,len(self.PartialFG)))
        self.NPVBelt_max=np.zeros((Nyears+1,len(self.PartialFG)))
        self.NPVSolid_min=np.zeros((Nyears+1,len(self.PartialFG)))
        self.NPVBelt_min=np.zeros((Nyears+1,len(self.PartialFG)))
        
        self.NPV=np.zeros((Nyears+1,len(self.PartialFG)))
        self.NPV_max=np.zeros((Nyears+1,len(self.PartialFG)))
        self.NPV_min=np.zeros((Nyears+1,len(self.PartialFG)))

        self.RTHolmberg=np.zeros((len(self.PartialFG)))
        self.RTBrammer=np.zeros((len(self.PartialFG)))
        self.RTBelt=np.zeros((len(self.PartialFG)))
        self.RTSolid=np.zeros((len(self.PartialFG)))
        self.RT=np.zeros((len(self.PartialFG)))
        
        self.AnnualRev_Belt=np.zeros(len(self.PartialFG))
        self.AnnualRev_Solid=np.zeros(len(self.PartialFG))
        self.AnnuaRev=np.zeros(len(self.PartialFG))
        
        self.Opex_Belt=np.zeros(len(self.PartialFG))
        self.Opex_Solid=np.zeros(len(self.PartialFG))
        self.Opex=np.zeros(len(self.PartialFG))
        
        self.FuelCost=np.zeros(len(self.PartialFG))
        self.ElecCost=np.zeros(len(self.PartialFG))

        self.Capex_Belt=np.zeros(len(self.PartialFG))
        self.Capex_Solid=np.zeros(len(self.PartialFG))
        self.Capex=np.zeros(len(self.PartialFG))
        
        self.FixC_Belt=np.zeros(len(self.PartialFG))
        self.FixC_Solid=np.zeros(len(self.PartialFG))
        self.FixC=np.zeros(len(self.PartialFG))
#Net present value loops
        for i in range(len(self.PartialFG)):
            [NPV_Holmberg,NPV_Brammer,NPV_SolidBand,NPV_BeltBand,
             NPV_SolidBand_max,NPV_BeltBand_max,NPV_SolidBand_min,NPV_BeltBand_min,
```

```
             RT_Holmberg,RT_Brammer,RT_SolidBand,RT_BeltBand,RT,
             AnnualRevenue, FuelCostIni, CapitalCost_index,CapitalCost_array,
             OPEX_array,ElecCost,FuelCost, FixedCost
             ]=NetPresentValue(self.Hours,RhoFG[i],Ac[i],
             self.MassFG_Dryer[:,i],self.MassFuel_daf[:,i], self.MassFuelini_daf)

            self.NPVHolmberg[:,i]=NPV_Holmberg
            self.NPVBrammer[:,i]=NPV_Brammer
            self.NPVSolid[:,i]=NPV_SolidBand
            self.NPVBelt[:,i]=NPV_BeltBand
            self.NPV[:,i]=(self.NPVBelt[:,i]+self.NPVSolid[:,i])/2

            self.NPVSolid_max[:,i]=NPV_SolidBand_max
            self.NPVBelt_max[:,i]=NPV_BeltBand_max
            self.NPV_max[:,i]=(self.NPVBelt_max[:,i]+self.NPVSolid_max[:,i])/2

            self.NPVSolid_min[:,i]=NPV_SolidBand_min
            self.NPVBelt_min[:,i]=NPV_BeltBand_min
            self.NPV_min[:,i]=(self.NPVBelt_min[:,i]+self.NPVSolid_min[:,i])/2

            self.RTHolmberg[i]=RT_Holmberg
            self.RTBrammer[i]=RT_Brammer
            self.RTBelt[i]=RT_BeltBand
            self.RTSolid[i]=RT_SolidBand
            self.RT[i]=RT

            self.AnnualRev_Belt[i]=AnnualRevenue[2]
            self.AnnualRev_Solid[i]=AnnualRevenue[3]

            self.Opex_Belt[i]=OPEX_array[2]
            self.Opex_Solid[i]=OPEX_array[3]

            self.FuelCost[i]=FuelCost
            self.ElecCost[i]=ElecCost
            self.FixC_Belt[i]=FixedCost[2]
            self.FixC_Solid[i]=FixedCost[3]
            self.Capex_Belt[i]=CapitalCost_array[2]
            self.Capex_Solid[i]=CapitalCost_array[3]

        self.FixC=(self.FixC_Belt+self.FixC_Solid)/2
        self.Capex=(self.Capex_Belt+self.Capex_Solid)/2
        self.Opex=(self.Opex_Belt+self.Opex_Solid)/2
        self.AnnualRev=(self.AnnualRev_Belt+self.AnnualRev_Solid)/2

# =============================================================================
# # Acillary functions
# =============================================================================
def Get_EffiGlobal(P,LHVini_ad,MassFuel_ad,m_air_esp_ad,hours):
    Cpmass_air=PropsSI("CPMASS","T",20+273.15,"P",P_ref,"Air")
    MassFuel_ad=MassFuel_ad+1*10**-10 #Avoiding zero in divisor
    hours=hours+1*10**-10
    Effi=np.divide(np.sum(np.multiply(P,hours)),np.sum(np.multiply(np.multiply(
            MassFuel_ad,hours),(LHVini_ad+np.multiply(m_air_esp_ad,Cpmass_air*20)))))
    return Effi

def LoadFactor(Pn,Hours,P_Boiler):
    LF=np.divide(np.sum(P_Boiler*Hours),Pn*365*24)
    return LF

def DemandCov(P_Boiler,Hours,Demand):
    DemCov=np.divide(np.sum(np.multiply(P_Boiler,Hours)),np.sum(np.multiply(Demand,Hours)))
    return DemCov
```