

# **A generic approach to the automated startup and shutdown of processing units using sequential function charts**

Lourens du Plessis

A dissertation submitted for the partial fulfillment  
of the requirements for the degree

**Master of Engineering (Control Engineering)**

in the

Faculty of Engineering, Built Environment and Information Technology  
University of Pretoria  
Pretoria

December 2002

---

## Synopsis

Automated start-up and shutdown procedures increase the profitability and safety of a process, but are difficult to implement due to the complex nature of the concepts that must be incorporated. Generic components used specifically for the implementation of automated startup and shutdown procedures were defined to streamline the implementation process.

The generic components developed are based on Sequential Function Charts and were applied to the startup of a fixed-bed gasification unit, for which a dynamic simulation model was developed. The application showed that the automated startup can be defined by a few generic components and that the flexibility of the startup procedure is increased through the incorporation of a fault accommodation module.

The use of a visual-based definition of sequential processes increases the understanding of the complex scheduling procedures as well as the efficiency of the development of these automated procedures.

In addition, iterative learning was incorporated into the generic definition to optimise controller performance during the non-linear phases of operation.

**KEYWORDS:** fault accommodation, fault detection, fault diagnosis, Grafcet, iterative learning, Lurgi fixed-bed gasifier, processing phases, scheduling, sequential function charts, startup and shutdown

---

## Sinopsis

Ge-outomatiseerde in-en-uitbedryfstellingsprosedures verhoog die winsgewendheid en veiligheid van 'n proses, maar is moeilik om te implementeer as gevolg van die komplekse aard van die konsepte en aksies wat geïntegreer moet word. Generiese komponente wat spesifiek vir die implementering van sodanige prosedures gebruik kan word is geïdentifiseer en ontwikkel om die implementering van die sisteem te vergemaklik.

Die generiese komponente wat ontwikkel is, is gebaseer op Sekwensiële Funksie Diagramme en is toegepas op die inbedryfstelling van 'n gepakte bed vergasser. 'n Dinamiese model is spesifiek hiervoor ontwikkel. Die toepassing van die generiese komponente het bewys dat 'n outomatiese inbedryfstellingsprosedure wel deur 'n aantal generiese komponente saamgestel kan word. Die inbedryfstelling is verder ook meer buigsaam gemaak deur die insluiting van 'n fout-akkommodasie module.

Die gebruik van 'n visueel-gebaseerde definisie van sekwensiële prosesse vergemaklik die werking van komplekse skeduleringsprosedures asook die effektiwiteit geassosieer met die ontwikkeling daarvan.

Iteratiewe leermetodes (Iterative Learning) is geïnkorporeer in die generiese definisie om die werkverrigting van die geassosieerde beheerder te optimeer gedurende die nie-lineêre fases van die inbedryfstellingsprosedure.

**SLEUTELWOORDE:** fout-akkommodasie, fout-diagnose, fout-identifikasie, Grafset, in- en-uitbedryfstelling, iteratiewe ontwikkeling, Lurgi gepakte bed vergasser, produksiefases, sekwensiële funksie diagramme, skedulering.

---

## Acknowledgment

I wish to thank my family for their enduring support and also, my mentor, Professor Philip de Vaal for his guidance and support.

I would further like to thank all my friends and colleagues, at the University of Pretoria, who made my stay at the Process Modelling and Control Group a pleasurable and a memorable one.

Lastly, I would like to acknowledge Sasol for financial support given during my post-graduate studies.

---

# Contents

<b>Synopsis</b>	<b>i</b>
<b>Keywords</b>	<b>i</b>
<b>Acknowledgment</b>	<b>iii</b>
<b>Nomenclature</b>	<b>ix</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Theoretical background</b>	<b>3</b>
2.1 Objectives of an automated startup procedure . . . . .	3
2.2 Integration of automated startup and shutdown procedures as a control procedure	4
2.2.1 Regulatory control . . . . .	5
2.2.2 Supervisory control . . . . .	5
2.3 Representing supervisory control systems . . . . .	6
2.4 Grafcet . . . . .	7
2.4.1 Concept definition . . . . .	7
2.4.2 Grafcet extensions . . . . .	8
2.4.3 Inclusion of startup and shutdown operations in Grafcet . . . . .	9
2.4.4 Inclusion of countermeasure planning in Grafcet . . . . .	11
2.5 Optimisation of controller performance . . . . .	13
2.5.1 Iterative learning methodology . . . . .	14

<b>3</b>	<b>Generic automated Grafcet definition</b>	<b>19</b>
3.1	Visual based approach . . . . .	19
3.2	Computer implementation . . . . .	20
3.3	Component definitions . . . . .	20
3.3.1	Gate . . . . .	20
3.3.2	Step . . . . .	20
3.3.3	Initial step . . . . .	21
3.3.4	Convolution and devolution bars . . . . .	22
3.3.5	Connection posts . . . . .	23
3.3.6	Receptivity . . . . .	25
3.3.7	Actions . . . . .	26
3.4	Macro . . . . .	28
3.5	Heating tank example . . . . .	30
<b>4</b>	<b>Automated gasifier startup and shutdown</b>	<b>34</b>
4.1	Startup and shutdown procedure synthesis . . . . .	34
4.2	Step 1: Definition of phases . . . . .	35
4.2.1	Heating phase . . . . .	35
4.2.2	Air blown phase . . . . .	36
4.2.3	Oxygen blown phase . . . . .	36
4.3	Step 2: Identify controlled and manipulated variables . . . . .	36
4.3.1	Heating phase . . . . .	37
4.3.2	Air blown phase . . . . .	37
4.4	Step 3: Control task development . . . . .	38
4.5	Supervisory controller overview . . . . .	40
4.6	Step 4: Integration of control tasks into the supervisory control system . . . . .	41
4.6.1	Initial phase . . . . .	41
4.6.2	Heating phase . . . . .	41
4.6.3	Air blown phase . . . . .	41
4.7	Step 5: Countermeasure planning system development . . . . .	41
4.7.1	Fault detection . . . . .	45
4.7.2	Fault diagnosis . . . . .	46
4.7.3	Fault accommodation . . . . .	47
4.8	Example: Automated phase scheduling simulation . . . . .	47
4.9	Example: Automated countermeasure planning simulation . . . . .	50
<b>5</b>	<b>Improvement of controller performance</b>	<b>53</b>
5.1	Oxygen phase description . . . . .	53
5.2	Model simulation and profile generation . . . . .	55

<b>6</b>	<b>Conclusions</b>	<b>56</b>
<b>7</b>	<b>Recommendations</b>	<b>58</b>
	<b>Bibliography</b>	<b>61</b>
<b>A</b>	<b>Fixed bed gasifier model</b>	<b>62</b>
A.1	Gasification of coal . . . . .	62
A.2	Gasification unit description . . . . .	62
A.2.1	Coal bunker and lock . . . . .	64
A.2.2	Gasification chamber . . . . .	64
A.2.3	Ash lock and condenser . . . . .	64
A.3	Reaction chemistry . . . . .	64
A.3.1	Ash layer . . . . .	64
A.3.2	Combustion zone . . . . .	65
A.3.3	Gasification zone . . . . .	65
A.3.4	Devolatilization zone . . . . .	66
A.3.5	Pre-heating zone . . . . .	66
A.4	Model description . . . . .	66
A.5	Solid–gas phase reaction component . . . . .	67
A.5.1	Assumptions . . . . .	67
A.5.2	Time dependent gas–solid phase description . . . . .	69
A.6	Gas phase reaction component . . . . .	73
A.6.1	Assumptions . . . . .	73
A.6.2	Steady state equilibrium calculation . . . . .	73
A.7	Drying and devolatilization description . . . . .	75
A.7.1	Assumptions . . . . .	75
A.7.2	Volatiles and drying calculation . . . . .	75
A.8	Gasifier model implementation . . . . .	76
<b>B</b>	<b>Simulation of a gasification unit</b>	<b>79</b>
B.1	Implementation of the computer simulation . . . . .	79
B.1.1	Dynamic bed model . . . . .	79
B.2	Model parameter definition . . . . .	80
B.3	Calculation of modelling data . . . . .	80
B.3.1	Heat of reaction . . . . .	80
B.3.2	Heat capacity . . . . .	82
B.4	Steady state profiles . . . . .	83
B.4.1	Oxygen blown simulation . . . . .	83
B.4.2	Fire–bed movement . . . . .	85

B.4.3	Air blown simulation . . . . .	85
B.5	Dynamic model description . . . . .	85
B.5.1	Initial conditions . . . . .	85
B.5.2	Dynamic profiles . . . . .	86
<b>C</b>	<b>Software description</b>	<b>88</b>
C.1	Gasifier modelling . . . . .	88
C.1.1	m–functions . . . . .	88
C.1.2	s–functions . . . . .	90
C.1.3	Simulink implementation . . . . .	90
C.2	Grafcet . . . . .	93
C.2.1	Overview . . . . .	93
C.2.2	m–function . . . . .	93
C.2.3	s–function . . . . .	93
C.2.4	Algorithms . . . . .	93
<b>D</b>	<b>Pressure model</b>	<b>101</b>
D.1	Equation development . . . . .	101



---

## Nomenclature

$\dot{H}$	Heat of reaction, $kJ/kmol$
$\dot{R}$	Rate of reaction, $kmol/m^3 \cdot min$
$e$	Error output trajectory
$G^*$	Uninvertible part of the linear model
$u$	Input iterative sequence
$u_d$	Nominal input trajectory
$y$	Output iterative sequence
$y_d$	Specified output reference trajectory
$A$	Packed bed area ( $m^2$ )
$C$	Concentration, $kmol/m^3$
$C_p$	Heat capacity, $kJ/kmol \cdot K$
$d_p$	Particle diameter, $m$
$D_{AB}$	Diffusion coefficient, $m^2/s$
$E_A$	Activation energy of the reaction, $J/mole$
$F$	Flow rate for the gas phase, $kmol/min$
$k''$	Reaction temperature coefficient
$k_{ash}$	Ash diffusion rate constant, $m_{gas}^3/m_{surface}^2 \cdot s$
$k_{diff}$	Gas diffusion rate constant, $m_{gas}/s$

$k_{eff}$	Effective reaction rate, $kJ/min \cdot m^3$
$K_{eq}$	Equilibrium constant
$k_{react}$	Reaction rate constant, $m_{gas}/s$
$N$	Number of sampling intervals in the trial
$P$	Partial pressure, $kPa$
$r$	Reaction equation rate, $kJ/min \cdot m^3$
$T$	Temperature (K)
$x_s$	Fraction of fixed carbon in the solid phase
$X_{eq}$	Equilibrium conversion
$Z$	Bed height (m)

### Subscripts

$i$	Gas phase component
$in$	Inlet flow rate
$j$	Solid phase component
$k$	Reaction equation number
$s$	Solid phase
$v$	Devolatilized component
$o$	initial

### Greek

$\Delta$	Difference of the variable
$\epsilon$	Packed bed voidage
$\gamma$	Step length of the gradient based learning algorithm
$\lambda$	Heat of vaporization, $kJ/kmol$
$\nu$	Stoichiometric coefficient

---

## List of Figures

2.1	General control system hierarchy with typical response times(Skogestad, 2000)	5
2.2	Simplified Grafcet representation . . . . .	7
2.3	Grafcet parallel operation . . . . .	8
2.4	Grafcet alternative path operation . . . . .	8
2.5	Higher level definition blocks . . . . .	9
2.6	Inner structure of a macro . . . . .	9
2.7	Partition using connection posts (Årsén, 1994) . . . . .	10
2.8	Grafcet based implementation of the different control phases . . . . .	10
2.9	Control task definition implemented as a Grafcet formalism . . . . .	11
2.10	Phase implementation and countermeasure planning Grafcet implementation . . . . .	12
2.11	A simple sequence pattern ((Årsén, 1996)) . . . . .	12
2.12	Fault detection and countermeasure planning Grafcet implementation . . . . .	13
2.13	Linear process description . . . . .	15
2.14	Block flow diagram of the iterative learning controller . . . . .	16
2.15	Linear model description with disturbances . . . . .	17
2.16	Combined iterative learning and feedback implementation . . . . .	18
3.1	Generic gate algorithm . . . . .	20
3.2	Generic step algorithm . . . . .	21
3.3	Generic step block interface . . . . .	21
3.4	Initial step implementation . . . . .	22
3.5	Convolution block definition . . . . .	22
3.6	Generic convergence bar interface . . . . .	23
3.7	Generic divergence bar interface . . . . .	23
3.8	Macro connection post . . . . .	23
3.9	Discrete fault detection using connection posts . . . . .	24
3.10	Connection post implementation . . . . .	24

3.11	Macro connection post implementation . . . . .	25
3.12	Receptivity implementation . . . . .	25
3.13	Generic receptivity block interface . . . . .	26
3.14	Generic ramp block interface . . . . .	26
3.15	Ramp implementation . . . . .	27
3.16	Generic set block interface . . . . .	27
3.17	Set implementation . . . . .	27
3.18	Generic display icons . . . . .	28
3.19	Display implementation . . . . .	28
3.20	Warning display user input . . . . .	29
3.21	Macro naming convention . . . . .	29
3.22	Macro graphical user interface . . . . .	29
3.23	Heating tank example . . . . .	30
3.24	Grafcet implementation of the heating tank example . . . . .	31
3.25	Height profile during the sequential process . . . . .	32
3.26	Temperature profile during the sequential process . . . . .	32
3.27	Operator output generated during the last step . . . . .	33
4.1	Heating phase control configuration . . . . .	37
4.2	Air blown phase control configuration . . . . .	38
4.3	Supervisory control system of the gasification unit . . . . .	40
4.4	Control task definition of the initial phase . . . . .	42
4.5	Control task definition of the heating phase . . . . .	43
4.6	Control task definition of the air phase . . . . .	44
4.7	Exit condition definition of the air phase . . . . .	45
4.8	Alarm management implementation . . . . .	46
4.9	Air flow accommodation definition . . . . .	47
4.10	Air feed startup profile . . . . .	48
4.11	Steam feed startup profile . . . . .	48
4.12	Top temperature profile . . . . .	49
4.13	Bottom temperature profile . . . . .	49
4.14	Supervisory controller during the normal startup procedure . . . . .	50
4.15	Supervisory control system during fault accommodation . . . . .	51
4.16	Fault accommodated air profile . . . . .	52
4.17	Bottom temperature profile with fault accommodation . . . . .	52
5.1	Pressure ramp operation . . . . .	54
5.2	Grafcet implementation of the iterative learning algorithm . . . . .	54
5.3	Initial and final pressure measurement . . . . .	55
5.4	Controller setpoint for feedback implementation . . . . .	55

A.1	The gasification unit . . . . .	63
A.2	Reaction zones inside the gasification chamber . . . . .	65
A.3	Simplified model of a gasification unit . . . . .	66
A.4	Identified mathematical components of the gasification model . . . . .	67
A.5	Coal feed mechanism with a baffle at the top . . . . .	68
A.6	Radial temperature distribution assumption . . . . .	68
A.7	Time dependent gas-solid phase description . . . . .	69
A.8	Shrinking core methodology . . . . .	71
A.9	Equilibrium reactor description . . . . .	74
A.10	Devolatilization and drying description . . . . .	75
A.11	Simulink implementation of the gasification unit . . . . .	77
B.1	Implementation of the dynamic and steady state equations . . . . .	79
B.2	Simulated description of the gasification unit . . . . .	80
B.3	Steady state temperature profile of the oxygen blown model (1000 minutes) . . . . .	83
B.4	Steady state composition profile of the oxygen blown model (1000 minutes) . . . . .	84
B.5	Steady state temperature gas phase profile (1000 minutes) . . . . .	85
B.6	Steady state temperature profile of the air blown model (2000 minutes) . . . . .	86
B.7	Dynamic temperature profile of the oxygen blown gasifier . . . . .	87
B.8	Dynamic temperature profile of the air blown gasifier . . . . .	87
C.1	Function dependencies . . . . .	89
C.2	Simulink dynamic bed implementation . . . . .	91
C.3	Simulink dry bed sub-system . . . . .	92
C.4	Simulink de-volatilization sub-system . . . . .	93
C.5	Function and simulink block description . . . . .	94
C.6	Step algorithm . . . . .	96
C.7	Convolution bar algorithm . . . . .	97
C.8	Receptivity algorithm . . . . .	97
C.9	Connection post algorithm . . . . .	98
C.10	Macro goto connection post algorithm . . . . .	98
C.11	Macro from connection post algorithm . . . . .	99
C.12	Gate algorithm . . . . .	99
C.13	OrSrcSearch algorithm . . . . .	100
D.1	Dynamic pressure model description . . . . .	101
D.2	Dynamic pressure calculation description . . . . .	102

---

## List of Tables

2.1	Generic control task definition (Yazdi, 1997) . . . . .	11
4.1	Controller and measurement specifications . . . . .	36
4.2	Control task definition for the heating phase . . . . .	38
4.3	Control task definition for the air blown phase . . . . .	39
4.4	Rule based fault diagnosis definition . . . . .	46
A.1	Reaction parameters (Yoon et al., 1978) . . . . .	71
A.2	Inlet coal composition . . . . .	76
B.1	Oxygen blown model parameters . . . . .	81
B.2	$C_p$ values as a function of temperature (Smith et al., 1996:638–639) . . . . .	82
B.3	Oxygen blown model results compared to plant data . . . . .	84
C.1	m–file description . . . . .	88
C.2	s–function descriptions . . . . .	90
C.3	Input and output block definitions . . . . .	90

---

---

# CHAPTER 1

---

## Introduction

The startup of a processing unit is usually non-linear and is characterised by the scheduling of complicated parallel and sequential control tasks, usually with downgraded controller performance and varying process constraints. This is because the plant is operated at different states often far from its original design and operational conditions.

Startup of such processing units is however largely left for manual implementation by the operator. The operator is therefore responsible for the scheduling of the different control tasks to take manual control actions to compensate for the down-graded controller performance due to non-linearities and to diagnose and compensate for faults that might occur during the startup of the process.

The workload of the operator during startup or shutdown of the process is high and can lead to poor plant performance or premature shutdowns due to lack of proper attention to all the variables of the processing unit.

The implementation of an automated startup and shutdown system will increase the safety and operational profitability of the plant because:

- an automated procedure taking care of the scheduling of the different control tasks will reduce the operator workload. The operator will as a consequence have more time to detect, diagnose and take counteractive measures for abnormal events or optimise plant performance.
- the resources used during startup can be monitored by the automated system and their use optimised through the implementation of more effective controller algorithms or non-linear optimization techniques.
- the states and variables will be continuously monitored according to the varying constraints of the different operational phases. Countermeasure planning (taking corrective

actions to faults) and alarm management can be incorporated into the automated structure should any operating constraints be violated during the different phases of operation.

The development of an automated start-up and shutdown procedure is conducted according to certain steps which are listed below:

- Obtain all the information necessary to fully describe the extent of the control problem.
- Determine which tools would be required to develop the automated startup and shutdown strategy.
- Develop and test the proposed automated procedure.
- Implement the automated procedure.

The implementation of such procedures is however difficult due to the varying nature of the concepts that must be integrated into one control system (i.e. discrete and continuous actions, normal operation and abnormal operation). The definition of generic automated components to facilitate the development of these automated procedures will therefore streamline the implementation of these systems. This will cause a reduction in implementation time of these systems and therefore an increase in the profitability via use of the automated system.

The purpose of this study is to:

- i) Determine the position of automated startup and shutdown systems in the control hierarchy. The tools developed specifically for that control system in the hierarchy can then be used, once it is established where automated startup and shutdown “fits in”.
- ii) Define the generic components needed for the implementation of the automated procedures.
- iii) Apply this to the model of a non-linear gasification unit to test the validity of such an approach to the implementation of such systems.

The dissertation will have the following layout in order to answer the questions posed during the study: A theoretical background is given to establish a basis of the different components needed for the implementation of automated startup and shutdown systems, the implementation of these systems in the control hierarchy and methods developed to represent and implement sequential processes. Generic components are then developed specifically for the implementation of the automated systems on any processing unit and are tested on the implementation of a startup procedure of a non-linear model of a gasification unit.

A chapter on the optimization of controller performance during the automated startup of processing units and on how this can be integrated within the structure of the automated startup or shutdown procedure.

Lastly, the main findings of the study are summarised and some recommendations are given.



---

---

# CHAPTER 2

---

## Theoretical background

The objectives of this chapter are to discuss and formulate the implementation of automated startup and shutdown procedures. This will pave the way for the definition and implementation of generic components that can be used for any automated startup and shutdown procedure. The formulation of the automated procedures also incorporates the concepts of fault detection, fault diagnosis and fault accommodation or countermeasure planning.

A technique used for the iterative optimization of sequential systems, called *iterative learning*, is presented for increasing base-layer controller performance during the automated startup and shutdown of processing units.

### 2.1 Objectives of an automated startup procedure

The scheduling of the different tasks associated with startup of chemical processing units are usually done manually. The manual control actions undertaken during a typical startup of a unit process, as found in the chemical industry, can be listed (Bahar et al., 1995) as:

**Binary actions**, that are associated with the discrete events during the startup procedure (e.g. switching pumps “on” or “off”).

**Prepare actions**, that must be conducted before control can be undertaken (i.e. switching a pump on before flow can be controlled).

**Control actions**; these are manual actions by the operator to track a predefined set point profile. Controllers can usually not be used, as the process is operated far from its normal operating regime.

**Corrective actions**, that must be implemented to reject disturbances that cause controlled variables to deviate from their setpoints.

**Interactive actions**, that must be made to variables due to the interaction of different variables on each other. The operator must rely on his knowledge of the plant to understand the different relationships between the different variables.

The automated implementation of the startup of a processing unit must therefore be able to schedule both discrete (binary) and continuous control tasks or at least assist the operator in managing some of these tasks. This could either be by describing the sequence of tasks to be performed or focusing the operator's attention on specific issues, such as the violation of constraints, during the startup procedure. The operator's workload should therefore be reduced (Matsumoto et al., 1993).

The different tasks performed can occur sequentially or in parallel (Bahar et al., 1995). The representation of the logic flow for the control tasks should be clear and unambiguous as it will reduce errors during the implementation and synthesis of the automated procedure.

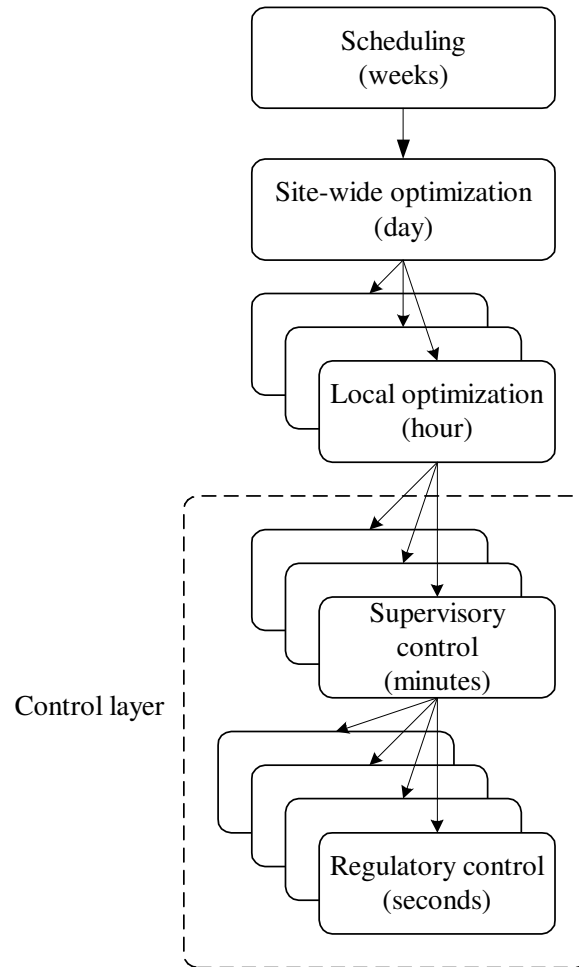
Additional control objectives for startup operations originate from certain optimized regulatory tasks like minimum off-specification products, minimum time and minimum utility consumption (Ganguly & Saraf, 1993; Han & Park, 1999). The startup procedure should therefore be as short as possible (Matsumoto et al., 1993) and executed accurately.

The startup procedure should furthermore allow flexible modification when unexpected errors and abnormalities occur during startup (Matsumoto et al., 1993). The automated system should therefore be able to monitor process conditions and accommodate this in the startup procedure.

## 2.2 Integration of automated startup and shutdown procedures as a control procedure

The control systems used for the control of chemical processes are arranged in a control hierarchy. The controlled process is at the bottom and the incorporation of business goals and management planning through plant-wide scheduling and optimization are at the top. The control hierarchy can be seen in figure 2.1.

The automated startup and shutdown procedure must be implemented by a system situated in the control layer. It is important to define the extent and components of a control system as it will determine the layout of the automated startup and shutdown procedure (i.e. will it be responsible for the control of the plant or will it send setpoints to a lower level). The regulatory and supervisory control layers will be discussed in further detail, in order to clearly define the environment best suited to the implementation of an automated startup or shutdown procedure.



**Figure 2.1:** General control system hierarchy with typical response times(Skogestad, 2000)

### 2.2.1 Regulatory control

Regulatory control systems include base layer control and advanced control systems both with the objective to reduce the variance of the controlled variables. The base layer control systems output directly to the actuators of the final control elements and usually have one actuator (controlled variable) associated with one measurement (single input single output system).

The advanced control systems have the setpoints of the base layer control as the generated controller outputs and are characterised by single control algorithms that use multiple inputs to generate multiple outputs or set points for the base layer control systems (Marlin, 2000).

### 2.2.2 Supervisory control

Supervisory control systems are situated above the regulatory control systems and are used for control applications such as set point control, monitoring, fault detection, diagnosis, scheduling, planning and production optimization (Årsén, 1994). Reactive scheduling is an important part of the supervisory controller and is needed when there is a change in a planned operation (Rengasamy, 1995). The original schedule must therefore be modified to accommodate these

changes (Rengasamy, 1995).

The supervisory control system must therefore:

- i) Schedule the different phases of operation for continuous (i.e. startup, production, shut-down) or batch process operation.
- ii) Deal with faults occurring during the different phases of process operation. The tasks associated with this are (Isermann, 1997):
  - fault detection that determines the faults present in a system and the time of detection,
  - fault diagnosis that determines the kind, size and location of the fault in the process, and
  - supervision to take appropriate actions (fault accommodation or countermeasure planning) to maintain the operation in case of faults.

Startup and shutdown procedures are characterised by the monitoring and scheduling of the different sequential and parallel control configurations. The automated implementation of these procedures (automated startup and shutdown) is therefore the responsibility of the supervisory control system (Rengasamy, 1995).

## 2.3 Representing supervisory control systems

The implementation of supervisory control systems necessitate the use of a documentation standard that will unambiguously represent the sequential and parallel control tasks performed during the different phases of plant operation. This will reduce the errors occurring during the implementation of the phased procedure synthesis and implementation.

The traditional documentation standards used for the representation of control systems are *Process Flow Diagrams* (PFD's) and *Piping and Instrumentation Diagrams* (P&ID's). These documentation standards were however developed to represent static (time independent) controller and plant configurations. Sequential processes cannot be represented by these documentation standards as the configuration of the control architecture and plant change with time during the different phases of startup or shutdown.

Documentation standards defined to represent sequential control schemes, such as typically executed by Programmable Logic Controllers (PLC's) handling interlock systems, are Sequential Function Charts (SFC's) and will complement the other documentation standards. A SFC formalism, called Grafcet, was developed specifically for (Årsén, 1994):

- supervisory level sequence control.
- modelling and simulation of discrete event processes.

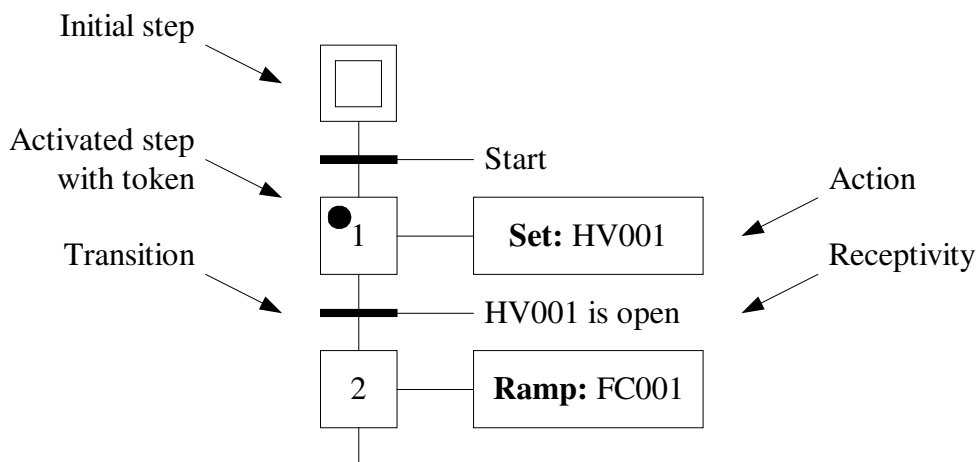
- monitoring and diagnosis of sequential processes.
- representation of operating procedures.
- partitioning of large rule bases.
- control and representation of general sequential reasoning procedures.

Grafcet has been applied to the development of supervisory controllers, the scheduling of batch processes and the execution of automated startup and shut down procedures for continuous processes (Johnsson & Årsén, 1998; Årsén, 1994; Yazdi, 1997).

## 2.4 Grafcet

### 2.4.1 Concept definition

Grafcet consists of two types of nodes: steps and transitions (figure 2.2). A *step* can either be active or inactive and a token resides inside the step when it is active. A *step* represents a state, phase or mode and has associated actions that are executed when the *step* is active. (Årsén, 1994)



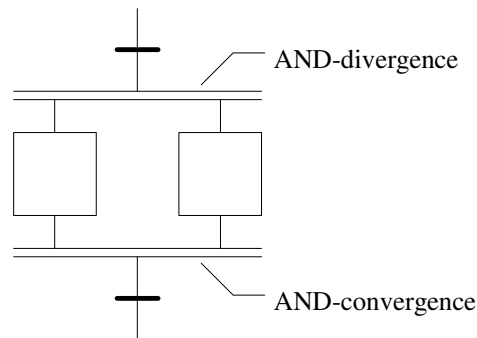
**Figure 2.2:** Simplified Grafcet representation

A *transition* is the connection between two steps. *Receptivity* is associated with each *transition*. The *receptivity* can either be a Boolean condition or an event, or an event together with a condition (Årsén, 1994). *Receptivity* is tested as soon as the preceding *step* of the *transition* is activated. When the *receptivity* is true, the preceding *step* is deactivated and the next *step* after the transition is activated.

Normally, the steps are aligned vertically and no arrow is used when the transition is downwards (Mandano et al., 1996). If the transition takes place upwards an arrow is included.

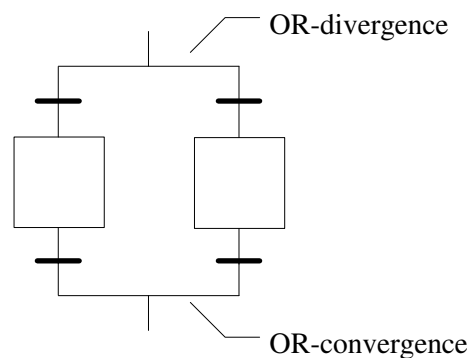
Parallel and alternative path operation are also specified by the Grafcet formalism.

**Parallel paths** are represented by two parallel bars that follow a transition, as can be seen in figure 2.3 (Årsén, 1994). The parallel path principle can mathematically be represented by the AND operation and the two parallel bars are therefore sometimes called the *and-divergence* and *and-convergence* bar (Johnsson & Årsén, 1998). When the *transition* is activated, all the *steps* below the *divergence bar* will be activated (Johnsson & Öhman, 1995). The *convergence bar* will be activated when all the above *steps* are activated.



**Figure 2.3:** Grafcet parallel operation

**Alternative paths** are represented by two or more transitions that follow a transition (figure 2.4). It can be seen that all transitions will be true if the above step is true and the condition is satisfied. This can be represented by the OR function hence the name *OR-divergence bar* and *OR-convergence bar* for the split and merge of the paths (Johnsson & Årsén, 1998).



**Figure 2.4:** Grafcet alternative path operation

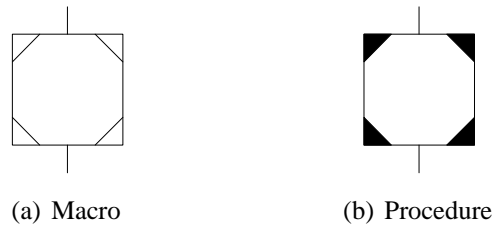
## 2.4.2 Grafcet extensions

Higher level elements were defined that will reduce the complexity of the sequential representation and are *macro's*, *procedures* and *connection posts*.

**Macro steps** are used to represent steps with an internal structure (Johnsson & Årsén, 1998).

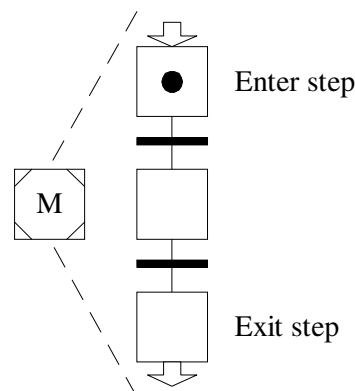
This is used to reduce the complicated control sequences into larger, and more understandable, sub-groups.

**Procedures** are defined for sequences that are executed more than once in the sequential representation (figure 2.5(b)).



**Figure 2.5:** Higher level definition blocks

The inner structure of the *macro* or *procedure* contains *steps* and *transitions* with a special *enter-step* and *exit-step* used to indicate the first and last steps of the *macro* or *procedure* (Årsén, 1994). The internal structure of a macro is shown in figure 2.6.



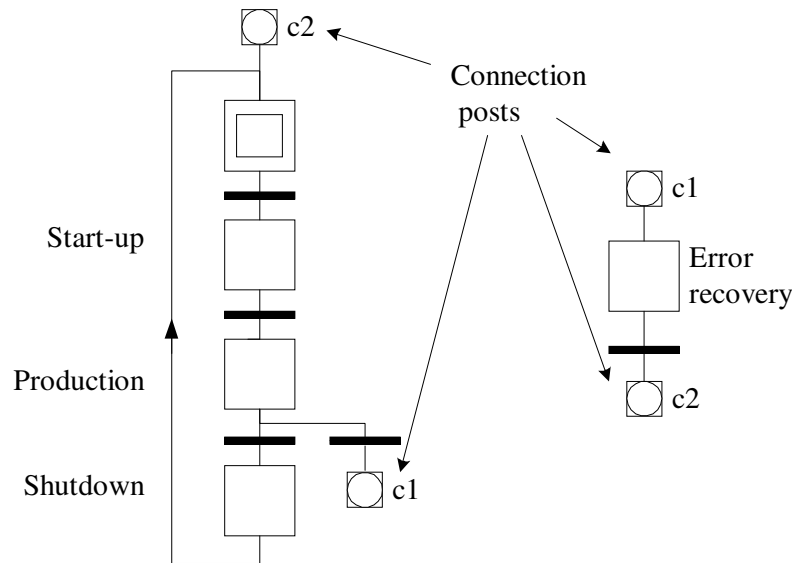
**Figure 2.6:** Inner structure of a macro

**Connection posts** were defined to represent links without showing them graphically on the workspace (Årsén, 1994). This reduces the complexity of the graphical representation by removing unnecessary lines between different *macro*'s or *blocks*. Figure 2.7 shows how the connection posts are used to partition the representation into two parts, one for normal operation and the other for error recovery.

### 2.4.3 Inclusion of startup and shutdown operations in Grafacet

#### Phase implementation

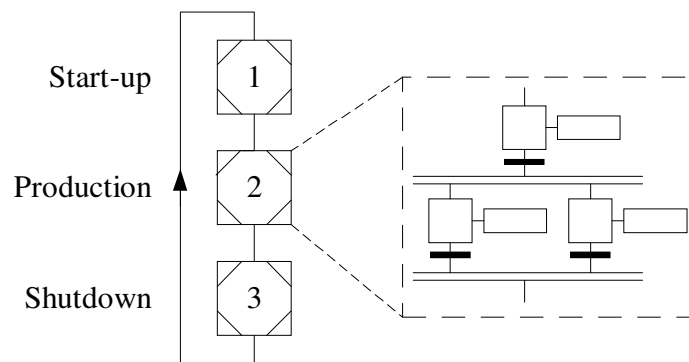
The operation of a continuous plant can be divided into different phases that includes the startup and shutdown of the plant. Phases are stages in the operating procedure where the plant and



**Figure 2.7:** Partition using connection posts (Årsén, 1994)

control configuration stay the same (Yazdi, 1997). The phase may contain several control tasks (Yazdi, 1997).

The Grafcet implementation of the sequential control part of the supervisory control system would therefore be structured into macro's defining the different phases (figure 2.8). The sequential steps defining the different control tasks of each phase are then situated inside the different macro's.



**Figure 2.8:** Grafcet based implementation of the different control phases

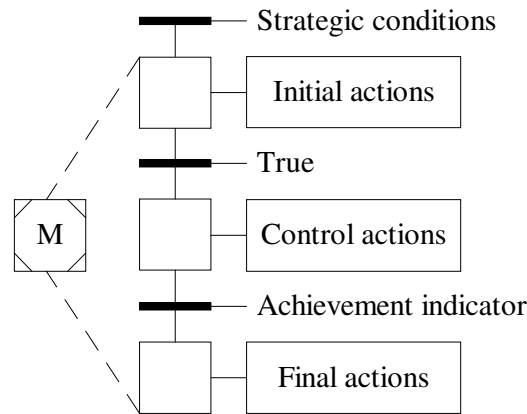
### Task implementation

The control task can be divided into different generic functions. Defining each of these functions will define the control task. The different functions of the generic control task are discussed in table 2.1 (Yazdi, 1997) and the Grafcet implementation can be seen in figure 2.9.



**Table 2.1:** Generic control task definition (Yazdi, 1997)

Function definition	Description
Goal	Defines the task objective.
Strategic conditions	Condition set that has to be valid at task initiation.
Execution conditions	A set of conditions that has to be valid during the task execution.
Initial actions	Actions that have to be performed before the control task can take place.
Control actions	Manipulating actuators and measuring process conditions to achieve a certain goal.
Achievement indicator	Defines the degree of goal achievement for the control task.
Final actions	Set of final actions that is activated as soon as the control task objective is reached.


**Figure 2.9:** Control task definition implemented as a Grafset formalism

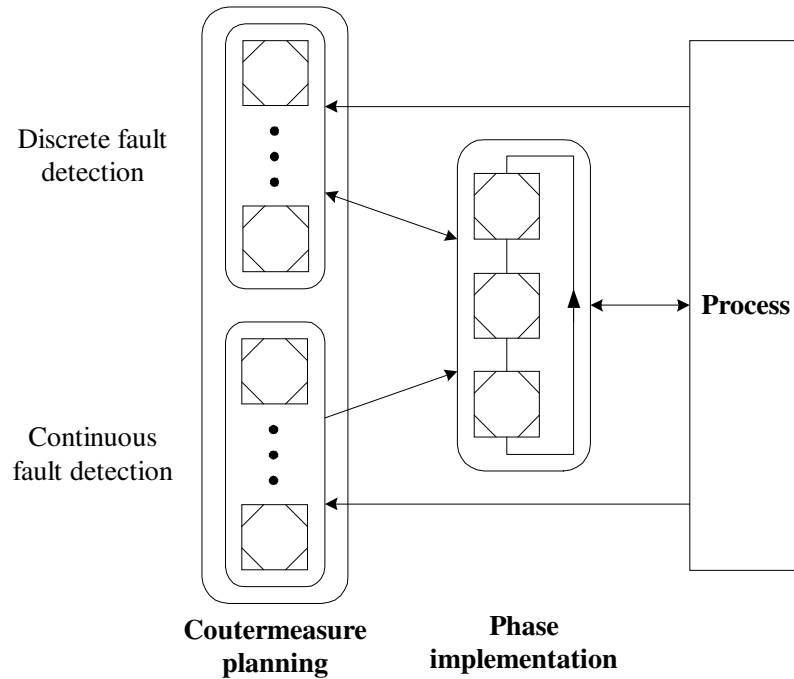
## 2.4.4 Inclusion of countermeasure planning in Grafset

### Supervisory structure

The supervisory control system is divided into two different sub-systems (figure 2.10) in order to include counter measure planning (Yazdi, 1997). The first system, denoted phase implementation, is responsible for the scheduling and optimization of the different phases of the startup or shutdown, while the second system (countermeasure planning) will be used to detect faults and implement countermeasure procedures.

The countermeasure planning system contains two different types of fault detection methods to monitor the different phases during the automated procedure implementation. The detection can either be discrete or continuous (Yazdi, 1997) and depends on the different states of the monitored variable during the different phases of the startup and shutdown procedure.

**Discrete detection** is activated at different (discrete) instances in the startup procedure. The supervised step must be activated and the pre-defined receptivity true before any counter



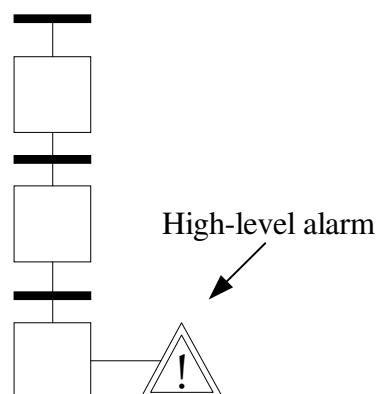
**Figure 2.10:** Phase implementation and countermeasure planning Grafcet implementation

measures will be taken.

**Continuous detection** is active during all the phases and counter measure will be taken as soon as the receptivity denoting a certain fault is true.

### Fault detection

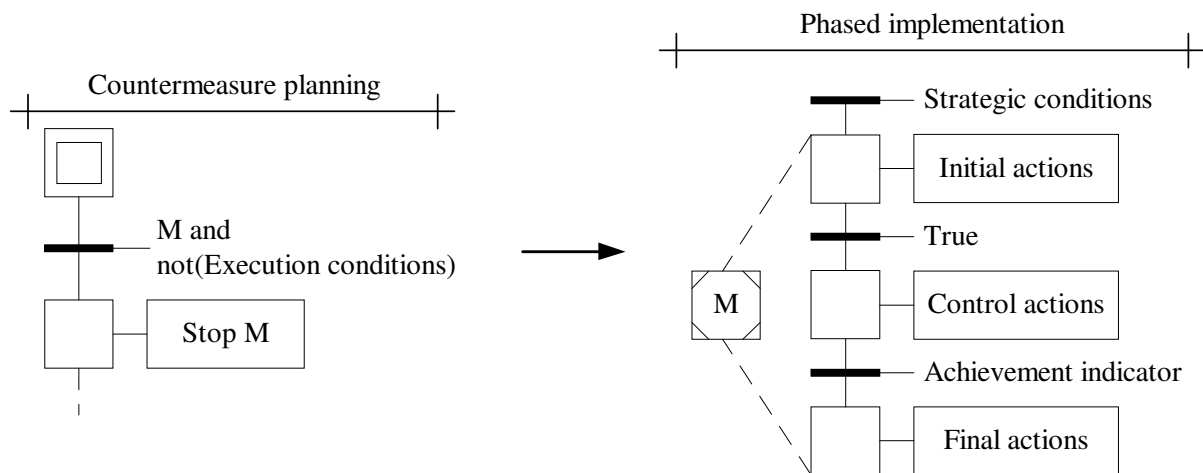
The Grafcet formalism can be used to represent the alarm patterns used for the detection of faults. The occurrence of an alarm can be associated with a transition event (Årsén, 1996) and the manifestation of an alarm can be represented using sequential *steps* and *transitions*. The representation of a simple sequence can be seen in figure 2.11 and shows how a high level alarm will be generated when the above *transitions* are satisfied.



**Figure 2.11:** A simple sequence pattern ((Årsén, 1996))

### Countermeasure system scheduling

The operations within the phase implementation system will be suspended by the countermeasure planning system as soon as a fault is detected (Yazdi, 1997). The countermeasure system will then implement procedures to compensate for the fault. The implementation of the two different supervisory sub-systems is shown in figure 2.12 and shows how the countermeasure planning sub-systems suspend the phase implementation sub-system as soon as the execution conditions are violated (not true) and the macro (M) is active.



**Figure 2.12:** Fault detection and countermeasure planning Grafcet implementation

## 2.5 Optimisation of controller performance

Startup of chemical processes which involve complex heat- and mass-transfer operations result in challenging control problems (Ganguly & Saraf, 1993). This is because the state variables are in many cases subjected to drastic changes during the startup operation which makes it difficult to apply the linear controllers, used for continuous operation, over the operating range of the startup (Han & Park, 1999). The parameters of the startup procedure that can be optimised are to (Ganguly & Saraf, 1993; Han & Park, 1999; Matsumoto et al., 1993; Shaikh & Lee, 1995):

- i) reduce startup time,
- ii) minimize emissions and waste generation,
- iii) maximize operation lifetime and
- iv) reduce resource usage.

The optimisation of these parameters can be accomplished in different ways and ranges from altering the startup operation off-line (Shaikh & Lee, 1995) or on-line (Pradubsripetch et al., 1996) to using model based non-linear optimisation techniques (Sørensen & Skogestad,

1996). These techniques are however very process specific and do not include the worsening base-layer controller performance due to the non-linear nature of the process during the startup or shutdown procedure.

The operators' work load can be reduced by optimising the base-layer controller performance (i.e. servo tracking). This will take care of the control and interactive actions that must be taken during the automated procedure. The automated startup can therefore be operated closer to the constraints to reduce startup time, resource usage, waste generation and emissions.

A more elaborate process specific optimisation technique can then be implemented on top of the current structure once the controller performance is sufficient (i.e. good set-point tracking and disturbance rejection).

An optimization algorithm used to improve the controller performance during the automated startup and shutdown of continuous processes must therefore have the following characteristics.

- The process must be able to adapt to varying conditions that are not necessarily expected. The optimiser should therefore learn from the previous implementation of the startup and shutdown procedures.
- The normal base layer controllers should be used during the startup procedure to ensure the safe operation of the plant.
- The optimisation algorithm should not be too computationally intensive making its use inefficient as it slows the startup and shutdown procedures.

The operation of continuous plants during the startup and shutdown phases of the process are similar to batch process control in that both control procedures are characterised by controlled variables varying over a wide range of operating conditions over a finite time interval.

### 2.5.1 Iterative learning methodology

An algorithm originally developed for application in the field of robotics (Amann et al., 1996) but has subsequently found application in batch process control (Lee et al., 1999, 1996) is iterative learning. Iterative learning uses a non-linear adaptive structure that utilises previous experience to optimise the controller performance over a finite time or trial run.

The advantages of iterative learning are:

- The algorithm is adaptive and can change to accommodate varying startup procedures.
- The optimisation technique monitors current controller performance and will accommodate it in the set points generated. There is therefore no need to change any of the base-layer controller parameters.

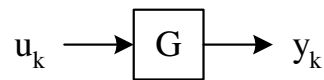
- The optimisation computation occurs off-line. The startup of the process will not be slowed due to intensive computation techniques as is the case for other non-linear optimisation methods.
- No prior model of the process is needed. The generation of non-linear models that will accurately model the startup of a chemical process is time-consuming and usually requires large computational resources by the model. It is not needed for the implementation of iterative learning.

### Ideal iterative controller design

Suppose the outputs ( $y$ ) of a linear open loop stable process (figure 2.13) as a function of its inputs ( $u$ ) can be described by:

$$y = Gu \quad (2.1)$$

with  $G$  the transfer function of any process.



**Figure 2.13:** Linear process description

The output sequence (output vector),  $\mathbf{y}_k$ , generated for a trial run  $k$  of the process over a finite time ( $N$ ) can be computed given the input sequence ( $\mathbf{u}_k$ ) with (Mezghani et al., 2001):

$$\mathbf{y}_k = \mathbf{G}\mathbf{u}_k \quad (2.2)$$

where:

$$\mathbf{y}_k^T \equiv [y_k(1), y_k(2), \dots, y_k(N)] \quad (2.3)$$

$$\mathbf{u}_k^T \equiv [u_k(0), u_k(1), \dots, u_k(N-1)] \quad (2.4)$$

and  $\mathbf{G}$  is a lower triangular matrix containing the impulse response coefficients of the linear process with the following structure:

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & 0 & \dots & 0 \\ g_1 & g_0 & 0 & \dots & 0 \\ g_2 & g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & g_{N-3} & \dots & g_0 \end{bmatrix}$$

Let  $\mathbf{y}_d$  and  $\mathbf{u}_d$  represent the specified output reference trajectory and the corresponding

nominal input trajectory. The output tracking error ( $e$ ) can then be defined by:

$$\mathbf{e} \equiv \mathbf{y}_d - \mathbf{y} \quad (2.5)$$

$$= \mathbf{G}(\mathbf{u}_d - \mathbf{u}) \quad (2.6)$$

This sequence can now be written for consecutive trial runs by defining the subscript  $k + 1$  as the current trial run:

$$\mathbf{e}_{k+1} = \mathbf{G}(\mathbf{u}_d - \mathbf{u}_{k+1}) \quad (2.7)$$

$$\mathbf{e}_k = \mathbf{G}(\mathbf{u}_d - \mathbf{u}_k) \quad (2.8)$$

$$\therefore \mathbf{e}_{k+1} - \mathbf{e}_k = -\mathbf{G}(\mathbf{u}_{k+1} - \mathbf{u}_k) \quad (2.9)$$

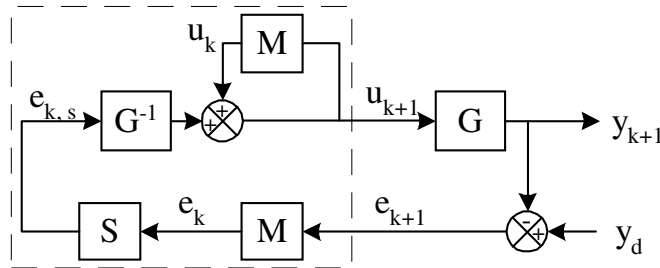
The current tracking error ( $e_{k+1}$ ) is minimised (i.e.  $e_{k+1} = 0$ ) to derive the optimal input profile ( $u_{k+1}$ ) that will result in no tracking error for the current trial run (Amann et al., 1996).

$$\mathbf{e}_k = \mathbf{G}(\mathbf{u}_{k+1} - \mathbf{u}_k) \quad (2.10)$$

$$\therefore \mathbf{G}^{-1}\mathbf{e}_k = \mathbf{u}_{k+1} - \mathbf{u}_k \quad (2.11)$$

$$\therefore \mathbf{u}_{k+1} = \mathbf{G}^{-1}\mathbf{e}_k + \mathbf{u}_k \quad (2.12)$$

The implementation of the iterative learning controller can be seen in figure 2.14. The figure shows that the current tracking error ( $e_{k+1}$ ) and output profile ( $u_{k+1}$ ) are stored ( $M$ ) for use in the calculation of the output profiles for the next trial run (i.e.  $u_{k+2}$ ).



**Figure 2.14:** Block flow diagram of the iterative learning controller

The previous tracking error ( $e_k$ ) is sent through a shift operator ( $S$ ) before it is multiplied with the inverse model of the plant. The shift operator determines if the output profile is generated for a feed-forward or feedback control implementation:

$$\mathbf{e}_{k,s} = \mathbf{S}(\mathbf{e}_k) \quad (2.13)$$

**Feed-forward implementation:** The manipulated output will be implemented one time increment in advance of the tracking error ( $e_k$ ). The tracking error index is accordingly moved

back (backward shift operation) to give:

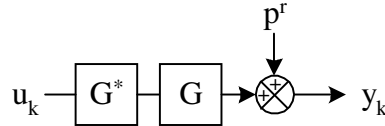
$$\mathbf{e}_{k,s} = [e_k(2), \dots, e_k(N), e_k(N)] \quad (2.14)$$

**Feedback implementation:** The tracking error index is moved forward for a feedback implementation (forward shift operation):

$$\mathbf{e}_{k,s} = [e_k(1), e_k(1), \dots, e_k(N - 1)] \quad (2.15)$$

### Modelling errors and disturbances

The process model must be expanded to include modelling errors and disturbances for the current trial (figure 2.15). The expanded model can now be used for the development of a controller algorithm that will be capable of handling the model errors and disturbances.



**Figure 2.15:** Linear model description with disturbances

The expanded model can be represented by:

$$\mathbf{y}_{k+1} = \mathbf{G}^* \mathbf{G} \mathbf{u}_{k+1} + \mathbf{p}_{k+1}^r \quad (2.16)$$

where  $\mathbf{G}^*$  is the uninvertible part and  $\mathbf{G}$  the invertible part of the transfer function. The unmeasured disturbances are represented by  $\mathbf{p}_{k+1}^r$ . The equation to be optimised can be derived in the same manner as equation 2.9 to give:

$$\mathbf{e}_{k+1} - \mathbf{e}_k = -\mathbf{G}^* \mathbf{G} (\mathbf{u}_{k+1} - \mathbf{u}_k) + \mathbf{p}^r \quad (2.17)$$

where  $\mathbf{p}^r$  represents the difference in the disturbance for the two trial runs.

The ideal controller can now be used to determine the current input needed (from equation 2.12).

$$\mathbf{e}_{k+1} - \mathbf{e}_k = -\mathbf{G}^* \mathbf{G} (\mathbf{G}^{-1} \mathbf{e}_k + \mathbf{u}_k - \mathbf{u}_k) + \mathbf{p}^r \quad (2.18)$$

$$\therefore \mathbf{e}_{k+1} = (1 - \mathbf{G}^*) \mathbf{e}_k + \mathbf{p}^r \quad (2.19)$$

It can be seen from equation 2.19 that an offset ( $e_{k+1} \neq 0$ ) will be generated for the current trial. An on-line feedback algorithm must therefore be inserted to compensate for the disturbances,  $\mathbf{p}^r$ , as well as modelling errors,  $(1 - \mathbf{G}^*) \mathbf{e}_k$ , that occur in the current trial.

### Gradient based iterative learning algorithm

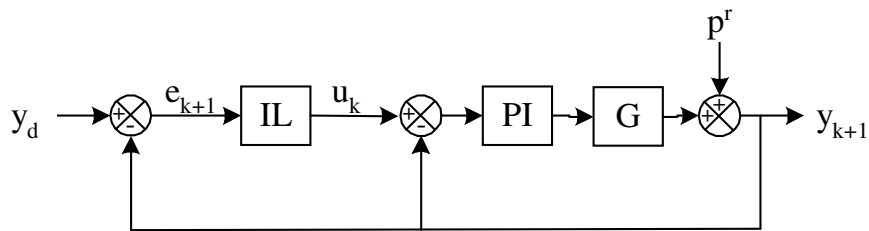
There are many possible configurations to solve the iterative optimization problem, which include the combination of iterative learning with model predictive control (Lee et al., 1999) and Smith (time delay) compensators (Xu et al., 2001). Most of the development has however been toward the implementation of the iterative learning technique on batch processes.

The iterative learning optimization algorithm used here is known as the gradient based approach and can be formulated as (Amann et al., 1996):

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \gamma_{k+1} \mathbf{G}^{-1} \mathbf{e}_k \quad (2.20)$$

where  $\gamma$  is a step length that can be chosen at each time step.

The on-line feedback algorithm of the base-layer controllers of the process can be used to remove the disturbances and modelling errors. The base-layer control performance during the automated startup and shutdown procedure is thus optimised with the use of iterative learning (figure 2.16)



**Figure 2.16:** Combined iterative learning and feedback implementation

with  $IL$  representing the iterative learning algorithm,  $PI$  the feedback PI-control algorithm and  $G$  the plant model.



---

---

## CHAPTER 3

---

### Generic automated Grafcet definition

The visual based procedure for the implementation of sequential processes, called Grafcet, was defined in the theoretical background. The different components defined in the Grafcet formalism will be developed specifically for implementation in SIMULINK with the purpose to develop a set of generic icons that can be used for the development and implementation of any startup or shutdown procedure.

#### 3.1 Visual based approach

The advantages of using a visual presentation to develop and implement the user interface of a supervisory control system, instead of a text-based approach are:

- The representation of the supervisory control system during its synthesis, development and operation stays the same. Implementation errors are greatly reduced as the presentation of the control system stays the same.
- There is no need to develop an interface translating text-based code to a more user friendly format. The current format is already graphical using tokens and colours to denote different events or states of the supervisory control system.
- Parallel sequences are shown clearly and unambiguously. The current state of the supervisory control system can be deduced faster and faults that occur can be detected more easily.
- The development of the supervisory control system in the visual based environment consists of easy click-and-drag operations using standard block icons. Less time can be spent on the development of the supervisory control system, and more on the synthesis of the control system, increasing the efficiency of the project and decreasing errors due to unexpected events.

## 3.2 Computer implementation

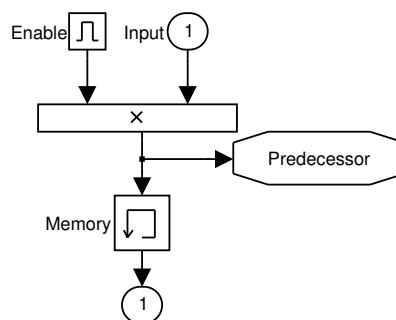
A visual-based programming language, SIMULINK that is part of the MATLAB suite of programming tools, was used to define the generic Grafcet components used for the development of the automated startup or shutdown procedures. SIMULINK is an advanced modelling and simulation feature that allows the user to build block diagrams of dynamic systems (Kheir et al., 1996). The programming architecture can be represented as a two dimensional block flow diagram that uses lines to represent the flow of information connecting the different dynamic or static components on a two-dimensional workspace.

## 3.3 Component definitions

The different generic components that are needed for the development of an automated startup or shutdown procedure will now be discussed. The flow diagrams and a short description of the custom functions (typeset in bold) developed can be seen in the appendix.

### 3.3.1 Gate

The *gate* (or transition) definition can be seen in figure 3.1. The *enabled* input (from the receptivity block) and *input* (from the step block) are multiplied. The output of the gate will be equal to one if both inputs are one. The **predecessor** function is used to deactivate the preceding *step* block when it receives an input of one.



**Figure 3.1:** Generic gate algorithm

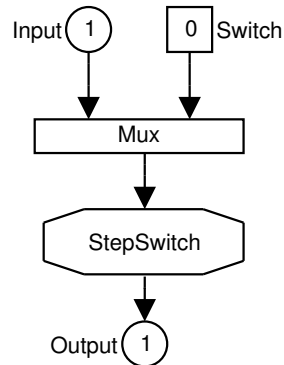
A memory block is included to ensure that the preceding *step* is deactivated before the next *step* is activated.

### 3.3.2 Step

The *step* was developed according to the specifications given in the Grafcet formalism, which is to activate the associated *action*, which is accomplished by setting the *step* output to one, when the *step* is activated; until it is deactivated by the *gate*. The token that specifies the active

*step* was replaced with a change in the background colour. The colour definition of the *step* is “green” when activated and “white” when deactivated.

The *step* implementation can be seen in figure 3.2 and shows that the **StepSwitch** function is used to control the output of the *step* according to the *input* and the *switch*.

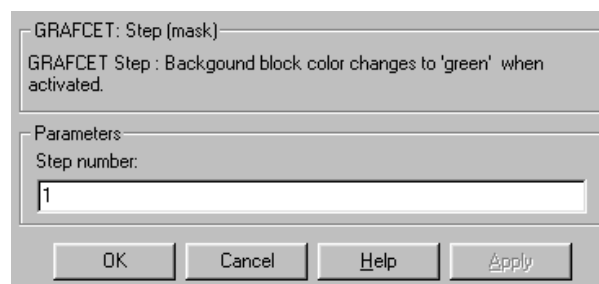


**Figure 3.2:** Generic step algorithm

The **StepSwitch** function is used to:

- i) set the *output* to one when the *input* is equal to one and the *switch* is equal to zero,
- ii) set the *step* background colour to “green” when the *output* is equal to one, and
- iii) reset the *step* by setting the *output* to zero and the background colour to “white” when the *switch* is equal to one. The *switch* is then also set back to zero.

The step number can be defined by the user, by double-clicking on the *step* icon. This will activate the *graphical user interface* (GUI) displayed in figure 3.3. The step number can now be inserted in the text box and the *step* will be updated as soon as the “Apply” or “OK” button is pressed.

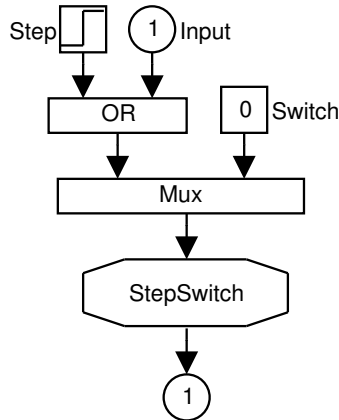


**Figure 3.3:** Generic step block interface

### 3.3.3 Initial step

The *initial step* is used to initialise a sequential operation or show the starting point of the sequential operation for a batch process (cyclic operation). The implementation of the *initial*

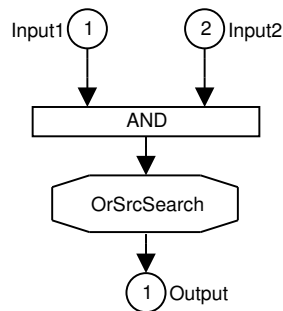
step can be seen in figure 3.4, and shows that the *initial step* can either be activated by the *input* used when the sequential process is cyclic or a *step block* when the simulation is initialised.



**Figure 3.4:** Initial step implementation

### 3.3.4 Convolution and devolution bars

The *convolution bar* (figure 3.5) performs either the AND or the OR operation on the *inputs* as specified by the user.

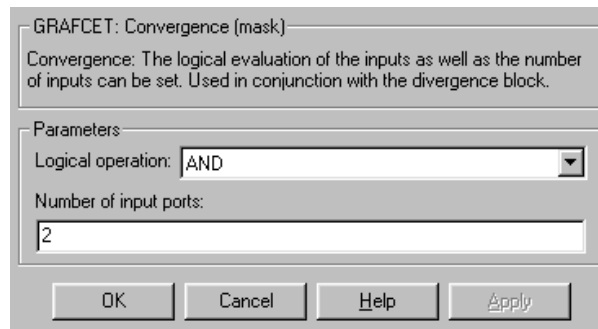


**Figure 3.5:** Convolution block definition

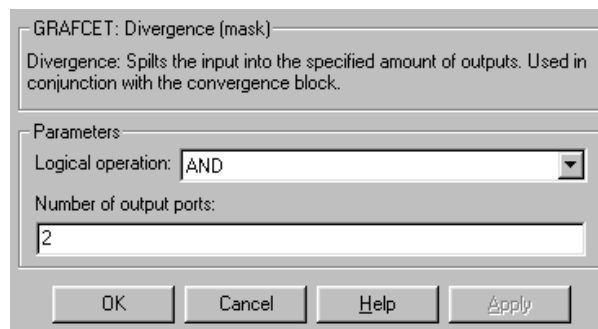
The **OrSrcSearch** function is used to determine all the *steps* connected to the *convolution bar* and deactivates the *steps* if the logical operation is satisfied (i.e. the *output* is one). This is of great importance to the OR operation where it is not necessary to complete all the tasks in order to satisfy the OR operation. The unfinished *steps* must therefore be deactivated.

The *devolution bar* stays the same for the AND or the OR operation as it only splits the input port signal into the specified amount of *outputs*.

The logical operation type can be chosen by the user using a drop-down list on the GUI (figures 3.7 and 3.6). The appearance will change according to the logical operation specified, as either “black” for the OR operation or two parallel bars for the AND operation. This is in accordance with the Grafcet formalism. The number of *inputs* or *outputs* must also be specified by the user.



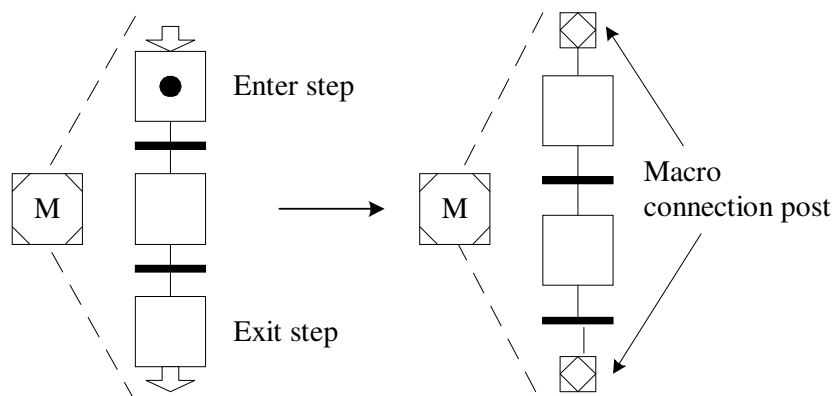
**Figure 3.6:** Generic convergence bar interface



**Figure 3.7:** Generic divergence bar interface

### 3.3.5 Connection posts

Two types of connection posts were defined. The normal *connection post* and a *macro connection post* that replaces the enter-step and exit-step methodology defined in the theoretical background. Figure 3.8 shows that the use of the *macro connection post* represents the Grafcet

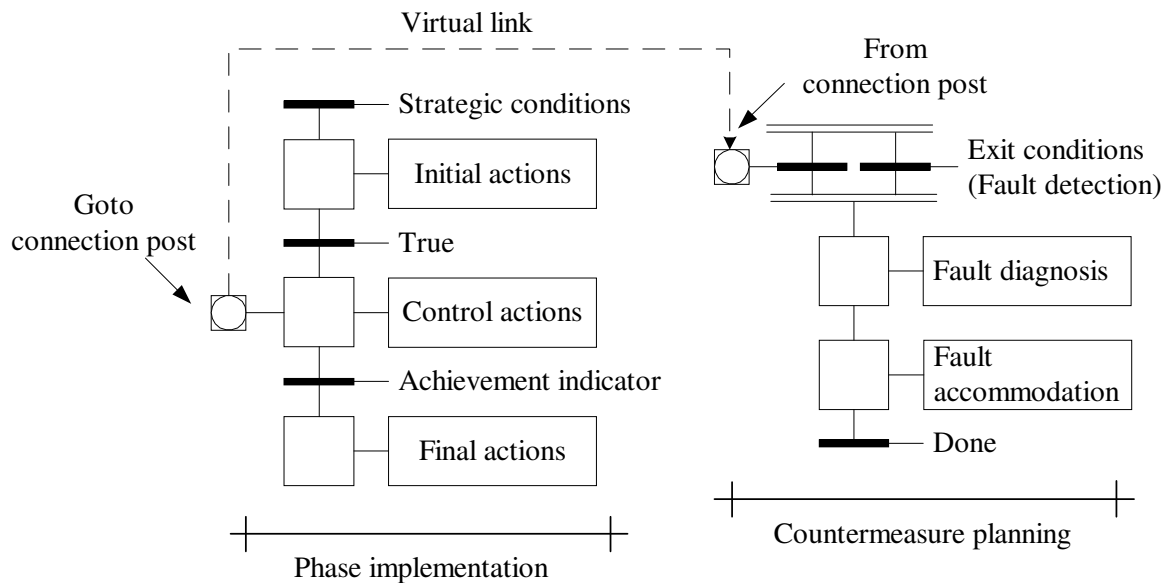


**Figure 3.8:** Macro connection post

formalism better than the enter-step and exit-step methodology. This is because a *transition* or *gate* does not follow the exit-step as is specifically defined by Grafcet.

### Connection post

Two *connection posts* are used to describe a virtual connection. Figure 3.9 shows how the *connection posts* can be used to enable the fault detection of the countermeasure planning system defined in the theoretical background. The fault will be detected should one of the exit conditions be satisfied and the *connection post* is activated (i.e. equal to a value of one). The fault will then be diagnosed and accommodated in the subsequent steps.



**Figure 3.9:** Discrete fault detection using connection posts

The implementation of the *goto* and *from* connection posts can be seen in figures 3.10(a) and 3.10(b). The **ConstGoto** function is used to send the *input* of the *goto connection post* to the *From goto* block of the *from connection post*.



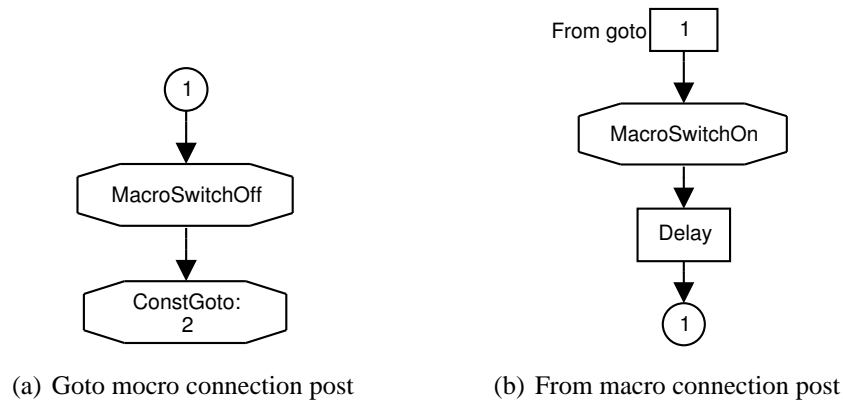
**Figure 3.10:** Connection post implementation

### Macro connection post

The implementation of the *macro connection posts* is shown in figure 3.11(a) and 3.11(b). The **MacroSwitchOff** function is included in the *macro goto connection post* and is used to set the background colour of the *macro* to “white” when the output is equal to one. The corresponding

**MacroSwitchOn** function is included in the *macro from connection post* definition and is used to:

- i) set the *macro* background colour to “green” when the *input* is equal to one,
- ii) reset all the *steps* in the *macro* that is still active. Active *steps* could remain if the *macro* operation was suspended by the countermeasure planning system.

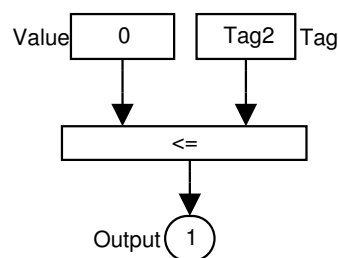


**Figure 3.11:** Macro connection post implementation

The delay block is used to ensure that all the blocks are deactivated before the first *step* in the *macro* is initialised.

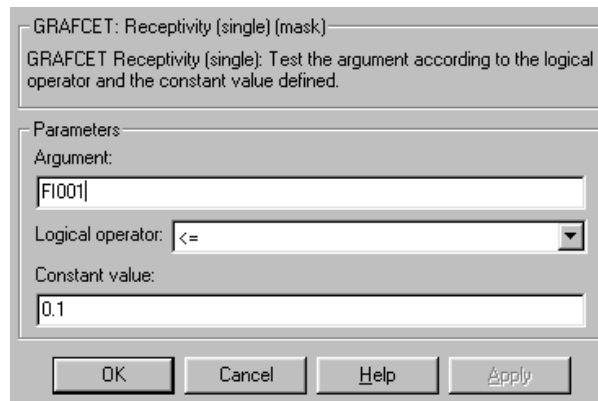
### 3.3.6 Receptivity

The *receptivity* icon is used to test the process measurement (*value*) and constant value (*tag*) with the condition specified by the user and can be seen in figure 3.12. The *output* of the *receptivity* will be assigned the value one when the condition is satisfied.



**Figure 3.12:** Receptivity implementation

The interface used to obtain the relevant information can be seen in figure 3.13. The tag number of the process variable (argument), the constant value and type of logical operation must be defined by the user.



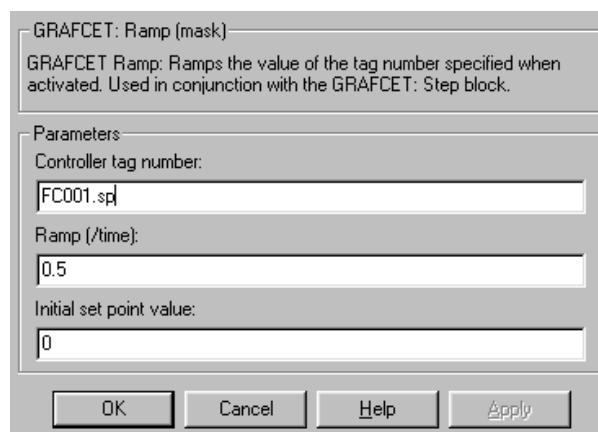
**Figure 3.13:** Generic receptivity block interface

### 3.3.7 Actions

*Actions* are enabled when the *step* associated with the *action* is activated. The different *actions* can be defined according to the different tasks executed during the scheduling of the supervisory controller. The generic blocks developed are listed and defined below:

#### Ramp

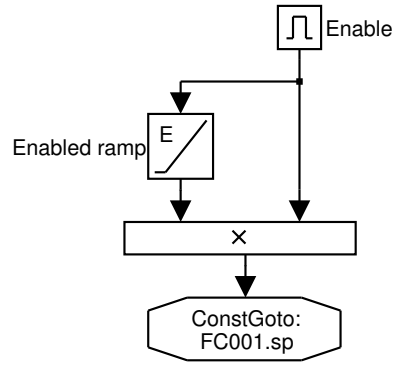
The value of a defined variable is ramped from an initial value at a set rate. The variable to be ramped is specified as a tag number (identifier number). The generic ramp interface (obtained when the current block is double-clicked) can be seen in figure 3.14



**Figure 3.14:** Generic ramp block interface

The implementation of the icon can be seen in figure 3.15 and shows that the **ConstGoto** function is used to assign the ramped value to the process. The ramp values are generated with an *enabled ramp block*, that generates the ramp outputs as soon as its input is non-zero.

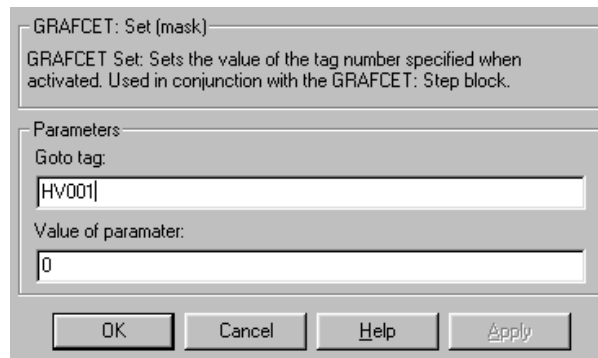




**Figure 3.15:** Ramp implementation

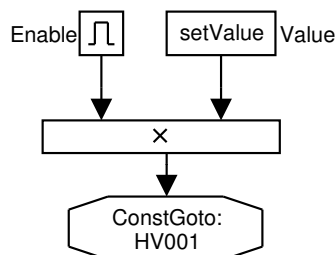
## Set

The specified variable is changed to the pre-defined value when the block is activated. The interface can be seen in figure 3.16.



**Figure 3.16:** Generic set block interface

The *set* implementation shows (figure 3.17) that the user defined *value* is sent to the process using the **ConstGoto** function, when the *set* action is enabled.



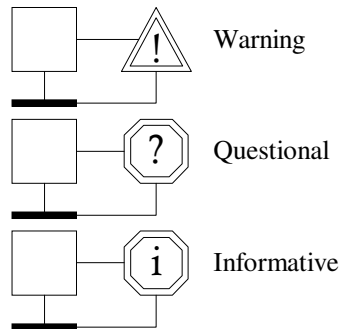
**Figure 3.17:** Set implementation

## Wait

The wait action block is used to pause the operation until the receptivity is true and the action block is de-activated. No user input is required for the use of the block.

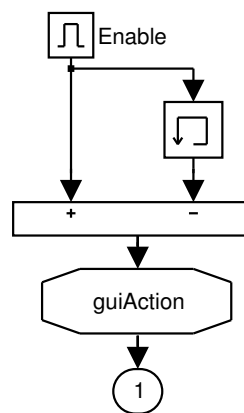
## Display

Information must sometimes be conveyed to the operator during the startup and shutdown procedure. Three different *display* icons were developed that can be used to display a certain message during the startup or shut down procedure. The different message types (type of icon displayed) that can be used are: information, question and warning (figure 3.18).



**Figure 3.18:** Generic display icons

The **GuiAction** function generates the display GUI with the user defined message when its input is equal to a value of one (figure 3.19). The *display output* is assigned a value of one when the “OK” button is pressed.

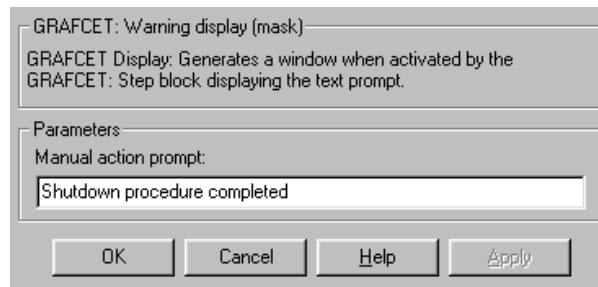


**Figure 3.19:** Display implementation

The message that should be displayed when the *display* is activated is supplied by the user, using the *display* GUI. The warning message GUI can be seen in figure 3.20

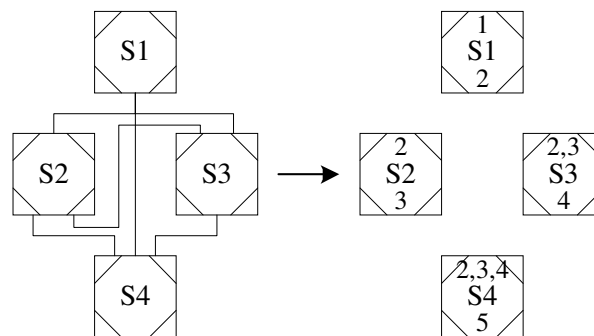
## 3.4 Macro

The *macro* is used to group different Grafcet sections, thereby reducing complex representations into simpler layered descriptions. The naming convention used to define a *macro* is divided into three parts. The top number describes the input *connection post* label, the middle number the *macro* name and the bottom number the output *connection post* label. The



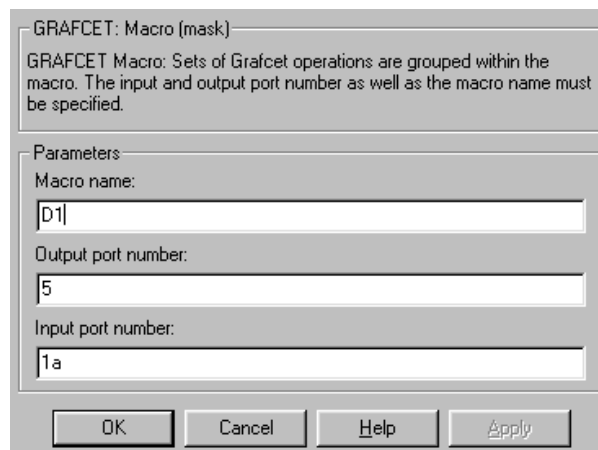
**Figure 3.20:** Warning display user input

naming convention reduces the complexity of macro connections as can be seen in figure 3.21, especially where multiple inputs or outputs are used.



**Figure 3.21:** Macro naming convention

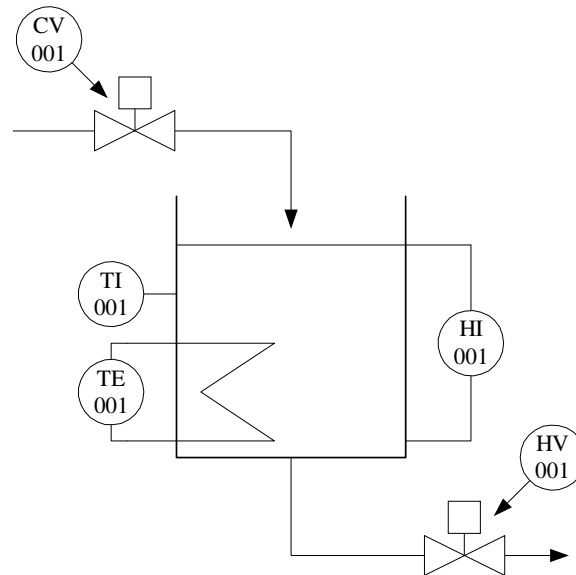
The *macroname* and labels are inserted by the user using the *macro* GUI as can be seen in figure 3.22



**Figure 3.22:** Macro graphical user interface

### 3.5 Heating tank example

An example is given to describe the interaction of the different generic components in the development of a sequential control problem. The process used for this purpose can be seen in figure 3.23 and is a tank that contains a heating element with measurements of liquid level (HI001) and temperature (TI001).



**Figure 3.23:** Heating tank example

The inlet and outlet flows can be manipulated using a control valve (CV001) and hand manipulated valve (HV001) respectively. The sequential procedure that must be implemented is given:

- **Step 1:** Open the feed valve (CV001) to fill the tank.
- **Step 2:** Wait until the water level (HI001) reaches one meter.
- **Step 3:** Start the heating of the fluid by switching the heating coil (TE001 = 500 kW) “on”.
- **Step 4:** Stop the inlet flow by closing the feed valve (CV001) when the level reaches four meters.
- **Step 5:** Notify the operator to open the hand valve (HV001) when the temperature of the fluid reaches  $130\text{ }^{\circ}\text{C}$  (400 K).

The Grafcet implementation of the sequential process can be seen in figure 3.24 clearly showing the current active (coloured) steps.

The *set* block is used to open and close the feed valve (CV001) and start the heating of the fluid by switching the heating element (TE001) on. The *wait* block is used to pause the

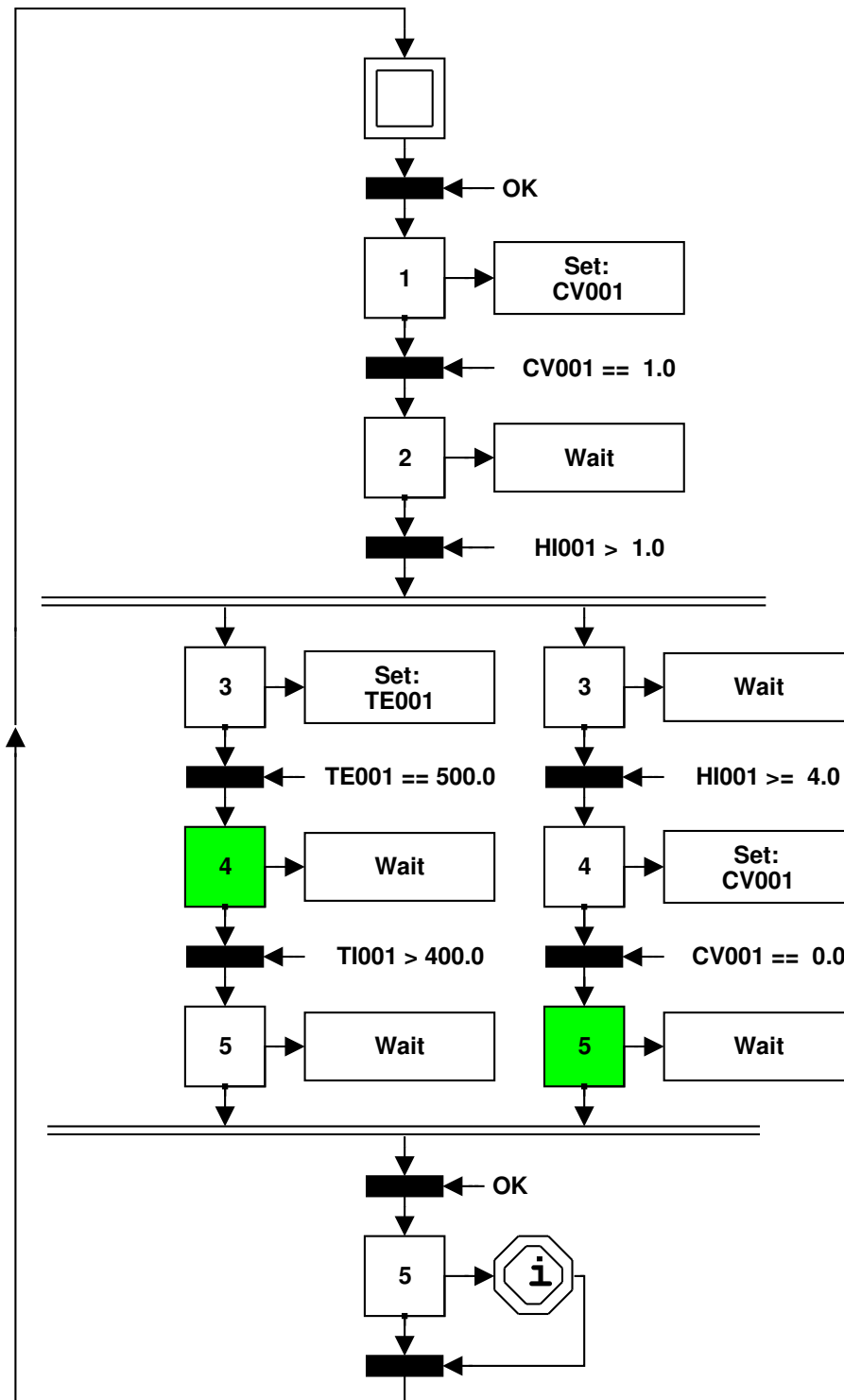
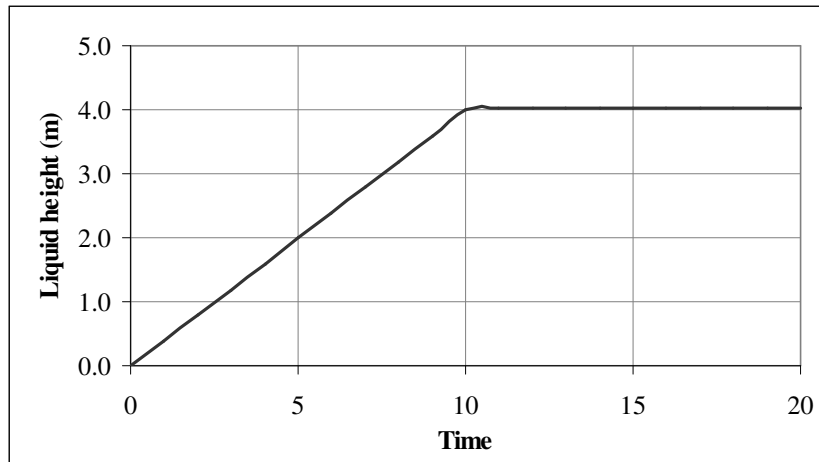


Figure 3.24: Grafcet implementation of the heating tank example

operation of the process until the *receptivity* block becomes true (i.e. the pre-defined level or temperature is reached). The operator notification (Step 5) is done using the *display* block.

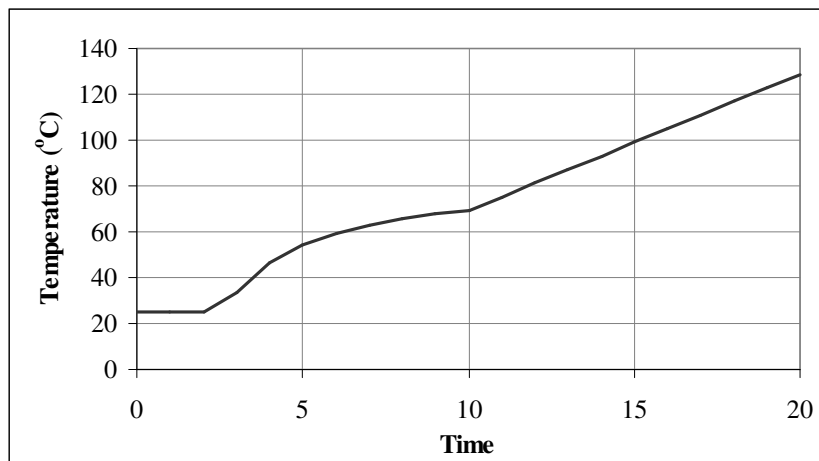
Two operations are conducted in parallel and both must be completed before the sequential operation can continue. The *convolution* and *devolution* bars are used by defining the AND operation in the drop-down list and two output or input ports.

The generated liquid level and temperature profiles for the sequential process can be seen in figures 3.25 and 3.26.



**Figure 3.25:** Height profile during the sequential process

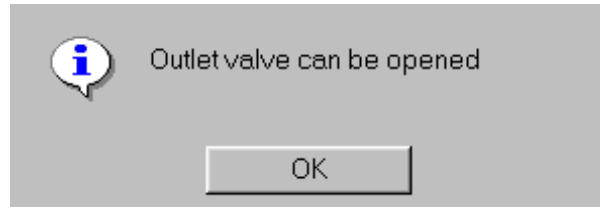
It can be seen that the level increases linearly until the pre-defined height is reached. The inlet flow is then stopped and the level stays constant. The temperature increases as soon as the liquid level is high enough (1m).



**Figure 3.26:** Temperature profile during the sequential process

The generated output that informs the operator that the outlet flow valve (HV001) can be opened can be seen in figure 3.27.

It can be seen that the implementation of the visual based approach gives an unambiguous presentation of the state of the automated process. The use of pre-defined generic compo-



**Figure 3.27:** Operator output generated during the last step

nents for the automated procedure implementation increases the ease of use and decreases the implementation time.

---

---

# CHAPTER 4

---

## Automated gasifier startup and shutdown

The generic components defined in the previous chapter were implemented on a more complicated example of a gasifier model, to investigate the ease of use and clarity of presentation of the automated startup of the unit operation, using the Grafcet formalism.

A short description of the different phases involved in the startup of a gasification unit is given. The controller configurations for the heating and air blown phases of the gasifier are given and the control tasks defined. The control tasks are then implemented to form the automated phase scheduling system. This control system is then implemented on the model of the gasification unit.

A fault in the air flow rate of the gasification unit is simulated to illustrate how the Grafcet countermeasure implementation can be used to add flexibility to the startup and shutdown procedure.

### 4.1 Startup and shutdown procedure synthesis

The implementation of a supervisory control system that will schedule the startup and shutdown of a processing must be preceded by steps to plan the startup operation and define the startup control structure. The procedure used for the synthesis of the automated startup and shutdown procedure for the gasification unit is listed:

**Step 1** Define the different phases of operation.

**Step 2** Identify the controlled and manipulated variables for each phase.

**Step 3** Develop the different control tasks for each of the controlled and associated manipulated variables.



**Step 4** Integrate the different control tasks and phases into the phased implementation supervisory control system.

**Step 5** Develop fault accommodation scheme for the different faults identified in the control task definition and integrate them into the countermeasure planning supervisory control system.

**Step 6** Implement the control system.

The different phases (stages in the procedure where the control and plant configuration stay the same) of operation must first be defined to represent the main goals that must be achieved during startup of the processing unit.

The different manipulated and controlled variables can be identified once the phases have been defined. The manipulated and controlled variables may change during the different phases of the startup and shutdown, that can influence the controllability of the process during the different phases. A control system must accordingly be developed for each phase of the plant operation.

The different control tasks can next be defined in order to:

- i) determine the sequence of events taking place during the procedure and,
- ii) to list the conditions that will indicate a fault occurring during the procedure.

The sequence of events and the list of fault conditions can then be defined according to the Grafset formalism and integrated into the phased implementation and fault accommodation supervisory controller structure. The system can then be implemented on the processing unit.

The different steps of the synthesis of the automated startup and shutdown procedure of the gasification unit will be discussed in further detail.

## **4.2 Step 1: Definition of phases**

Three phases can be identified for the gasification unit operation startup and shutdown. The three phases are listed and described below.

### **4.2.1 Heating phase**

The gasifier is purged from any stagnant gases in the reaction chamber using the inlet steam flow. The coal is loaded in the reaction chamber and the steam supply is used to heat the solid material to the specified reaction temperature.

### 4.2.2 Air blown phase

Air instead of oxygen is used to develop the different reaction zones as the combustion reaction takes place at a much lower temperature. This is due to the inert nitrogen introduced with the oxygen, that lowers the reaction temperature. The grate in the reaction chamber will not be damaged at the start when there is no ash layer. The ash layer can then be developed that will protect the grate from the high temperatures (1100-1500 °C) reached during the normal operation of the gasifier with oxygen.

### 4.2.3 Oxygen blown phase

Pure oxygen is introduced in the reaction chamber and the air supply is closed. The reaction chamber is then pressurized to its normal operating pressure and the reaction gases produced are introduced into the main gas header.

## 4.3 Step 2: Identify controlled and manipulated variables

The different control configurations for the different phases can next be defined. Only the first two phases (heating and air phases) are discussed further, as the principles conveyed in this chapter are the same for the other phases. The measurements emergency cut-off valves and the controller identification numbers (tag numbers) used to model the gasification unit, is listed in table 4.1. A detailed discussion on the development of the model of the gasification unit as well as its implementation is given in the appendix.

**Table 4.1:** Controller and measurement specifications

Tag number	Controlled variable
<i>Controllers</i>	
FC001	Air feed flow (kmol/min)
FC003	Steam feed flow (kmol/min)
SC001	Coal feed (kmol/min)
SC002	Ash removal rate (m <sup>3</sup> /min)
PC001	Reaction chamber pressure (kPa)
<i>Measurements</i>	
FI001	Air feed flow (kmol/min)
FI003	Steam feed flow (kmol/min)
PI001	Reaction chamber pressure (kPa)
<i>Valves</i>	
HV001	Air flow ECV
HV003	Steam flow ECV
HV006	Outlet flow to vent ECV

### 4.3.1 Heating phase

The controller configuration can be seen in figure 4.1. The only controlled variable during the heating phase of the gasification unit is the steam supply. The pressure is not regulated and the outlet steam is vented.

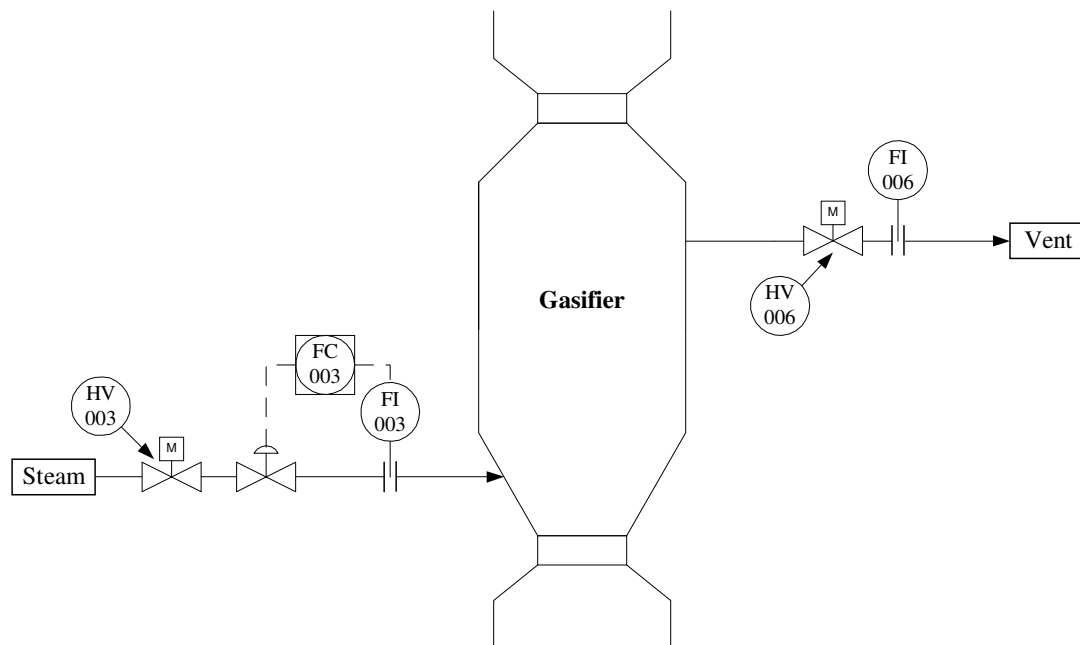


Figure 4.1: Heating phase control configuration

### 4.3.2 Air blown phase

The five controlled variables for the air blown phase are listed below: (figure 4.2).

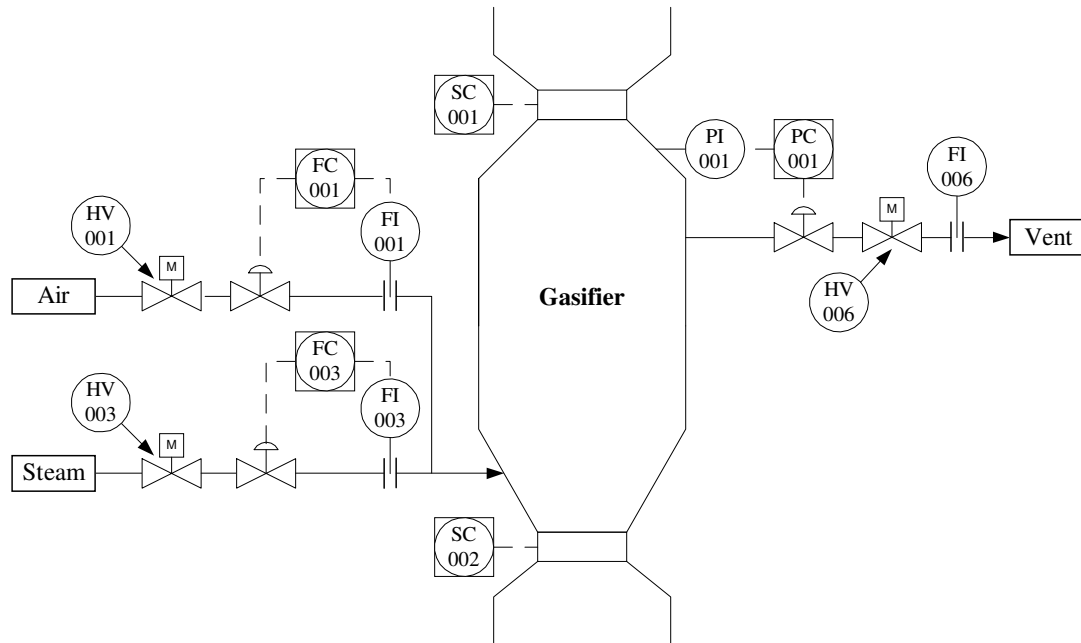
**Air flow** is ramped to its specified set-point.

**Steam flow** is introduced to reduce the temperature of the combustion reaction and as reagent for the gasification reactions. It is therefore ramped and controlled at the set-point.

**Reaction chamber pressure** is controlled by manipulating flow of the reaction gases out of the reaction chamber.

**Coal** must be fed to replenish that lost due to the reaction and therefore keep the fire-bed at a fixed height.

**Ash** produced during the reactions must be removed to avoid accumulation of solid material in the reaction chamber.



**Figure 4.2:** Air blown phase control configuration

## 4.4 Step 3: Control task development

The control tasks are defined using the task definition table discussed in the theoretical background. The control task definition for the heating phase can be seen in table 4.2, while the control definitions of the five controlled variables of the air blown phase can be seen in table 4.2.

The two tables show how the the sequences of operation of the different controlled variables are defined to be used for the design of the phased implementation system as well as the listing of the fault conditions that is used to define the countermeasure planning system.

**Table 4.2:** Control task definition for the heating phase

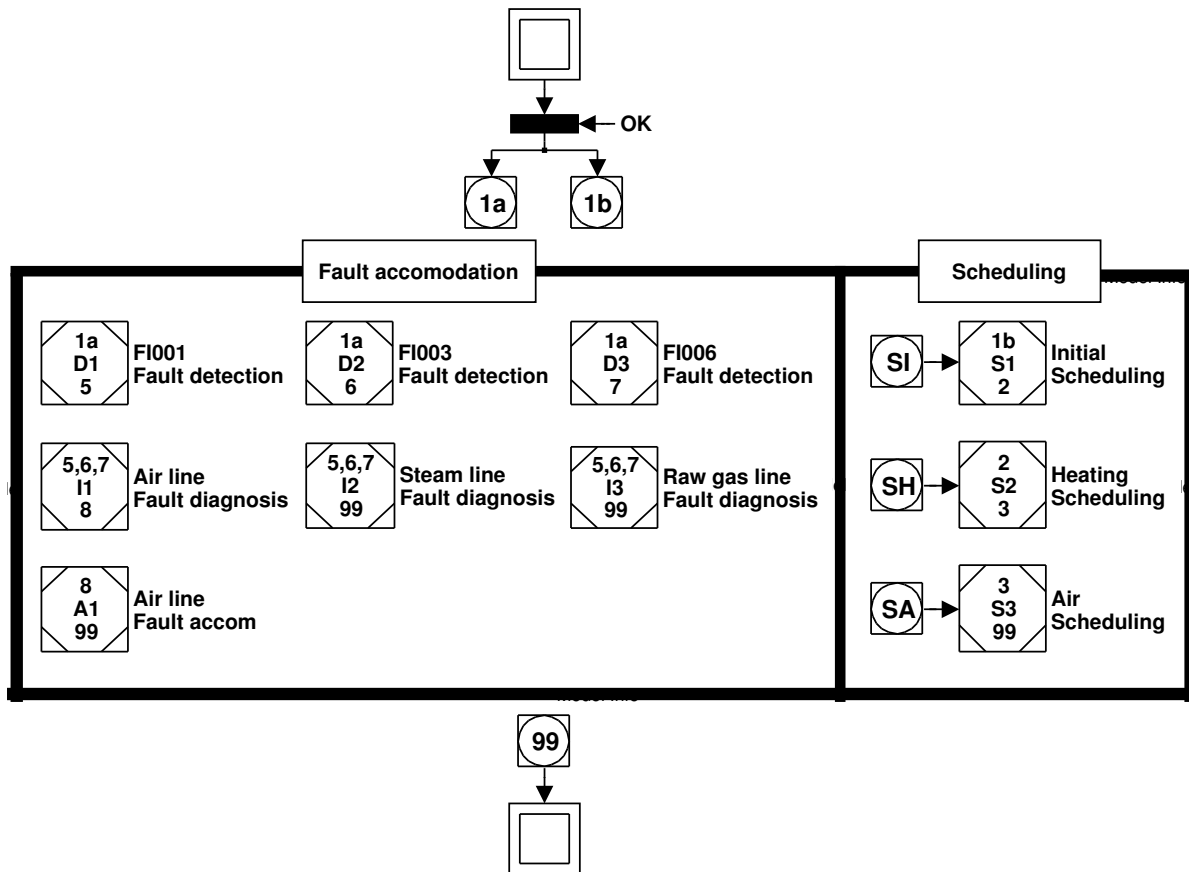
Goal:	Remove stagnant gases and heat the reaction chamber.
Strategic conditions:	Gasification unit is ready for commissioning, all emergency cut-off valves (ECV) are closed, and the controller modes are set to manual.
Execution conditions:	Steam flow must be less than 20 % of operating flow. There is exit gas flow.
Initial actions:	Open the steam and vent ECV's.
Control actions:	Ramp the steam flow to desired set point.
Final actions:	Stop the steam supply. Close the steam ECV.

**Table 4.3:** Control task definition for the air blown phase

Air flow	
Goal:	Introduce air into the feed stream.
Strategic conditions:	Gasification unit must be heated.
Execution conditions:	Sufficient air flow from supply. There is exit gas flow.
Initial actions:	Open the air ECV.
Control actions:	Ramp the air flow to desired set point.
Final actions:	None
Steam flow	
Goal:	Introduce steam into the feed stream.
Strategic conditions:	Gasification unit must be heated and purged.
Execution conditions:	Sufficient steam flow from supply. There is exit gas flow.
Initial actions:	Open the steam ECV
Control actions:	Ramp the steam flow to desired set point.
Final actions:	None
Reaction chamber pressure	
Goal:	Control the pressure in the combustion chamber.
Strategic conditions:	Sufficient steam and air supply.
Execution conditions:	Sufficient steam and air supply. There is exit gas flow. Pressure within the upper bound.
Initial actions:	Open the vent ECV
Control actions:	Control the pressure according to the set point.
Final actions:	None
Coal supply	
Goal:	Introduce fresh coal into the combustion chamber.
Strategic conditions:	A sufficiently large ash bed has formed.
Execution conditions:	Sufficient steam and air supply. There is exit gas flow.
Initial actions:	None
Control actions:	Introduce the specified amount of coal.
Final actions:	None
Ash removal	
Goal:	Remove ash from the gasification chamber.
Strategic conditions:	A sufficiently large ash bed has formed.
Execution conditions:	Sufficient steam and air supply. There is exit gas flow.
Initial actions:	None
Control actions:	Remove the specified amount of ash.
Final actions:	None

## 4.5 Supervisory controller overview

The supervisory control system of the automated startup and shutdown procedure for the heating and air blown phases can be seen in figure 4.3. The initial, heating and air blown phases constitute the phase implementation part of the supervisory control system while the fault detection, fault diagnosis and fault accommodation macros describes the countermeasure planning part. The three connection posts (SI, SA and SH) are used to suspend the actions inside



**Figure 4.3:** Supervisory control system of the gasification unit

the macros by setting them to zero if a fault should occur during the normal scheduling of the startup procedure.

The macros defined for the two different supervisory subsystems (phased implementation and countermeasure planning system) in this supervisory controller structure will be discussed in further detail.

## 4.6 Step 4: Integration of control tasks into the supervisory control system

The phased implementation system is used to schedule the different phases of the automated startup procedure. The detailed implementation (Grafcet implementation) of the three phases are subsequently defined in this section.

### 4.6.1 Initial phase

The initial phase (figure 4.4) is inserted as the first phase of the automated startup to ensure that all the controller modes are zero (i.e switched off) and all emergency cut-off valves are closed. The “set” block is used extensively to do this.

### 4.6.2 Heating phase

The steam ECV (HV001) and vent ECV (HV006) are opened and the steam flow rate (FI003) is ramped at 0.5 kmol/min until a maximum flow rate of 4 kmol/min is reached (figure 4.5). The heating phase is terminated when the top temperature reaches 330 °C.

### 4.6.3 Air blown phase

Air is introduced into the gasification chamber by ramping the air flow at 0.5 kmol/min until a flow rate of 2.9 kmol/min is reached. The ramp step must be monitored by the countermeasure planning system (figure 4.6). An output port is accordingly inserted on the step that will activate the monitoring of that step. Coal (SC001) is introduced and ash is removed (SC002) when the top temperature (TI001) reaches 800 K. The introduction of coal is to stabilise the upwards movement of the fire-bed. The pressure of the reaction chamber is set at 400 kPa and the raw gas is vented.

## 4.7 Step 5: Countermeasure planning system development

The countermeasure planning system is divided into three layers, namely fault detection, fault diagnosis and fault accommodation. The three layers were defined in different macros as was shown in figure 4.3. The detailed Grafcet implementation of the three layers is discussed further.

Three measurements are monitored during the heating and air blown phases (figure 4.3). They are: steam feed flow, air feed flow and outlet raw gas flow. A blocked flow line can therefore be detected by describing the unique characteristics associated with each fault. The implementation of a system that will detect, diagnose and accommodate blocked flow is of importance for the gasification unit as:

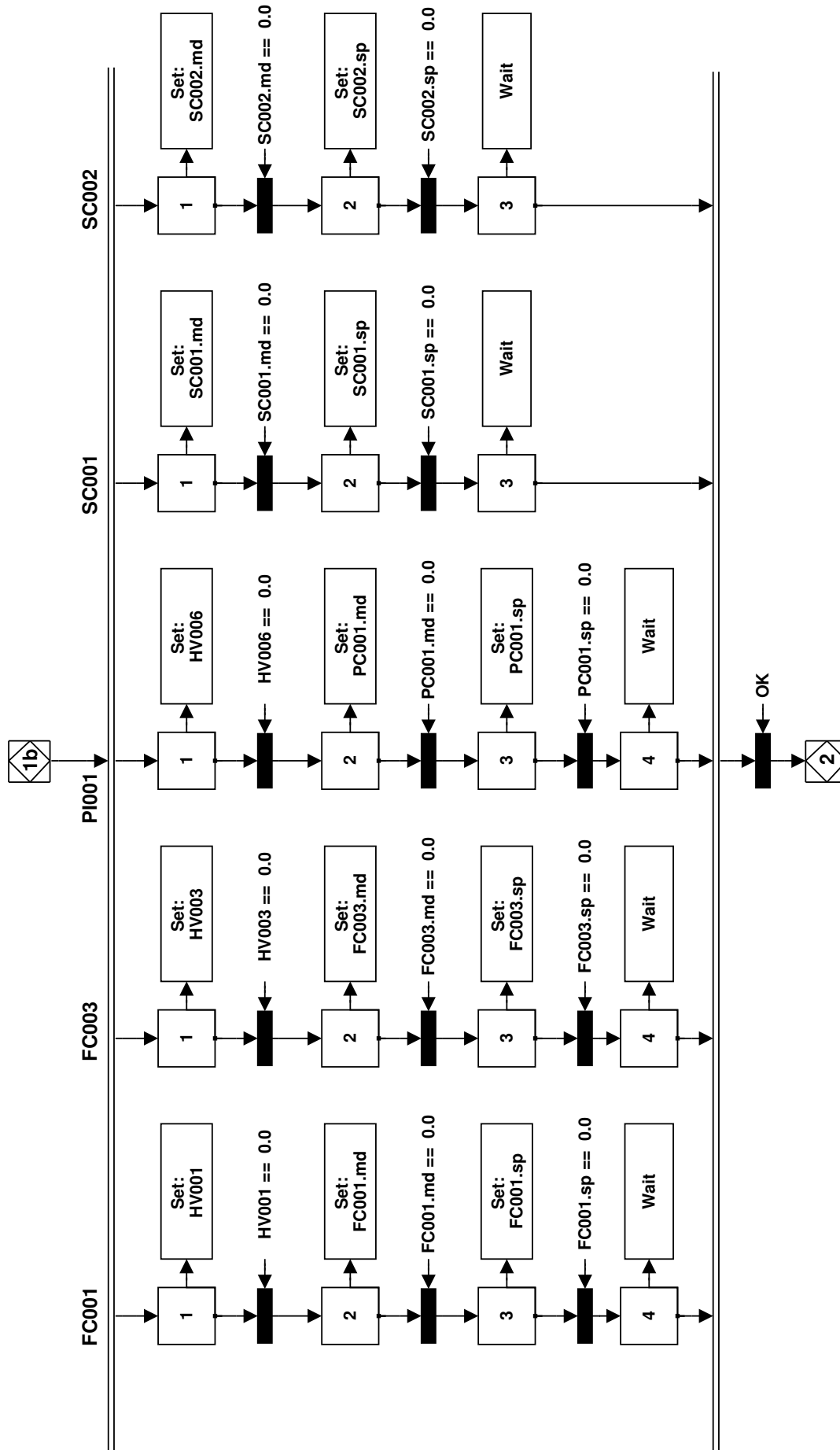


Figure 4.4: Control task definition of the initial phase



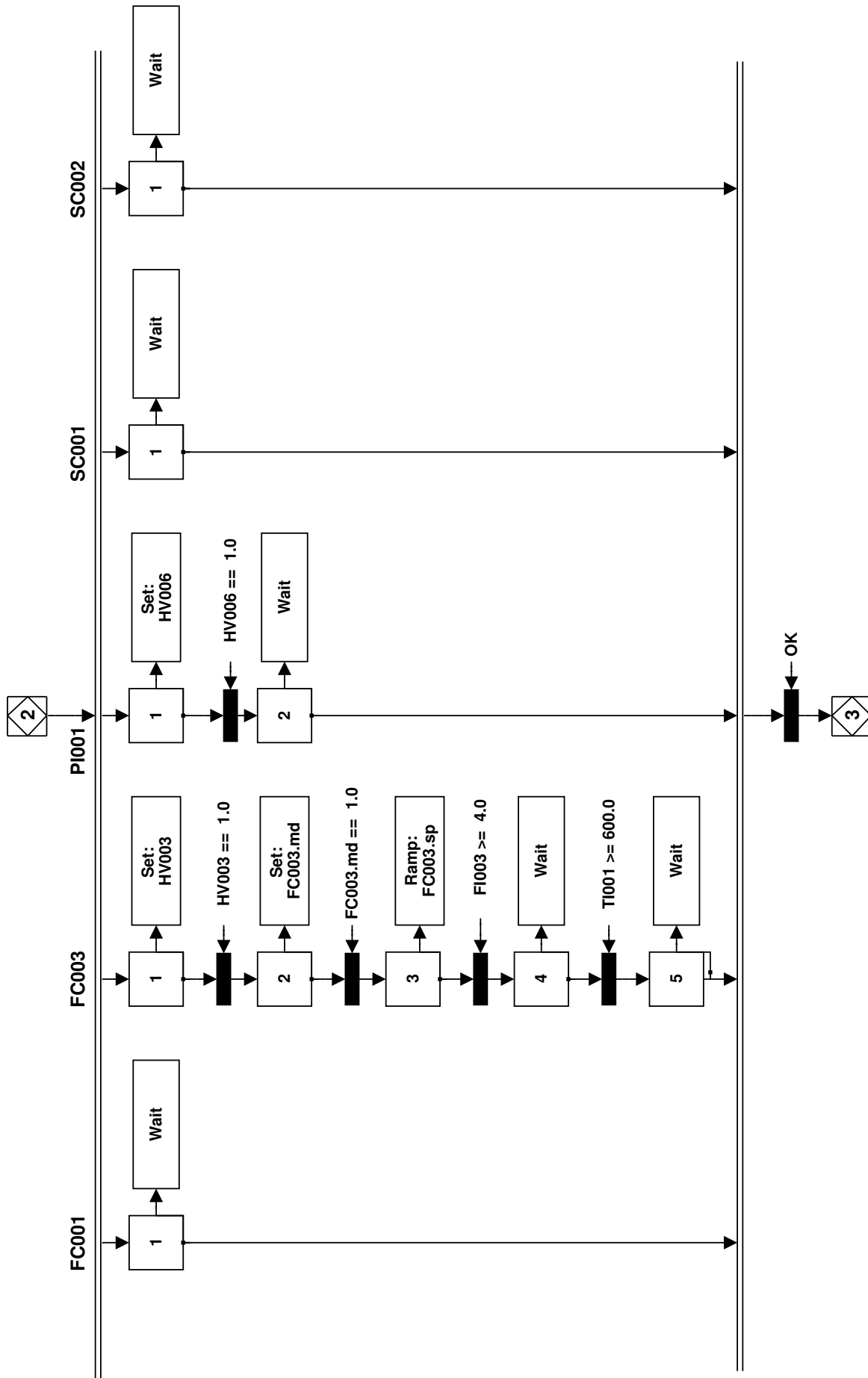


Figure 4.5: Control task definition of the heating phase

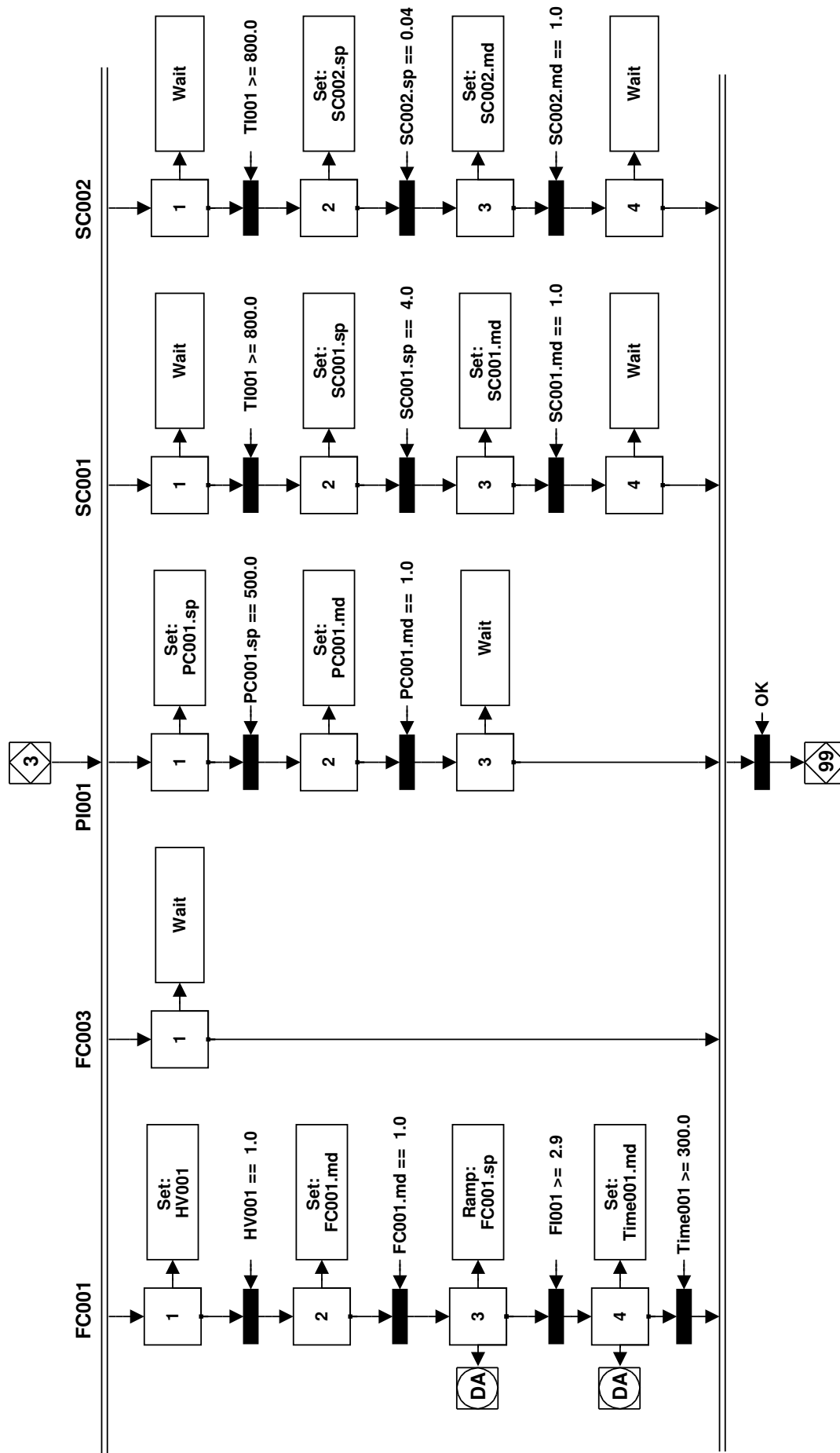


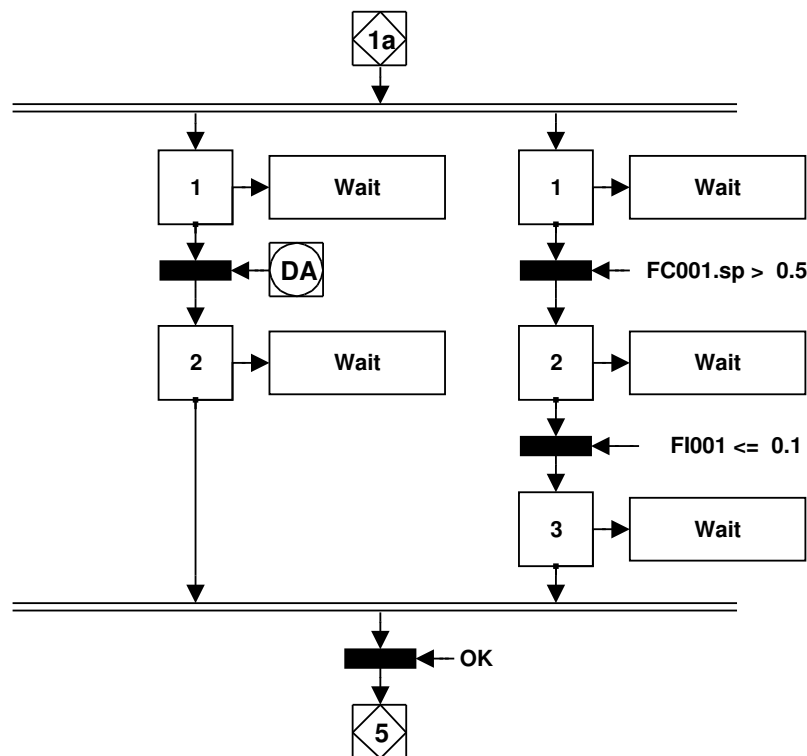
Figure 4.6: Control task definition of the air phase

- i) Condensate accumulation will form rust in the steam line that can inhibit flow, especially during the initial heating phase.
- ii) The air flow line can be blocked by debris as it is not used often.
- iii) Small coal particles and ash are entrained in the raw gas exit flow that can be deposited in the outlet line or valve to inhibit flow.
- iv) The steam or air feed flow valves can fail.

### 4.7.1 Fault detection

The steam feed and raw gas outlet flows are monitored throughout the automated implementation while the air flow is monitored at discrete steps. This is because the air flow will be stopped during the oxygen blown phase. False alarms will be generated if the air flow rate is monitored in the oxygen blown phase.

The Grafset fault detection implementation of the air flow monitor can be seen in figure 4.7. A fault will be detected as soon as the flow rate setpoint is above 0.5 kmol/min, the flow rate is below 0.1 kmol/min and the correct step is active. The fault detection of the steam and raw gas



**Figure 4.7:** Exit condition definition of the air phase

flow is not shown, but has the same implementation. A fault will be detected if the steam flow rate setpoint is larger than 0.5 kmol and the flow rate of steam or raw gas outlet flow is smaller than 0.1 kmol/min.

### 4.7.2 Fault diagnosis

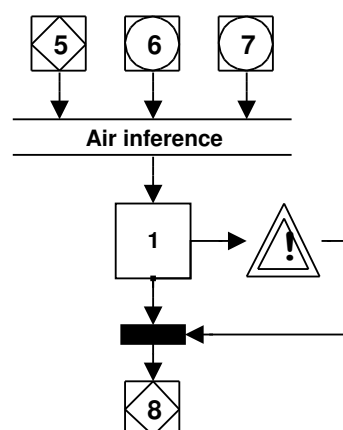
A rule based fault diagnosis system was developed and implemented in the countermeasure planning structure. The fault diagnosis system is needed to reduce the multiple alarms that are generated by the same root cause (i.e. the raw gas outlet alarm will be generated if the steam flow is blocked during the heating phase). It is therefore important to identify the root cause in order to implement the correct fault accommodation strategy, and reduce the confusion associated with abnormal conditions due to the multiple alarms that are generated.

The knowledge-based definition of the different faults detected can be seen in table 4.4 where the three flows are monitored and combined with the AND operation to give a root cause (i.e.  $FI001 \neq 0$  AND  $FI003 = 0$  AND  $FI006 \neq 0$  if the steam flow is blocked during the heating phase). Two alarm conditions exist for the steam flow root cause; one for each phase. The other two root cause conditions are unique throughout the two phases.

**Table 4.4:** Rule based fault diagnosis definition

Root cause	FI001 = 0	FI003 = 0	FI006 = 0
Steam flow blocked (heating phase)	Not true	True	Not true
Steam flow blocked (air phase)	Not true	True	True
Air flow blocked	True	Not true	Not true
Raw gas outlet flow blocked	Not true	Not true	True

The three measurements are tested by the three macros defined for each fault (figure 4.3). The implementation of the blocked air flow monitor is shown in figure 4.8. It can be seen that the connection post (DA) is used in the detection methodology. The detection system or monitor is therefore a discrete fault detection monitor. The other two monitors (not shown) are continuous.



**Figure 4.8:** Alarm management implementation

A message associated with a specific fault is displayed (using the warning *display*) as can be seen in figure 4.8.

### 4.7.3 Fault accommodation

The fault accommodation of a fault in the air feed flow entails suspension of the operations of the air phase (setting connection post SA equal to zero) and extending the heating phase until the problem is solved (see figure 4.9). The air flow controller is then set to manual, a message is generated that the heating phase has been extended and the operator must click “Continue” to restart the air phase. The air phase will be re-initialised by setting the connection post (SA) back to one as soon as the user confirmation is obtained.

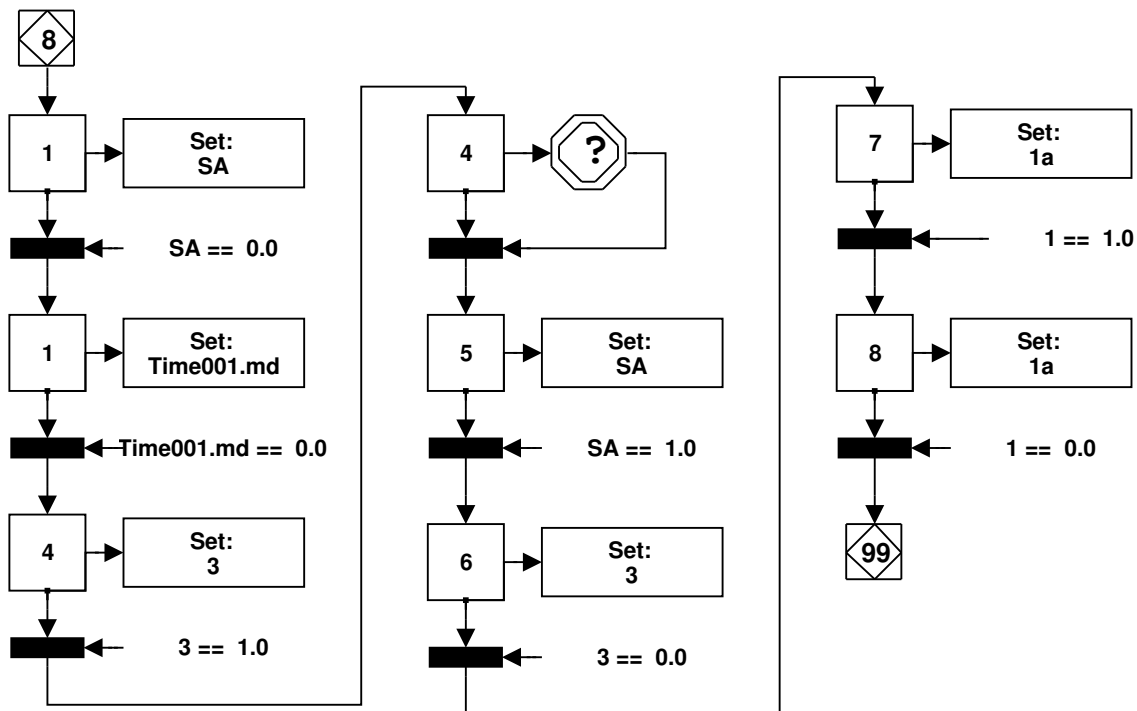


Figure 4.9: Air flow accommodation definition

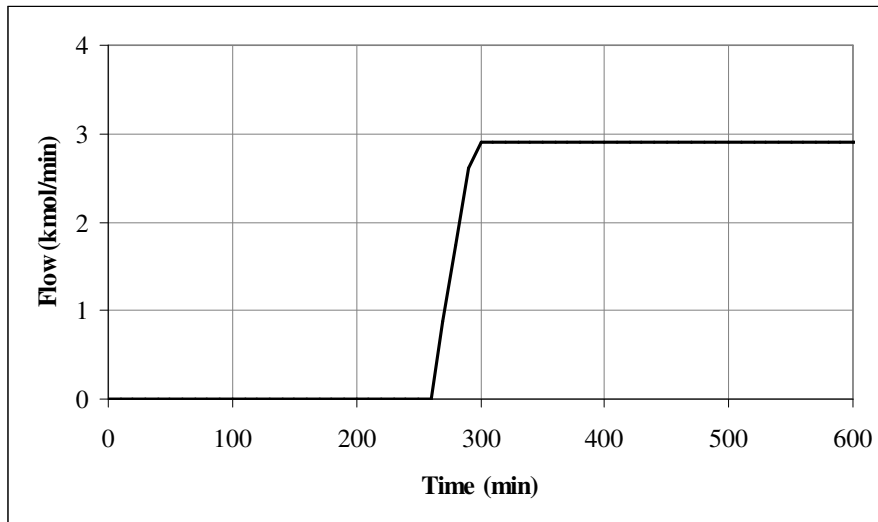
## 4.8 Example: Automated phase scheduling simulation

The automated procedure for the heating and air blown phases were simulated in SIMULINK using the supervisory control system as defined above and the model of the gasification unit as defined in the appendix.

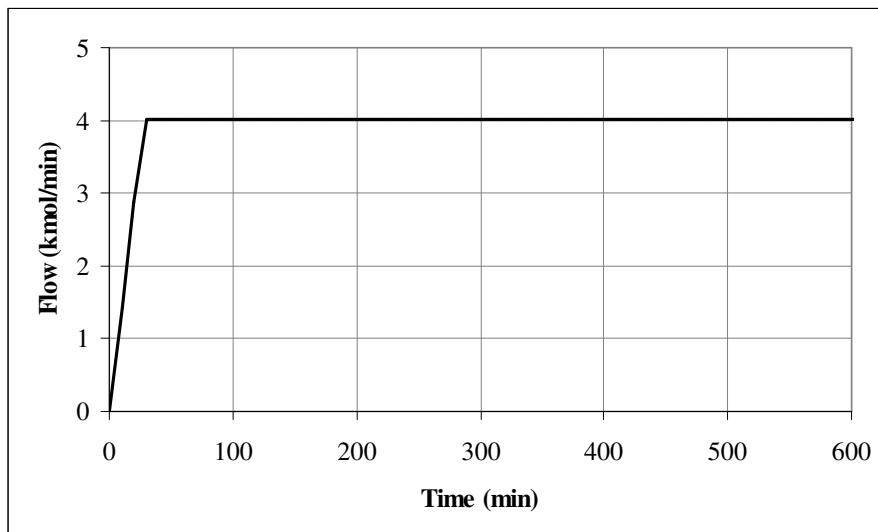
The air and steam input profiles for the startup can be seen in 4.10 and 4.11. The steam feed is ramped linearly to reduce the stress caused by the increase of pressure brought about by the warm gas. This will increase the life-time of the reaction chamber. The steam flow rate stays constant throughout the duration of the rest of the heating and air blown phases.

The air input is ramped linearly as soon as the top temperature of the gasifier reaches 330 °C (280 min) that signals the end of the heating phase and the beginning of the air phase.

The top (figure 4.13) and bottom (4.12) temperature profiles for the automated startup of



**Figure 4.10:** Air feed startup profile

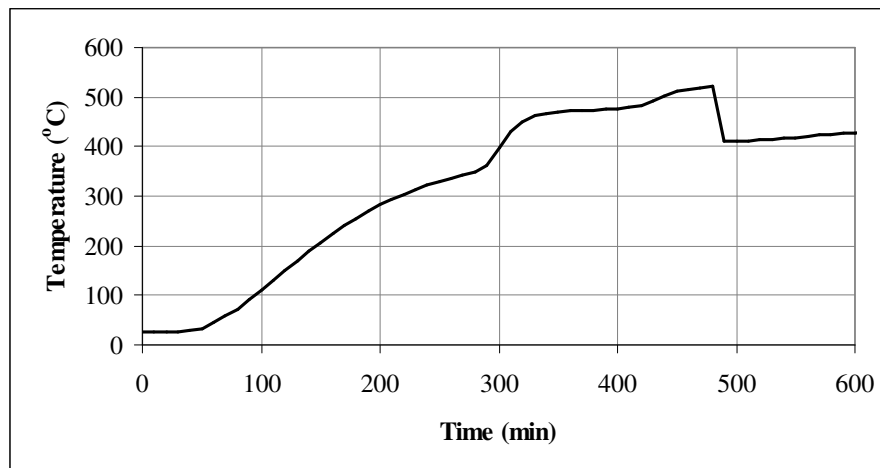


**Figure 4.11:** Steam feed startup profile

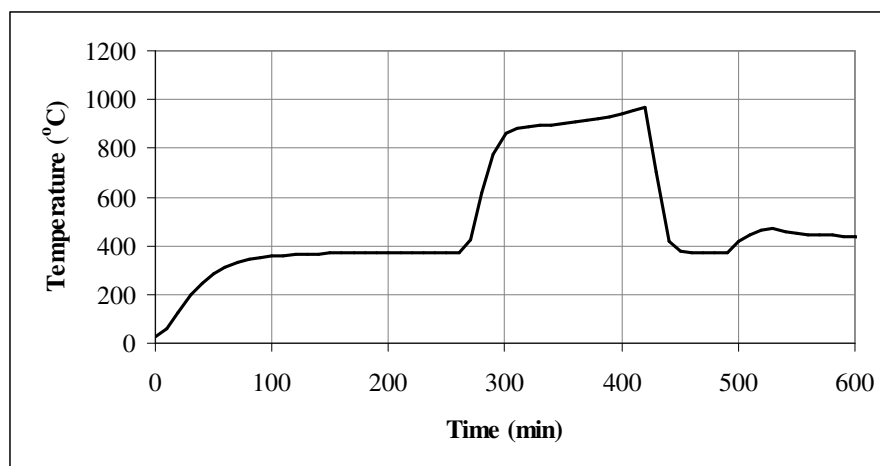
the heating and air blown phases is shown. The fire-bed progress at the bottom can clearly be seen in the bottom temperature profile. The temperature increases steadily as the steam warms the reaction bed during the heating phase.

The bottom temperature increases rapidly when air is introduced indicating that the exothermic combustion reaction takes place, but drops as the carbon becomes depleted to form the ash bed. The temperature rises and stabilises again, as soon as the solid flow is initiated because the fire-bed moves downwards with the solid phase to a new equilibrium position.

The top temperature rises steadily as the gas heats the solid bed. A more drastic increase in the top temperature is again seen as soon as air is introduced into the reaction chamber. This is because the fire-bed moves further up in the reaction chamber as the coal is depleted at the bottom of the reaction chamber as well as the higher temperatures generated that is associated



**Figure 4.12:** Top temperature profile



**Figure 4.13:** Bottom temperature profile

with the combustion reaction taking place.

The sudden drop in temperature is due to the coal feed that is introduced as soon as the ash layer is formed. This is because the coal is entering at temperatures cooler than that found in the reaction chamber and will cool the outlet gas temperature.

The supervisory controller scheduling of the startup phases can be seen in figure 4.14. The three detection macros are “green” showing that all the monitors are active. It can furthermore be seen that the current phase in the gasifier startup is the heating phase as the macro colour is “green” and the connection post used to suspend the current macro action (SH) is “white”. No fault is therefore detected or accommodated and the startup can continue normally.

This example shows that:

- i) the synthesis used to develop and implement the automated startup of the gasification unit gives a complete description of the definition of a supervisory controller that can be used to schedule this automated procedure, completely.

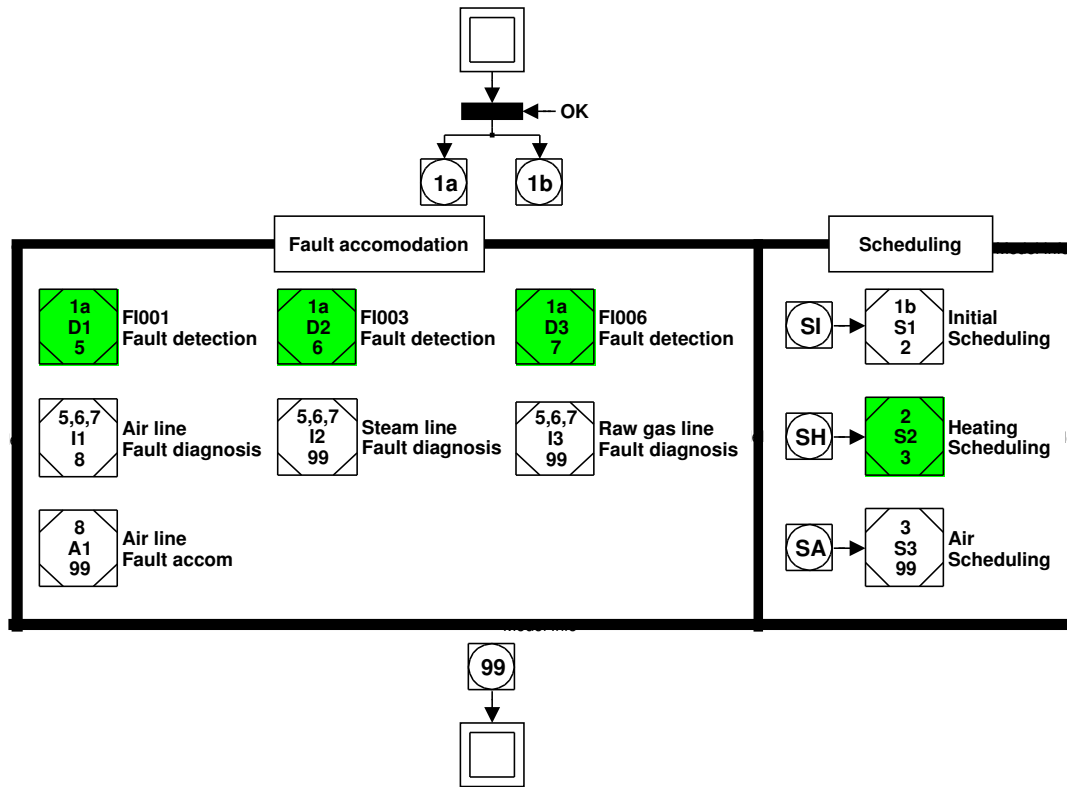


Figure 4.14: Supervisory controller during the normal startup procedure

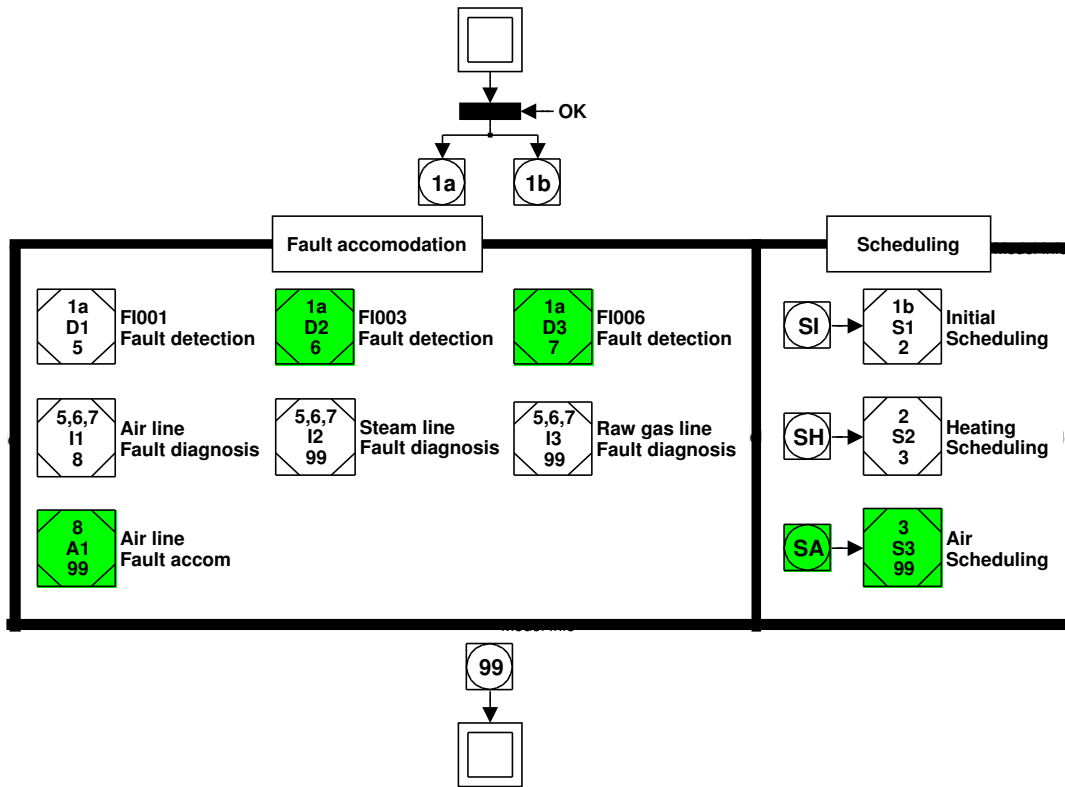
- ii) the generic components defined for use during the automated implementation of the startup and shutdown of processing units is sufficient in describing the complex interactions associated with the automated procedure.
- iii) the use of visual based icons and colours in representing the automated procedure, increases the understanding of the process as well as the efficiency of developing and implementing these procedures.
- iv) other non–Grafcet components such as the fault diagnosis knowledge–base can be integrated easily into the Grafcet formalism as long as the inputs and outputs of these systems comply with Boolean logic.

## 4.9 Example: Automated countermeasure planning simulation

This example is used to show the operation of the countermeasure planning system. A fault in the air flow was simulated by keeping the air flow zero during the air phase.

The fault was detected and diagnosed with the knowledge–based system and the heating phase prolonged to accommodate the fault. The countermeasure planning is shown in figure 4.15. It can be seen that the fault accomodation block is “green”, and is conformation





**Figure 4.15:** Supervisory control system during fault accommodation

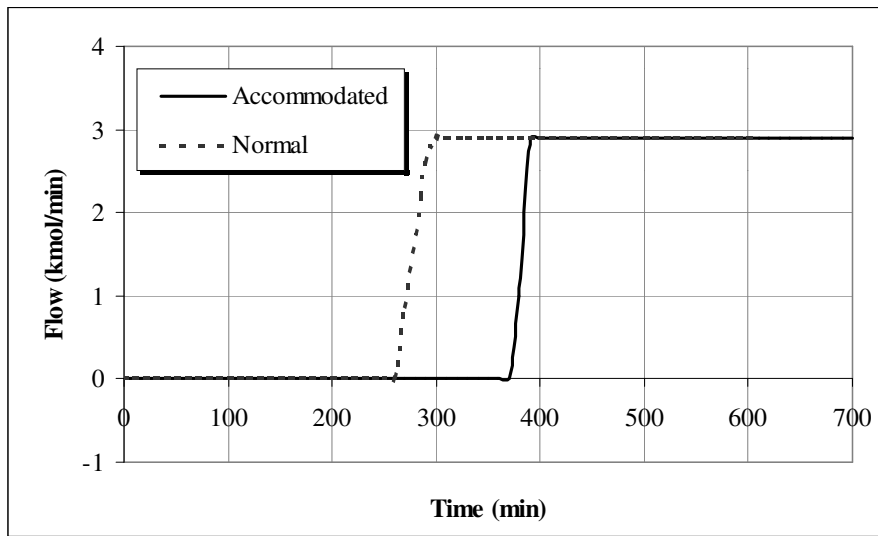
that a fault in the air line was detected and diagnosed correctly. The connection post used to suspend the air phase macro (SA) is also “green” to show that the macro operation is suspended. The air flow phase will then be re-initiated as soon as the operator confirmed that the problem was solved. The air line fault detection will then be activated and the air phase operation will commence.

The fault accommodated profile for the air flow can be seen in figure 4.16.

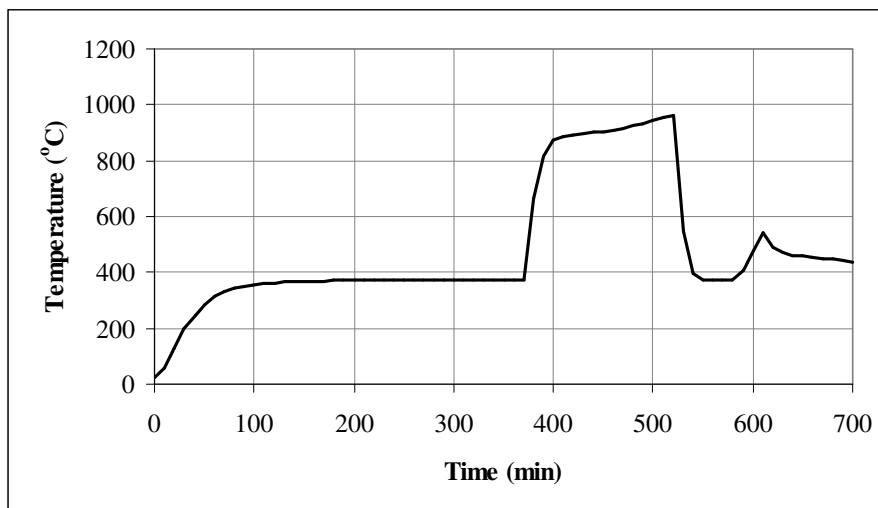
The temperature profile of the fault accommodated startup shows (figure 4.17) how the heating phase was prolonged to accommodate the air flow problem. The air blown phase continued normally as soon as the problem was solved. The secondary increase in temperature (600 min) is larger than the normal case, because the unit was initially heated more at the top during the prolonged heating phase. The oscillation of the fire bed before reaching steady state will therefore be more pronounced.

The example clearly shows how the incorporation of the fault accommodation system, that assists the operator, adds flexibility to the startup procedure in that different phases can be suspended or re-initialised to accommodate the different fault scenarios that can occur during the automated implementation of the startup procedure.

The visual approach gives a clear, unambiguous view of the complex scheduling that occurs during the countermeasure planning of a fault that is detected. This will increase operator confidence in the system because every action taken during the operation can be tracked and checked.



**Figure 4.16:** Fault accommodated air profile



**Figure 4.17:** Bottom temperature profile with fault accommodation

---

---

## CHAPTER 5

---

### Improvement of controller performance during automated startup procedures

Iterative learning is incorporated into the supervisory control system as a method to improve the base-layer controller performance during the startup or shutdown of a processing unit. A model describing the time dependent pressure profile of the reaction chamber during the final steps of the oxygen phase was developed and can be seen in the appendix. The iterative learning algorithm was subsequently applied to the model.

#### 5.1 Oxygen phase description

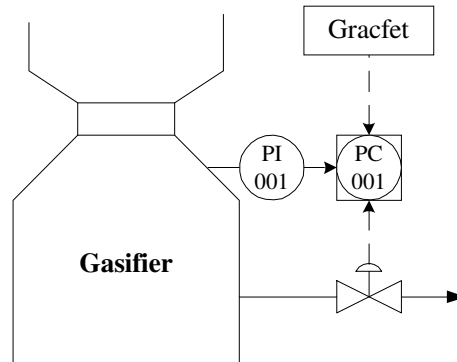
Oxygen is let into the reaction chamber during the oxygen blown phase of the automated startup of the gasification unit operation. The air flow is stopped and the steam flow is increased to operating conditions. The final control task is to ramp the pressure of the reaction chamber from 1500 kPa to the operating pressure of 2500 kPa with the supervisory control system (Grafcet). The control architecture for the pressure ramp can be seen in figure 5.1

The learning algorithm for current trial ( $k+1$ ) can be written:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \gamma \mathbf{e}_k \quad (5.1)$$

The current output profile is calculated from the previous profile output ( $\mathbf{u}_k$ ) and the resultant tracking error ( $\mathbf{e}_k$ ) for that trial.

The iterative step length  $\gamma$  is used to determine the convergence rate of the learning algorithm. A large step length can however lead to instability especially if the output measurement is subjected to noise. There is a trade-off between a fast rate of convergence and the instability of the output profile generation. A lead block can be added to improve stability, if needed.

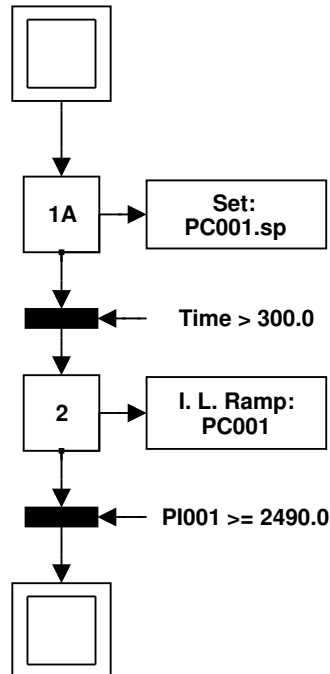


**Figure 5.1:** Pressure ramp operation

The calculation of the output profile is executed off-line. The output profile is determined in advance and can be read from a file during the on-line implementation.

The iterative learning algorithm can be incorporated into the supervisory control system as a re-defined “ramp action” block (figure 5.2) that will:

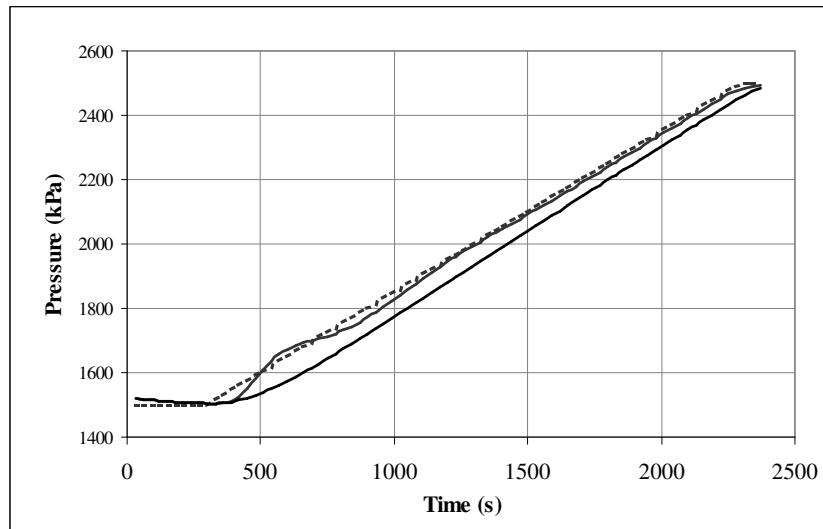
- read the generated output profile ( $\mathbf{u}_{k+1}$ ) from a file and implement it on the process and,
- write the current tracking error to a file that can be used for the generation of the next output profile ( $\mathbf{u}_{k+2}$ ).



**Figure 5.2:** Grafcet implementation of the iterative learning algorithm

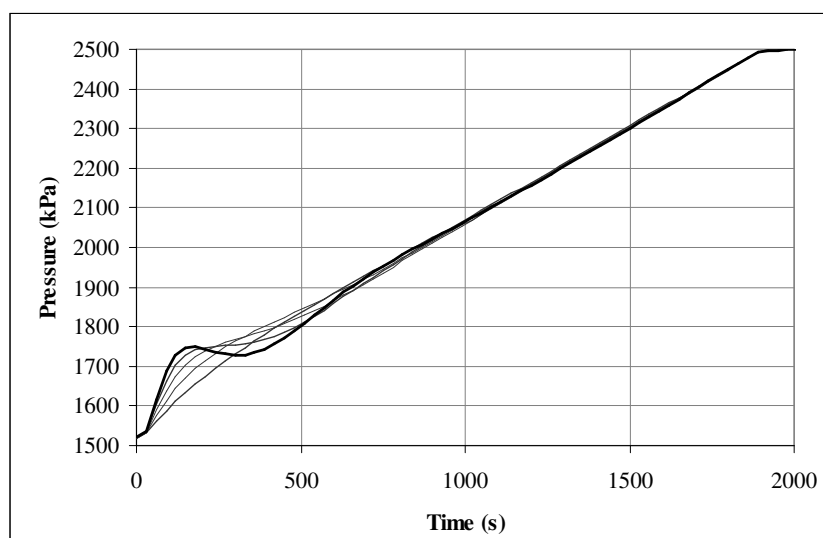
## 5.2 Model simulation and profile generation

The initial and final trial run (fifth iteration) pressure measurement during the ramp is seen in figure 5.3. The figure shows that the optimised pressure profile (solid lines) follows the setpoint (dashed line) more closely than the initial profile. This will reduce the startup time as the operating pressure will be reached faster.



**Figure 5.3:** Initial and final pressure measurement

The simulated output profiles for the pressure ramp (for five iterations) can be seen in figure 5.4. It can be seen from the figure how the initial linear ramp specification to the pressure controller is changed to give a more complex non-linear profile that reduces the tracking error.



**Figure 5.4:** Controller setpoint for feedback implementation

---

---

# CHAPTER 6

---

## Conclusions

The development of automated startup and shutdown procedures and their implementation as a supervisory controller was studied. The following conclusions were made:

- The use of automated startup and shutdown procedures increase the safety and profitability of the plant either by reducing the operator workload and incorporating fault accommodation strategies. The definition and development of generic components that can be used for the implementation of automated startup and shut down procedures will therefore streamline the implementation process to increase the profitability and use of these processes.
- An automated startup and shutdown procedure must monitor the process and schedule its operation. It must therefore be incorporated into the standard supervisory controller configuration that is used for normal operation. Grafcet (methodology used originally to represent PLC's) can be used to represent the supervisory control implementation (Årsén, 1994), as it was developed specifically for visually representing sequential operations.
- The aspects of fault detection, diagnosis and accommodation can be integrated into the Grafcet methodology. The supervisory control system must be divided into a phase implementation and a countermeasure planning part in order to accommodate the different aspects.

Generic visual-based icons were defined in SIMULINK for the synthesis and development of any automated startup or shutdown procedure. The visual-based approach is superior to the text-based approach in that the control system is developed faster and represented more clearly, as an example of the sequential heating of a fluid in a tank showed.

The startup procedure was synthesized, developed and implemented for the heating and air blown phases, for a non-linear model of a gasification unit, that incorporated the aspects of

## CHAPTER 6. CONCLUSIONS

fault detection, fault diagnosis and fault accommodation. A rule-based alarm management system was defined and was incorporated as a fault diagnosis system into the supervisory control system.

The simulations conducted showed that the processing unit can be started up successfully and unambiguously using the pre-defined generic blocks and that the incorporation of fault accommodation adds flexibility to the startup procedure. The use of visual icons and colours furthermore increased the understanding of the interaction of the different components and the efficiency in the development of these procedures.

It was finally showed that iterative learning can be used to optimise controller performance during the scheduling of non-linear phases. The iterative learning algorithm was integrated into the Grafset formalism by defining a new iterative ramp action block. The block will output the optimized profile to the process from a file and store the current tracking error to be used for the next trial run.

---

---

# CHAPTER 7

---

## Recommendations

This investigation was focused on defining and developing an automated procedure for the implementation of supervisory controllers, with the specific application to automated startup and shutdown.

It is recommended that further investigations be conducted on:

- implementing the tools developed for the automated startup and shutdown on a real unit operation in order to study the effects of noise, uncertainty and real-time implementation. These aspects were not incorporated into this study as only models were used to test the definitions.
- the incorporation of the normal phase operation as part of the scheduling of the supervisory controller defined here, in order to provide a unified view of process control, scheduling and optimization.
- the use of more effective techniques to detect and diagnose faults with the use of, for example, neural networks and fuzzy logic can be investigated as well as its implementation into the Grafset formalism.



---

## Bibliography

- Årsén, K. (1994) “Grafcet for intelligent supervisory control applications”, *Automatica*, 30 (10), 1513–1525.
- Årsén, K. (1996) “Integrated control and diagnosis of sequential processes”, *Computers and chemical engineering*, 4 (9), 1277–1286.
- Amann, N.; Owens, D. and Rogers, E. (1996) “Iterative learning control using optimal feedback and feedforward actions”, *International Journal of Control*, 65 (2), 277–293.
- Bahar, M.; Jørgensen, S.; Koggersbol, A. and Yazdi, H. (1995) “Knowledge-based control structuring of a distillation plant start-up”, *Control Engineering Practice*, 3 (3), 423–430.
- de Ponte, P.; Gouws, F.; Keuler, J. and Schoeman, B. (2001) “Automated gasifier start-up”, Internal report, University of Pretoria, South Africa.
- Fogler, H. (1992) *Elements of Chemical Reaction Engineering*, volume 2 Prentice Hall, London.
- Ganguly, S. and Saraf, D. (1993) “Startup of a distillation column using non-linear analytical model predictive control”, *Industrial Engineering and Chemical Research*, 32, 1667–1675.
- Han, M. and Park, S. (1999) “Startup of distillation columns using profile position control based on a nonlinear wave model”, *Industrial Engineering and Chemistry Research*, 38, 1565–1574.
- Hochgesand, G. (1989) “Gas production”, in *Ullman’s Encyclopedia of Industrial Chemistry*, Elvers, B. (Ed.), volume A12, VCH Verlagsgesellschaft, Germany.
- Isermann, I. (1997) “Supervision, fault-detection and fault-diagnosis methods - an introduction”, *Control Engineering Practice*, 5 (5), 639–652.
- Johansson, S. and Öhman, M. (1995) “Prototype implementation of the PLC standard IEC 1131-3”, Master’s thesis, Lund Institute of Technology, Sweden.

- Johnsson, C. and Årsén, K. (1998) “Grafchart for recipe-based batch control”, *Computers and Chemical Engineering*, 22 (12), 1811–1828.
- Kheir, N.; m, K. A.; Auslander, D.; Cheok, K.; Franklin, G.; Masten, M. and Rabins, M. (1996) “Control systems engineering education”, *Automatica*, 32 (2), 147–166.
- Kosky, P. and Floess, J. (1980) “Global model for countercurrent coal gasifiers”, *Industrial engineering and chemical process design and development*, 19, 586–592.
- Lee, K.; Bang, S.; Yi, S.; Son, J. and Yoon, S. (1996) “Iterative learning control of heat-up phase for a batch polymerization reactor”, *Journal of process control*, 6 (4), 255–262.
- Lee, K.; Chin, I.; Lee, H. and Lee, J. (1999) “Model predictive control technique combined with iterative learning for batch processes”, *AIChE Journal*, 45 (10), 2175–2187.
- Levenspiel, O. (1999) *Chemical Reaction Engineering*, John Wiley and Sons, New York.
- Ludwig, R.; Estévez, L. and Smith, J. (1985) “The differential-reactor approximation under dynamic conditions”, *Chemical Engineering Science*, 40 (5), 759–767.
- Mandano, E.; Marcos, J. and Pérez, S. (1996) *Programmable Logic Devices and Logic Controllers*, Prentice Hall, London.
- Marlin, T. (2000) *Process Control, Designing Processes and Control Systems for Dynamic Performance*, McGraw Hill, second edition.
- Matsumoto, H.; Kaji, A.; Nigawara, S.; Tokuhira, M. and Suzuki, Y. (1993) “An operation support expert system based on on-line dynamic simulation and fuzzy reasoning for startup schedule optimization in fossil power plants”, *IEEE Transactions on Energy Conversion*, 8 (4), 674–680.
- Mezghani, M.; Roux, G.; Cabassud, M.; Dahou, B.; Lann, M. L. and Gasamatta, G. (2001) “Robust iterative learning control of an exothermic semi-batch chemical reactor”, *Mathematics and Computers in Simulation*, 57, 367–385.
- Monazam, E. and Shadle, L. (1998) “Predictive tool to aid design and operations of pressurized fixed coal bed gasifiers”, *Industrial Engineering and Chemistry Research*, 37, 120–130.
- Perry, R. and Green, D. (1997) *Perry’s chemical engineers’ handbook*, McGraw-Hill, New York.
- Pradubsripetch, D.; Lee, S.; Adriani, A. and Naka, Y. (1996) “Real time generation of operating procedures to support flexible start-up operation”, *Computers and chemical engineering*, 20 (Suppl.), 1203–1208.

## BIBLIOGRAPHY

- Rengasamy, R. (1995) *A framework for integrating process monitoring, diagnosis and supervisory control*, PhD thesis, Purdue University, Purdue.
- Shaikh, A. and Lee, C. (1995) “Minimize flaring during ethylene plant startup”, *Hydrocarbon processing*, July, 89–96.
- Skogestad, S. (2000) “Plantwide control: the search for the self-optimizing control structure”, *Journal of Process Control*, 10, 487–507.
- Smith, J.; van Ness, H. and Abbott, M. (1996) *Introduction to chemical engineering thermodynamics*, McGraw-Hill, New York.
- Sørensen, E. and Skogestad, S. (1996) “Optimal startup procedures for batch distillation”, *Computers and chemical engineering*, 20 (Suppl.), S1257–S1262.
- Xu, J.; Hu, Q.; Lee, T. and Yamamoto, S. (2001) “Iterative learning control with smith time delay compensator for batch processes”, *Journal of Process Control*, 11, 321–328.
- Yazdi, H. (1997) *Control and Supervision of Event-Driven Systems*, PhD thesis, University of Lund, Sweden.
- Yoon, H.; Wei, J. and Denn, M. (1978) “A model for moving-bed coal gasification reactors”, *AIChE Journal*, 24 (5), 885–903.

---

---

# APPENDIX A

---

## Fixed bed gasifier model

### A.1 Gasification of coal

Gasification of coal is an essential first step in the coal-based petrochemical industry to produce hydrogen ( $H_2$ ) rich carbon monoxide (CO) and carbon dioxide ( $CO_2$ ). This raw gas is used (after the removal of  $CO_2$ ) for the production of a large variety of valuable organic compounds including waxes and surfactants. The coal is converted into the raw gas by reaction with steam ( $H_2O$ ) and oxygen ( $O_2$ ).

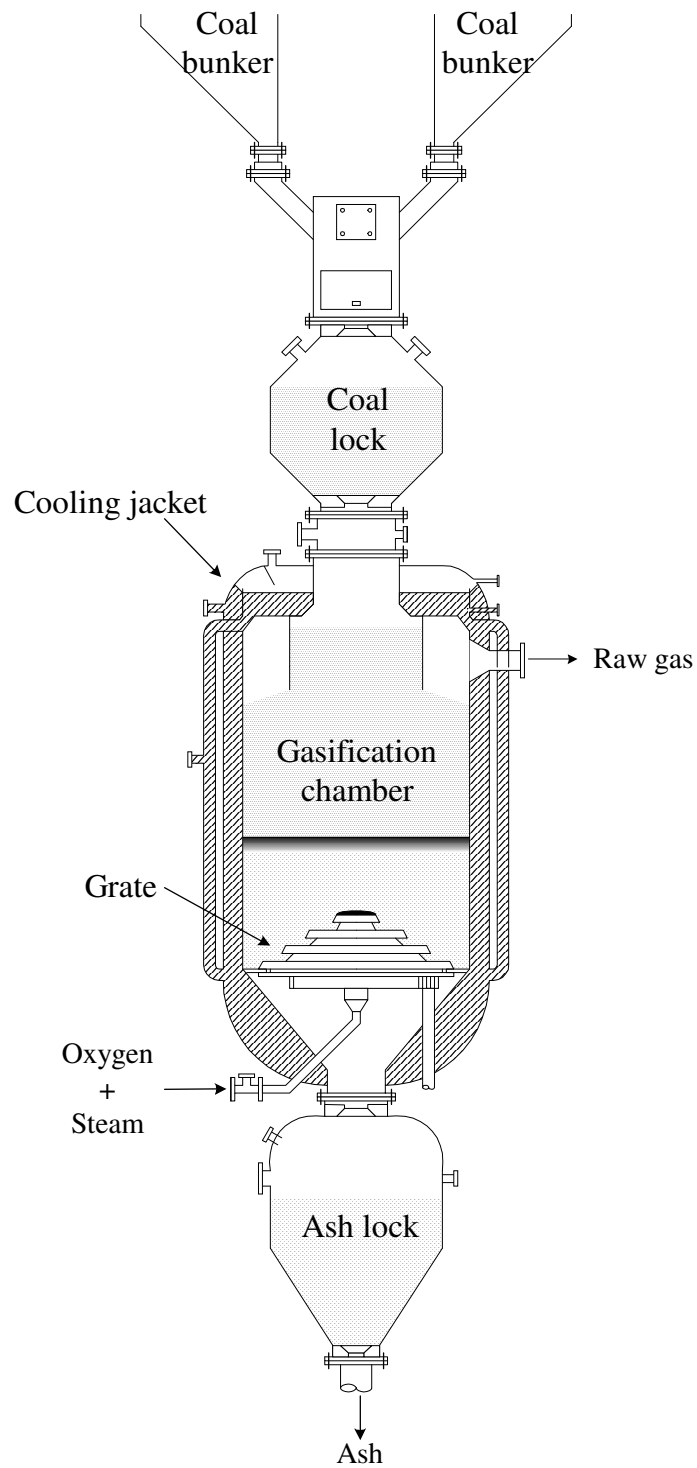
The other components that are generated are volatile compounds such as ammonia ( $NH_3$ ), hydrogen sulphide ( $H_2S$ ) and sulphur dioxide ( $SO_2$ ) that are generated due to the elemental sulphur and nitrogen found in the coal as well as longer chained tars, oils and naphtha.

Gasification takes place in a gasification unit or gasifier. There are many types of gasification units and are characterised by the direction of the flow of solid and gas phases as well as the particle size of the coal bed. The gasifier modelled is a counter-current gasifier and is known as a Lurgi moving-bed gasifier and is used for the gasification of coal with an average size distribution of 3–50 mm.

### A.2 Gasification unit description

The gasification unit can be broken down into the following sections (figure A.1):

- Coal bunker and lock
- Gasification chamber
- Ash lock and condenser



**Figure A.1:** The gasification unit

### **A.2.1 Coal bunker and lock**

The coal bunker is situated at the top of the gasification unit and the coal is fed into the bunker with a conveyor belt that is operated by a high and low level trip switch (de Ponte et al., 2001).

The coal lock is used to feed coal into the pressurised (2700 kPa) gasification chamber. The coal lock is sealed on both sides by a hydraulic valve that is opened and closed according to a timed cycle (de Ponte et al., 2001).

### **A.2.2 Gasification chamber**

The gasification chamber is a double walled vessel. The space between the two walls is filled with a mixture of water and high-pressure steam that is generated due to the reaction heat in the chamber. The steam generated is mixed with oxygen and is fed into the gasification chamber as one of the reactants.

The gasification chamber houses a variable speed-rotating grate that determines the amount of ash removed. The grate has a number of important functions:

- It distributes the reactant gas evenly through holes into the ash bed.
- It carries ash out of the bed and breaks lumps to prevent blockage.
- Controls the height of the reaction zone in the gasification chamber.

### **A.2.3 Ash lock and condenser**

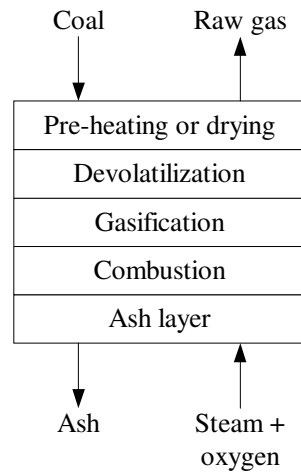
The ash lock is situated at the bottom of the gasification chamber. It is sealed, as is the case with the coal lock, at the top and bottom by two hydraulic valves. The ash lock is depressurised and emptied according to a timed cycle (de Ponte et al., 2001).

## **A.3 Reaction chemistry**

The reactions that take place inside the gasification chamber can be divided into different reaction zones. The zones identified can be seen in figure A.2 (de Ponte et al., 2001):

### **A.3.1 Ash layer**

No reaction takes place inside the ash layer, but the inlet gas flow is heated from about 360 to 450 °C, due to direct contact, as it flows through the ash layer. The ash is cooled to 400 °C at the bottom. The ash layer furthermore acts as a distributor of the reactants (Hochgesand, 1989).



**Figure A.2:** Reaction zones inside the gasification chamber

### A.3.2 Combustion zone

Oxygen ( $O_2$ ) and char (C) reacts to form carbon dioxide ( $CO_2$ ) according to the following reaction:

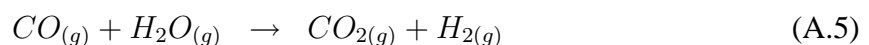
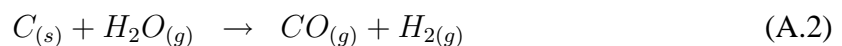


This reaction is exothermal, extremely rapid and proceeds to completion with the disappearance of oxygen ( $O_2$ ) (Monazam & Shadle, 1998). This zone is therefore the warmest in the gasification chamber and the temperature can reach 1500 °C.

Excess steam is supplied to the gasification chamber to cool the gasification reaction temperature (Hochgesand, 1989). The reaction heat generated is used to fuel endothermic reactions in the reaction zones at the top of the combustion zone.

### A.3.3 Gasification zone

Char from the devolatilization zone comes in contact with steam as well as the hot combustion gases generated in the combustion zone to produce mainly hydrogen ( $H_2$ ), carbon monoxide (CO) and methane ( $CH_4$ ) by reacting with carbon dioxide ( $CO_2$ ), char (C), and carbon monoxide (CO). These reactions are therefore endothermic and use the heat generated in the combustion zone directly beneath it. The reactions are (Monazam & Shadle, 1998):



Reactions A.2 and A.3 are slow, endothermic and favoured at temperatures above 750 °C.

Reaction A.4 is exothermic and slow and is favoured at elevated pressures and a bed temperature below 600 °C. The water-gas shift reaction (reaction A.5) is catalysed by a coal surface and is subsequently rapid and favoured at temperatures above 600 °C. (Monazam & Shadle, 1998)

### A.3.4 Devolatilization zone

The hot product gasses produced in the gasification and combustion zones come in contact with the coal to yield gaseous compounds and char (pyrolysis) (Hochgesand, 1989). The tars are also cracked to produce oils (de Ponte et al., 2001).

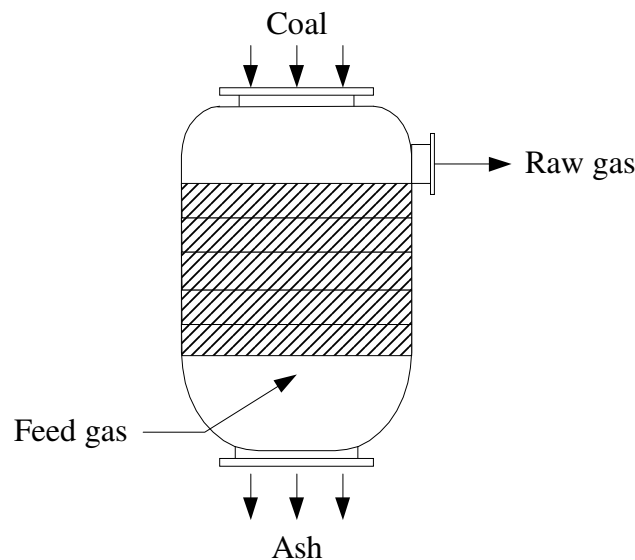
The main gaseous compounds produced are hydrogen sulphide ( $H_2S$ ) and ammonia ( $NH_3$ ) that are produced from the elemental sulfur and nitrogen found in the coal.

### A.3.5 Pre-heating zone

The coal comes in contact with the warm gases that are produced and all the moisture is driven off. The dry coal is then heated to 200 °C (de Ponte et al., 2001).

## A.4 Model description

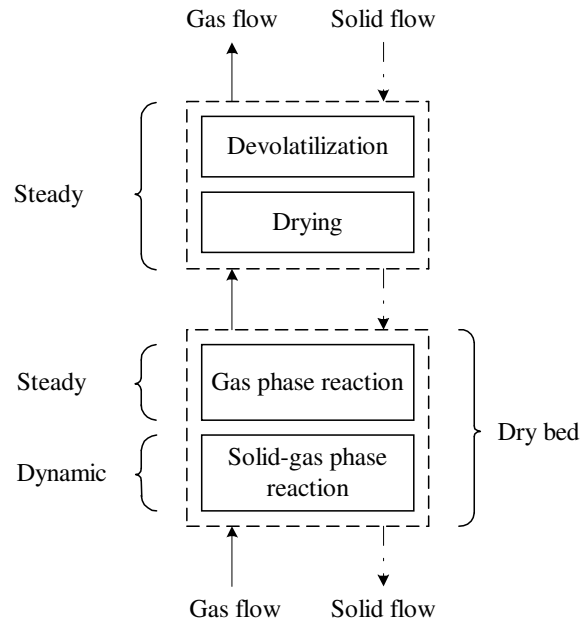
The input and output flow definitions used in the derivation of the non-linear time dependent model of a gasification unit can be seen in figure A.3.



**Figure A.3:** Simplified model of a gasification unit

A set of dynamic and steady-state components was defined that describes the interaction between the different reaction, mass and energy balances (figure A.4) and the mathematical development thereof is discussed in the following sections.





**Figure A.4:** Identified mathematical components of the gasification model

## A.5 Solid–gas phase reaction component

The component (figure A.4) is used to describe the time dependent non-linear gasification reactions that occur inside the coal particle. The shrinking core model together with component and energy balances are used to describe the different rate expressions for the reactions. The reaction temperature and composition of the section outlet flow can be calculated using the model.

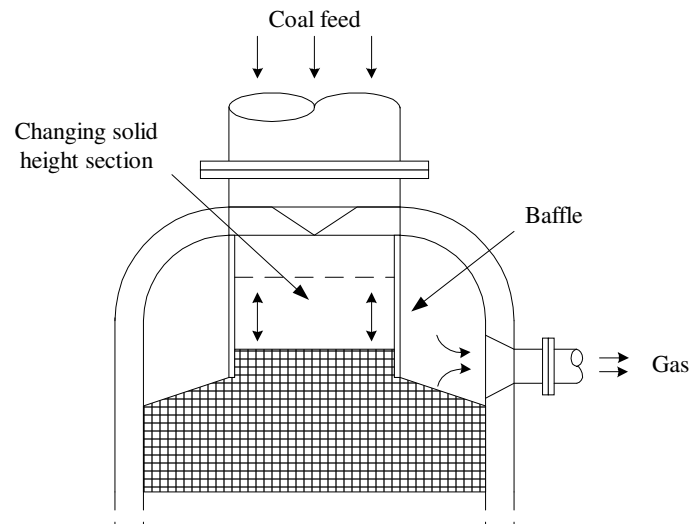
### A.5.1 Assumptions

#### Solid height

A change in the height of the solid material does not affect the path of the reaction gases through the packed bed, when the gasifier is correctly loaded. This is because a baffle divides the top of the gasifier into two annular sections (see figure A.5). The solid height will therefore change inside the baffle while the reaction gas exit on the outside, unaffected by the change in height. The time dependence of solid height was ignored in all the time dependent solid phase material balances (i.e.  $\frac{dh_{solid}}{dt} = 0$ ).

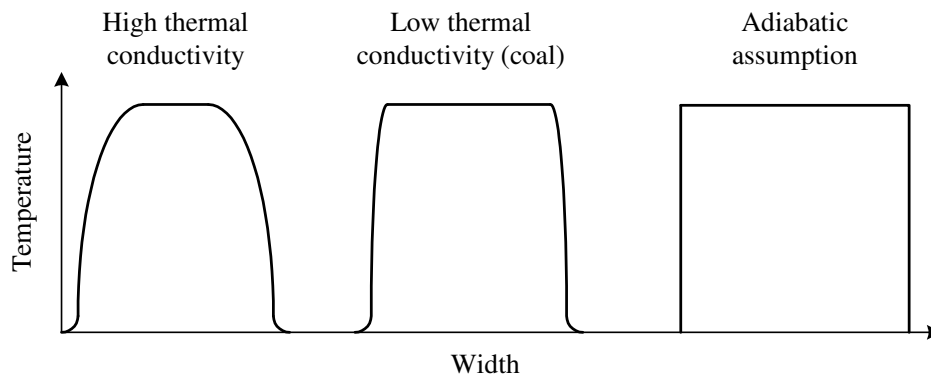
#### Radial temperature profile

The radial thermal conductivity of the coal inside the reaction chamber is very low (Yoon et al., 1978) as can be seen in figure A.6. The radial temperature distribution profile of the packed bed can be modelled by dividing the the bed into two annular sections; an adiabatic core and a small boundary layer. The boundary layer of the coal is however very small and is estimated



**Figure A.5:** Coal feed mechanism with a baffle at the top

to be 10 mm thick (Yoon et al., 1978). The boundary layer and heat loss to the mantle were therefore neglected as their influence is small (Yoon et al., 1978).



**Figure A.6:** Radial temperature distribution assumption

### Solid and gas phase temperatures

It was assumed that heat transfer between the solid and gas phases is high as the reactions take place inside the solid particle. It was therefore assumed that the temperature difference between the solid and gas phases is negligible (Yoon et al., 1978) for the gas–solid phase reactions.

### Gas phase flow dynamics

The gas phase residence time is in the order of seconds while that of the solid phase is hours (Yoon et al., 1978). The gas phase hold-up can be assumed to be zero as the gas phase dynamics approach steady state in relation to that of the slower solid phase dynamics. The inlet and outlet volumetric flow rates of the gas were therefore assumed to be equal and constant.

### Solid phase flow

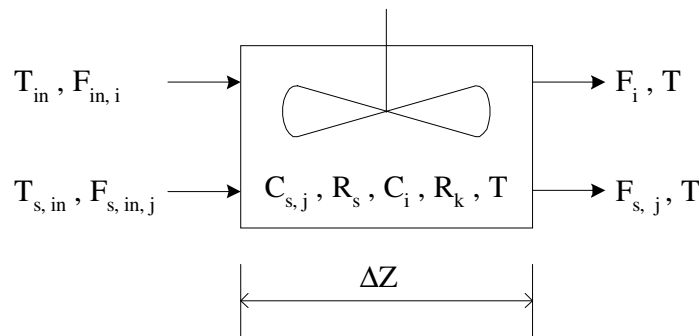
The density of a particle changes as the reaction occurs. It was assumed with the use of the shrinking core model (solid–gas phase reaction model) that the volume of such a particle stays constant. The volumetric flow of the packed bed was therefore assumed constant as it moved downwards through the reaction chamber.

### Continuous stirred tank reactor

It is assumed that the dynamic behavior of a predefined volume of the packed bed can be represented by a continuously stirred tank reactor (CSTR). The composition and temperature profiles inside the volume are therefore the same throughout the volume. It follows that the compositions and temperature of the well-mixed volume and outlet flows are also the same.

## A.5.2 Time dependent gas–solid phase description

The non-linear differential equations for a well-mixed section of the packed bed can be developed incorporating the assumptions made for the gas–solid phase reactions. The system used for the development of the set of non-linear differential equations can be seen in figure A.7. The description of the differential equations follows.



**Figure A.7:** Time dependent gas–solid phase description

### Component balances

The component balance description, for  $i$  gas phase components, is required to determine the product gas phase concentration ( $C_i$ ) as a function of the inlet flow ( $F_{i,in}$ ) and reaction ( $\dot{R}_i$ ) taking place:

$$\epsilon A \Delta Z \frac{dC_i}{dt} = F_{in,i} - F_i + (1 - \epsilon) A \Delta Z \dot{R}_i \quad (\text{A.6})$$

The outlet flow rate ( $F_i$ ) can be described as a function of concentration with the use of the specified volumetric flow rate ( $G$ ):

$$F_i = G C_i \quad (\text{A.7})$$

The solid phase component balance (for  $j$  components) determines the solid concentration ( $C_{s,j}$ ):

$$(1 - \epsilon)A\Delta Z \frac{dC_{s,j}}{dt} = F_{s,in,j} - F_{s,j} + (1 - \epsilon)A\Delta Z \dot{R}_s \quad (\text{A.8})$$

and the outlet flow rate ( $F_{s,j}$ ) is determined from the specified solid volumetric flow rate ( $G_s$ ):

$$F_{s,j} = G_s C_{s,j} \quad (\text{A.9})$$

### Energy balance

Making use of the assumption that temperature can be assumed to be uniform in the chosen volume the energy balance for the gas–solid phase reactions is used to determine the reaction temperature ( $T$ ) of the mixture, given the inlet temperatures of the gas ( $T_{in}$ ) and solid phases ( $T_{s,in}$ ):

$$Q_{in} = \sum_{i=1} F_{in,i} C_{p,i} (T_{in} - 298) + \sum_{j=1} F_{s,in,j} C_{p,s,j} (T_{s,in} - 298) \quad (\text{A.10})$$

$$Q_{out} = \sum_{i=1} F_i C_{p,i} (T - 298) + \sum_{j=1} F_{s,j} C_{p,s,j} (T - 298) \quad (\text{A.11})$$

$$Q_{reaction} = (1 - \epsilon)A\Delta Z \sum_{k=1} (-\Delta \dot{H}_k) \dot{R}_k \quad (\text{A.12})$$

$$\therefore \epsilon A\Delta Z \frac{d(C_i C_{p,i} T)}{dt} + (1 - \epsilon)A\Delta Z \frac{d(C_{s,j} C_{p,s,j} T)}{dt} = Q_{in} + Q_{reaction} - Q_{out} \quad (\text{A.13})$$

### Reaction rate equations

The kinetic rates of the reactions follow the Arrhenius relationship to temperature (i.e.  $k'' = k_0'' e^{-\frac{E}{RT}}$  (see table A.1) and is a function of the difference of the partial ( $P_i$ ) and equilibrium pressure ( $P_i^*$ ) of the gas phase reactant as well as the fraction of fixed carbon in the solid phase ( $x_s$ ) (Yoon et al., 1978).

$$r_k = k_{eff,i} (P_i - P_i^*) x_s \quad (\text{A.14})$$

It was assumed that the equilibrium pressure is close to zero (i.e.  $P_i^* = 0$ ). The reaction

rates for the equations are reduced to:

$$r_1 = k_{eff,1} P_{O_2} x_s \quad (A.15)$$

$$r_2 = k_{eff,2} P_{CO_2} x_s \quad (A.16)$$

$$r_3 = k_{eff,3} P_{H_2O} x_s \quad (A.17)$$

$$r_4 = k_{eff,4} P_{H_2}^2 x_s \quad (A.18)$$

The partial pressures ( $P_i$ ) of the different species can then be written in terms of the concentrations ( $C_i$ ) with the assumption that the gases are ideal:

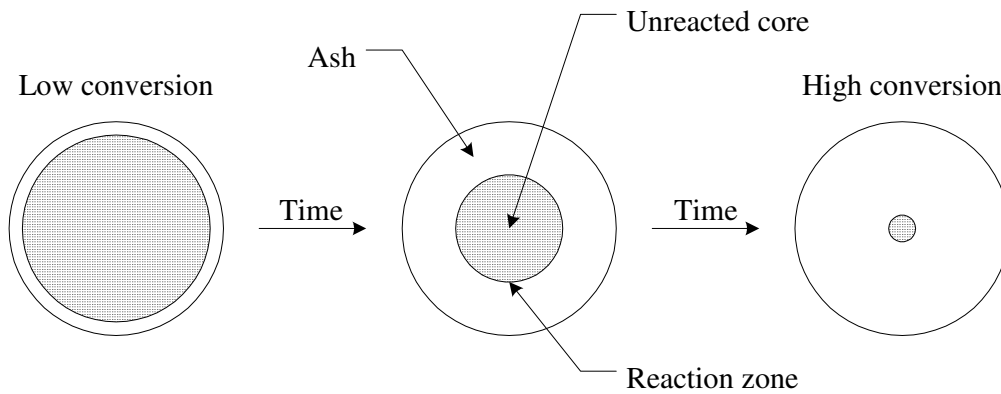
$$P_i = C_i RT \quad (A.19)$$

**Table A.1:** Reaction parameters (Yoon et al., 1978)

Reaction	$k_0 \frac{kmol}{kmol_C \cdot kPa \cdot min}$	$E \frac{kJ}{kmol}$
$C + O_2 \rightarrow CO_2$	1.08E6	113049
$C + H_2O \rightarrow CO + H_2$	6.48E5	146545
$C + CO_2 \rightarrow CO + H_2O$	2.46E2	146545
$C + 2H_2 \rightarrow CH_4$	5.04E-6	67201

The shrinking core model is used to describe solid–gas phase reactions. It can therefore be used to model the heterogeneous combustion reaction in the gasifier (Monazam & Shadle, 1998; Ludwig et al., 1985).

The reaction process can be visualized as a core of fresh material that shrinks as the reaction between the gas and solid takes place on the core surface and the ash is left behind. This can be seen in figure A.8.



**Figure A.8:** Shrinking core methodology

The reaction kinetics can be developed according to the rate controlling steps. Each of the controlling steps that can be identified will subsequently be discussed.

### Reaction rate controlling

The kinetics of all the heterogeneous reactions can be modelled by a first order rate equation with an Arrhenius relationship for the specific reaction rate (Hochgesand, 1989). It is assumed that the reaction rate is independent of pressure, because the diffusion effects will have a larger influence through the change of pressure than that of the reaction at the particle surface.

The reaction rate controlling constant, for reaction  $k$ , can be written (Yoon et al., 1978):

$$k_{react,k} = k_{0,k} e^{-\frac{E_k}{RT}} \quad (\text{A.20})$$

### Diffusion through the ash layer

The mass transfer coefficient ( $k_{diff}$ ) can be obtained as a function of the Reynolds (Re) and Schmidt (Sc) numbers through the Frössling correlation (Fogler, 1992):

$$\begin{aligned} Sh &= 2 + 0.6Re^{\frac{1}{2}}Sc^{\frac{1}{3}} \\ \therefore k_g &= \frac{D_{AB}}{d_p} \left( 2 + 0.6Re^{\frac{1}{2}}Sc^{\frac{1}{3}} \right) \end{aligned} \quad (\text{A.21})$$

It can be assumed that the Schmidt number is one ( $Sc = 1$ ) for gases (Levenspiel, 1999) and the diffusion rate constant can therefore be written:

$$k_{diff} = \frac{D_e}{D_p} (2 + 0.6Re^{0.5}) \quad (\text{A.22})$$

### Ash diffusion rate constant

The ash diffusion rate constant can be written as a function of the diffusion rate constant (Mon-azam & Shadle, 1998):

$$k_{ash} = \epsilon^{2.5} k_{diff} \quad (\text{A.23})$$

### Effective reaction rate modelling

This can be combined to give the effective reaction rate that can be incorporated into the four reaction rate equations, reaction A.15 to A.18 (Levenspiel, 1999):

$$k_{eff,i} = \frac{1}{\frac{0.5}{k_{ash}} + \frac{1}{k_{diff}} + \frac{3}{k_{react,i}}} \quad (\text{A.24})$$

The reaction rates of the different components can be developed according to the stoichiometric coefficients for the different reactions:

$$R_i = \sum_{k=1} \nu_i r_k \quad (\text{A.25})$$

## A.6 Gas phase reaction component

The component calculates the steady state equilibrium conversion of the reaction gas and determines the heat released to the gas due to the exothermic water–gas–shift reaction.

### A.6.1 Assumptions

#### Equilibrium reaction

The reaction occurs in the gas phase and is catalyzed by the coal particles (Yoon et al., 1978). It can therefore be assumed that the reaction is in equilibrium in relation to the slower combustion and gasification reactions that occur inside the particle (Monazam & Shadle, 1998).

#### Equilibrium constant calculation

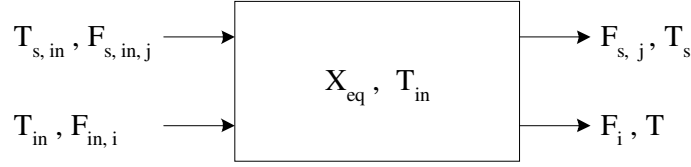
The equilibrium constant of the water–gas shift reaction is a function of temperature and should be evaluated at the reaction temperature (exothermic reaction). It was however assumed that the error introduced would be small if the constant was determined with the inlet temperature instead of the reaction or outlet temperature. This is because the water–gas shift reaction is only mildly exothermic and the temperature difference brought about by the reaction will therefore be small.

#### Solid and gas phase temperatures

It was assumed that the solid and gas phase temperatures are the same. This is consistent with the assumption made for the modelling of the gas-solid phase reactions.

### A.6.2 Steady state equilibrium calculation

The equations describing the water–gas shift reaction were developed for a steady state equilibrium reactor (figure A.9).



**Figure A.9:** Equilibrium reactor description

### Equilibrium reaction calculation

The relationship between the conversion ( $X_{eq}$ ) and the different components involved with the equilibrium constant is: (Kosky & Floess, 1980):

$$K_{eq}(T) = \frac{F_{eq,CO_2} F_{eq,H_2}}{F_{eq,CO} F_{eq,H_2O}} \quad (A.26)$$

$$\therefore K_{eq}(T) = \frac{(F_{CO_2} - X_{eq} F_{CO})(F_{H_2} - X_{eq} F_{CO})}{(F_{CO} - X_{eq} F_{CO})(F_{H_2O} - X_{eq} F_{CO})} \quad (A.27)$$

The equilibrium constant  $K_{eq}$  can be calculated as a function of the inlet temperature with the use of the Arrhenius relationship (Kosky & Floess, 1980):

$$K_{eq}(T_{in}) = 0.0265e^{\frac{-7860}{RT_{in}}} \quad (A.28)$$

### Component balance

The steady state ( $\frac{dC_i}{dt} = 0$ ) gas phase component balance is:

$$F_{i,in} - F_i + \nu_i X_{eq} F_{CO} = 0 \quad (A.29)$$

The outlet gas flow rates ( $F_i$ ) can accordingly be calculated given the equilibrium conversion ( $X_{eq}$ ). The inlet and outlet flow rates for the solid phase are the same (i.e.  $F_{s,j,in} = F_{s,j}$ ) as the reaction takes place in the gas phase only.

### Energy balance

The steady state energy balance for the gas-solid phase reaction can be used to determine the outlet reaction temperature ( $T$ ) if the equilibrium conversion ( $X_{eq}$ ) and outlet flows are known ( $F_{in}$  and  $F_{s,in}$ ):

$$\sum_{j=1} F_{s,j} C_{p,s,i}(T_{in} - T) + \sum_{i=1} F_i C_{p,i}(T_{in} - T) + (-\Delta H) X_{eq} F_{CO} = 0 \quad (A.30)$$



## A.7 Drying and devolatilization description

This is a steady-state description (energy balance) of the heat necessary to remove the moisture inside the coal. The heating of the char and devolatilized gases due to mixing with the warm reaction gases evolved at the bottom of the reactor is also incorporated in the same energy balance.

### A.7.1 Assumptions

#### Volatilization and drying reactions

The volatilization and drying of the coal to form char occurs rapidly in the upper part of the solid bed. It was assumed that volatilization and drying occurs instantly and completely (Yoon et al., 1978). The heats of reaction for the volatilization reactions are assumed to be zero (Kosky & Floess, 1980).

#### Water–gas shift reaction

The reaction gas is quenched with cold feed over a very short distance at the top of the reactor, thus terminating the water–gas shift equilibrium (Kosky & Floess, 1980). The water–gas shift equilibrium calculation was therefore not included.

#### Solid and gas temperatures

It was assumed that the exit solid and gas temperatures are the same.

### A.7.2 Volatiles and drying calculation

The calculation of the energy necessary to heat and the dry the coal, and the mixing of the volatile matter and reaction gas is incorporated into one steady state calculation (figure A.10).

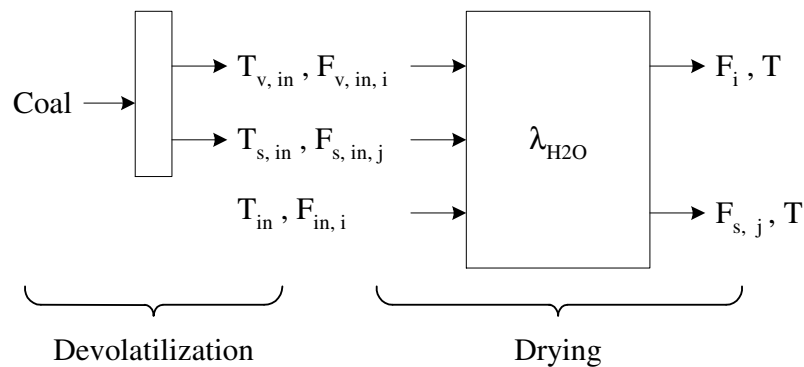


Figure A.10: Devolatilization and drying description

The input coal composition used to define the volatilization ( $F_{v,i,in}$ ) and solid flows ( $F_{s,j,in}$ ) can be seen in table A.2.

**Table A.2:** Inlet coal composition

Coal composition (%v\v)	(Hochgesand, 1989)
Volatile matter	14
Fixed carbon	56
Moisture	9
Ash	21
Coal composition (%v\v)	(Yoon et al., 1978)
CO	20.6
CO <sub>2</sub>	6.1
H <sub>2</sub>	13.1
CH <sub>4</sub>	50.3
Other	9.9

The energy balance, with the assumption that the  $C_p$  values for the different streams at the inlet and outlet temperature conditions are the same, is:

$$Q_{heat} = \lambda_{H_2O} F_{v,H_2O} \quad (A.31)$$

$$Q_{in} = F_{s,j,in} C_{p,s} (T_{s,in} - 298) + F_{v,i,in} C_{p,v} (T_{v,i,in} - 298) + F_{i,in} C_p (T_{in} - 298) \quad (A.32)$$

$$Q_{out} = (F_{s,j} C_{p,s} + F_{v,i} C_p + F_i C_p) (T - 298) \quad (A.33)$$

The energy required to vaporize the moisture in the coal ( $\lambda_{H_2O}$ ) is only incorporated when the outlet temperature is higher than 373 K. The total energy balance can be written:

if  $T > 373$  K:

$$Q_{out} = Q_{heat} + Q_{in} \quad (A.34)$$

if  $T < 373$  K:

$$Q_{out} = Q_{in} \quad (A.35)$$

## A.8 Gasifier model implementation

The SIMULINK implementation of the dynamic gasifier model used for the simulation of the startup and shutdown procedure can be seen in figure A.11. The supervisory controller implementation is situated on the workspace inside the "Grafcet" block.

The controllers used in the simulation of the gasification unit are static. This is because the dynamics of the feedback loops of the controllers are too fast (time constants of seconds)

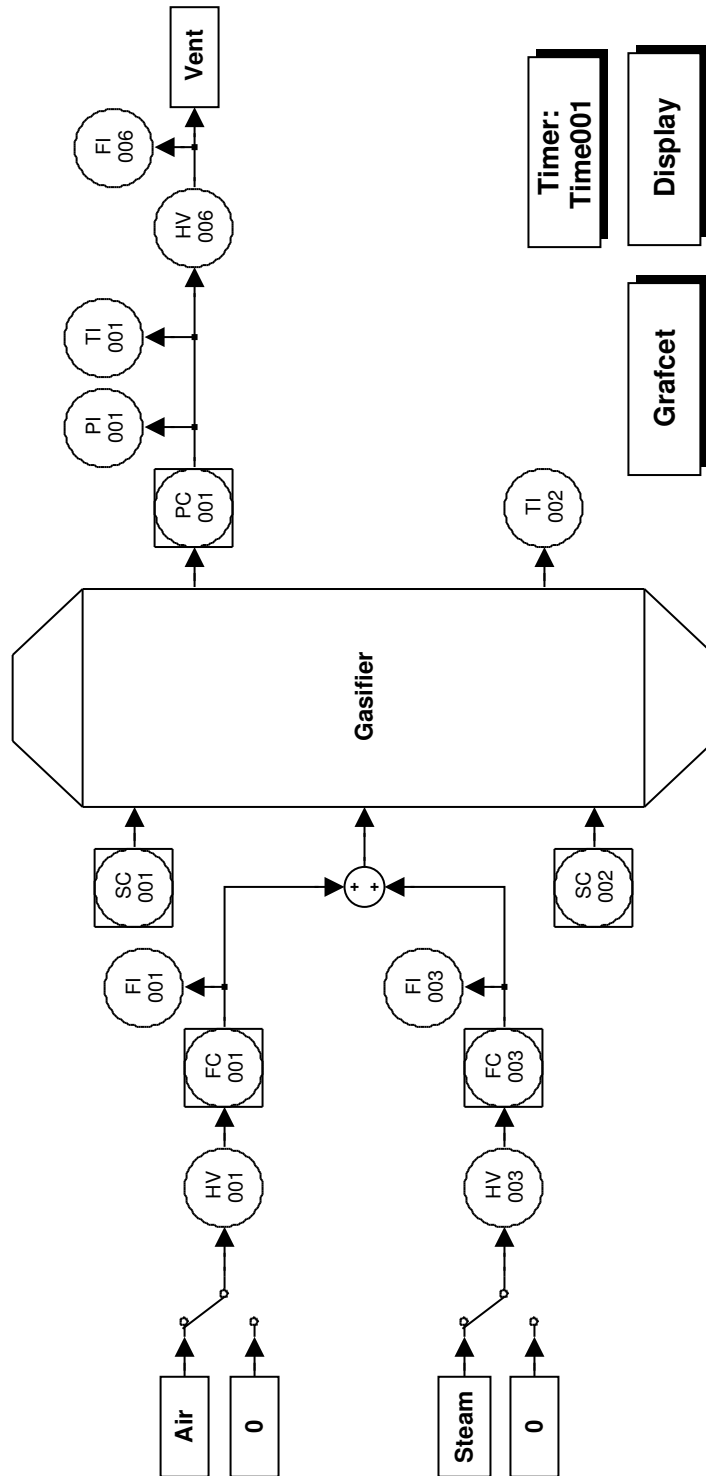


Figure A.11: Simulink implementation of the gasification unit

to influence the working of the supervisory control system that operates in time intervals of minutes. The standard feedback control loops were accordingly replaced with algebraic steady-state functions relating the controlled to the manipulated variables.

### Flow controller

The algorithm used to determine the output flow given the flow set point ( $F_{set}$ ) and input flow ( $F_{in}$ ) is:

$$F_{out} = x_{mode} \left( \frac{F_{set} - F_{min}}{F_{max} - F_{min}} \right) F_{in} \quad (A.36)$$

### Pressure control

The gas volumetric gas flow rate is a non-linear function of the pressure.

$$P = \frac{\sum F_i RT|_{steady}}{G_{gas}} \quad (A.37)$$

The non-linear controller algorithm used is:

$$G_{gas} = \frac{62829}{P} \quad (A.38)$$

### Emergency cut-off valve

The modelled emergency cut-off valves are used as a safety measure should the control valve fail and the fraction opening of the valve ( $x_{set}$ ) is specified to give the outlet flow rate ( $F_{in}$ ):

$$F_{out} = x_{set} F_{in} \quad (A.39)$$

---



---

# APPENDIX B

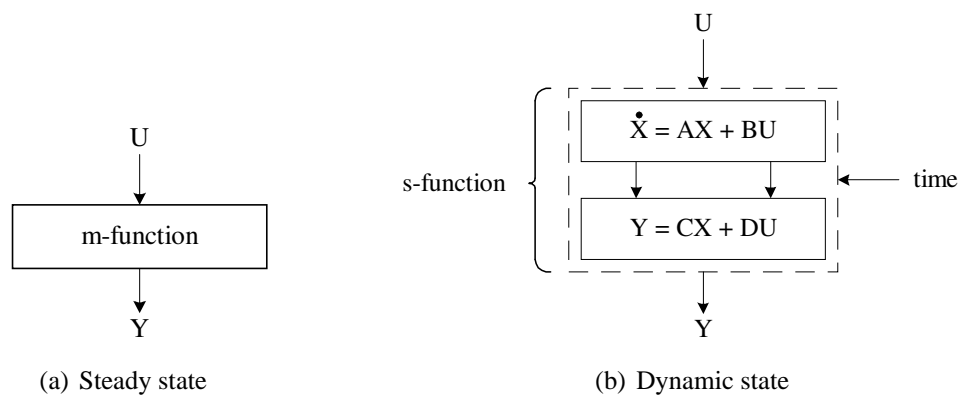
---

## Simulation of a gasification unit

### B.1 Implementation of the computer simulation

The model of the gasification unit was implemented in Simulink. The dynamic equations describing the gas–solid phase reactions were implemented in the form of a non-linear differential equations. The standard block function used to achieve this is a s–function and are used to generate the time dependent outputs given model inputs and initial states (figure B.1(b)).

The steady-state equations describing the water–gas shift reaction and devolatilization and drying of coal were implemented as static functions (m-functions) that generate outputs according to the given inputs for each integrated time step (figure B.1(a)).

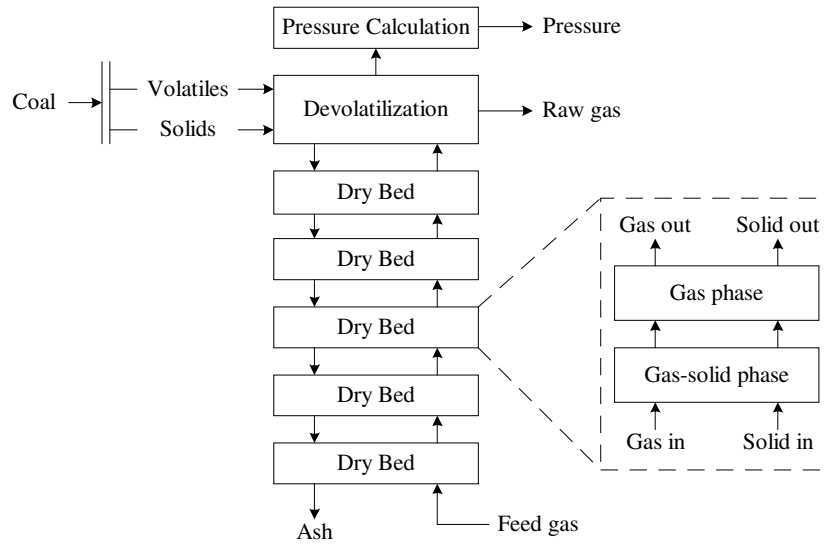


**Figure B.1:** Implementation of the dynamic and steady state equations

#### B.1.1 Dynamic bed model

The packed bed was divided into several distinct zones with the assumption that each one was well-mixed. The well-mixed zones were connected in a counter current way to simulate

upwards and downwards flow of the gas and solid phases (figure B.2).



**Figure B.2:** Simulated description of the gasification unit

The steady-state mathematical component that describes the devolatilization and drying of the coal as well as the pressure calculation block ( $P = \sum C_i RT$ ) was inserted at the top of the reactor. A generic repeating subsystem (dry bed) was defined to model the successive incremental elements of the packed bed inside the reaction chamber. The dry bed subsystem contains the dynamic solid-gas phase component (s-function) as well as the gas phase component (m-function) at different heights inside the bed.

## B.2 Model parameter definition

The model parameters defined for the model as operated with steam and oxygen (oxygen blown) and also with air and oxygen (air blown) can be seen in table B.1. The air blown feed stream specifications were taken as twenty percent of the oxygen blown gasifier simulation.

## B.3 Calculation of modelling data

### B.3.1 Heat of reaction

The heat of reaction can be determined from the *heat of formation* temperatures ( $\Delta H_{f,298}^\circ$ ) (Smith et al., 1996:640). The heat of reaction can then be determined at the relevant temperature ( $T$ ) by (Smith et al., 1996:135):

$$\Delta H_{r,T} = \Delta H_{r,298} + \int_{298}^T \Delta C_p dT \quad (\text{B.1})$$

**Table B.1:** Oxygen blown model parameters

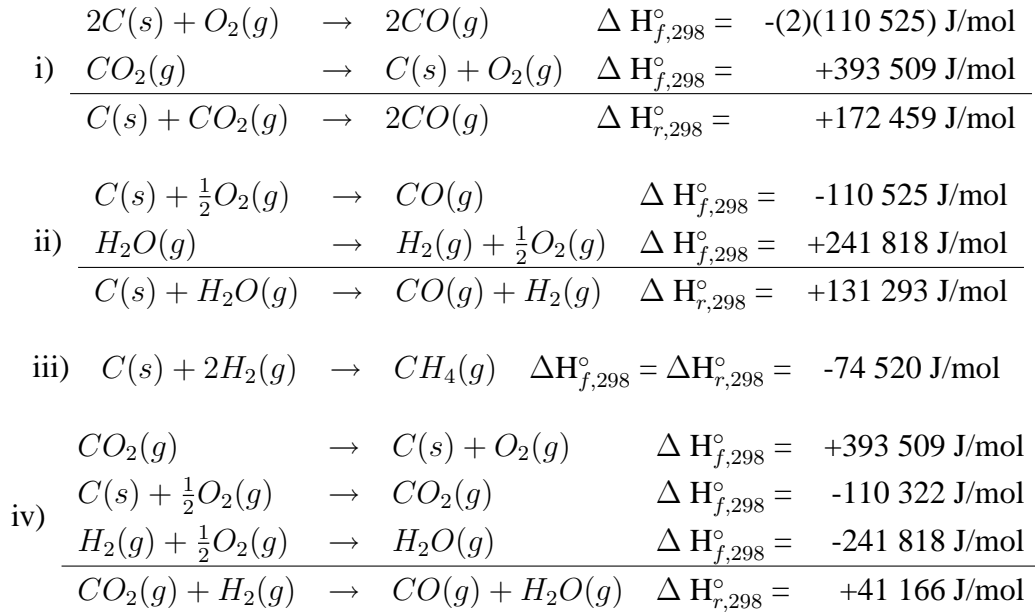
Parameter	Value	Units	Reference
<b>Operational specifications</b>			
<i>Oxygen blown</i>			
Oxygen ( $F_{O_2}$ )	6	kmol/min	(Yoon et al., 1978)
Steam ( $F_{H_2O}$ )	31	kmol/min	(Yoon et al., 1978)
Coal feed ( $F_{Char}$ )	24	kmol/min	
Pressure (P)	2.5	MPa	(Yoon et al., 1978)
<i>Air blown (20% of oxygen flow)</i>			
Oxygen ( $F_{O_2}$ )	0.61	kmol/min	
Nitrogen ( $F_{N_2}$ )	2.3	kmol/min	
Steam ( $F_{H_2O}$ )	4	kmol/min	
Coal feed ( $F_{Char}$ )	4.6	kmol/min	
Pressure (P)	0.5	MPa	
Gas temperature ( $T_{g,in}$ )	370	$^{\circ}C$	(Yoon et al., 1978)
Solid temperature ( $T_{s,in}$ )	25	$^{\circ}C$	
<b>Packed bed properties</b>			
Bed diameter ( $D$ )	3.66	m	(Yoon et al., 1978)
Bed height ( $\Delta Z$ )	3	m	(Yoon et al., 1978)
Density ( $C_{s,o}$ )	143	kmol/m <sup>3</sup>	(Perry & Green, 1997)
Particle diameter ( $D_s$ )	10	mm	(Yoon et al., 1978)
Voidage ( $\epsilon$ )	0.4		(Perry & Green, 1997)
<b>Gas phase properties</b>			
Viscosity ( $CO_2$ at 500 $^{\circ}C$ )	1.08E-4	Pa · min	(Perry & Green, 1997)
Diffusivity ( $D_{gas,coal}$ )	1E-3	m <sup>2</sup> /min	(Levenspiel, 1999)
<b>Initial conditions</b>			
Outlet temperature (T)	25	$^{\circ}C$	
Outlet gas flows ( $F_i$ )	0	kmol/min	
Gas phase concentration ( $C_i$ )	0	kmol/m <sup>3</sup>	
Outlet solid flows ( $F_{s,j}$ )	0	kmol/min	
Ash concentration ( $C_{s,ash}$ )	23	kmol/min	
Carbon concentration ( $C_{s,char}$ )	63	kmol/min	

$$\Delta C_p = \sum_i \gamma_i C_{pi} \quad (\text{B.2})$$

### Heat of combustion reaction



### Heat of gasification reactions



### B.3.2 Heat capacity

The heat capacity ( $\frac{C_p}{R} = A + BT + CT^2 + DT^{-2}$ ) as a function of temperature can be seen in table B.2. The assumption could not be made that the  $C_p$  values are independent of temperature (i.e.  $dH = C_p dT$ ) as the operating range of the temperature is too large. The data for the

**Table B.2:**  $C_p$  values as a function of temperature (Smith et al., 1996:638–639)

Chemical species	A	$10^3$ B	$10^6$ C	$10^{-5}$ D
C(s)	1.771	0.771		- 0.867
O <sub>2</sub> (g)	3.639	0.506		- 0.227
CO(g)	3.376	0.557		- 0.031
CO <sub>2</sub> (g)	5.457	1.045		- 1.157
H <sub>2</sub> O(g)	3.470	1.450		- 0.121
H <sub>2</sub> (g)	3.249	0.422		- 0.083
CH <sub>4</sub> (g)	1.702	9.081	- 2.164	

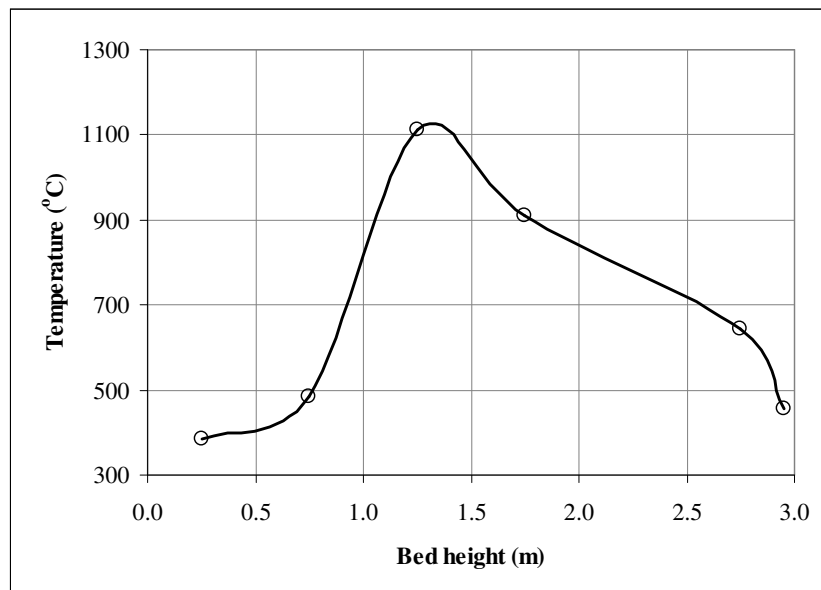


## B.4 Steady state profiles

The steady state profiles of the different simulations were obtained and was compared with steady-state models found in literature (Yoon et al., 1978; Hochgesand, 1989; Kosky & Floess, 1980).

### B.4.1 Oxygen blown simulation

The steady state temperature profile for the gas and solid phases can be seen in figure B.3 and the composition profile in figure B.4.

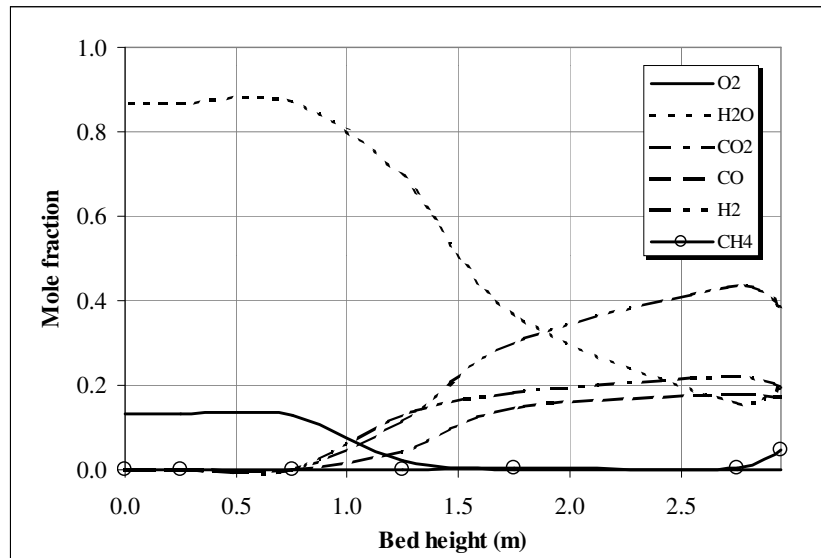


**Figure B.3:** Steady state temperature profile of the oxygen blown model (1000 minutes)

The effect of the ash layer (area of no reaction and low temperatures) at the bottom of the gasifier can clearly be seen in the temperature profile. The temperature profile also shows that the combustion zone is situated between the bed height of 0.5 and 1.5 m. This is confirmed by the drop of oxygen composition at the specified height.

The bed then gradually cools as the endothermic gasification reactions occur to produce hydrogen ( $H_2$ ), methane ( $CH_4$ ) and carbon monoxide ( $CO$ ). The cooling of the gas mixture together with the accompanying increase of the solid temperature at the top is due to the drying and devolatilization of the coal.

The comparison of the raw gas compositions of the model with that of literature can be seen in table B.3. It can be seen that the product compositions are predicted in the correct order (i.e.  $x_{H_2} > x_{CO_2}$ , etc.).



**Figure B.4:** Steady state composition profile of the oxygen blown model (1000 minutes)

**Table B.3:** Oxygen blown model results compared to plant data

Description	Plant	Model	(Yoon et al., 1978)
CO <sub>2</sub>	28	24	27
CO	22	21	22
H <sub>2</sub>	38	48	44
CH <sub>4</sub>	10	6	6
Other	2	1	1
Total	100	100	100

## B.4.2 Fire-bed movement

The position of the combustion zone (fire-bed) is a function of the amount of coal fed into the gasifier (Yoon et al., 1978). The steady state temperature profiles of the coal feed at 336 kg/min and 360 kg/min can be seen in figure B.5. The figures show that the bottom temperature increases while the top temperature decreases as the fire-bed moves downward.

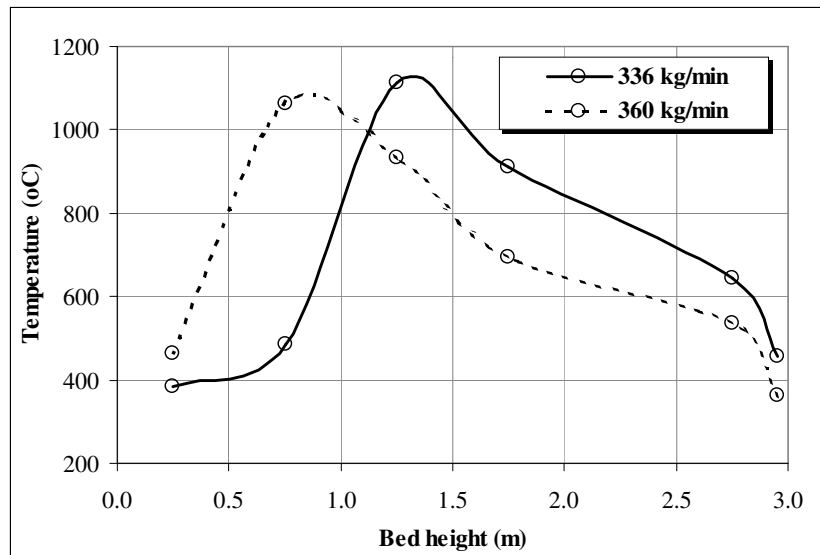


Figure B.5: Steady state temperature gas phase profile (1000 minutes)

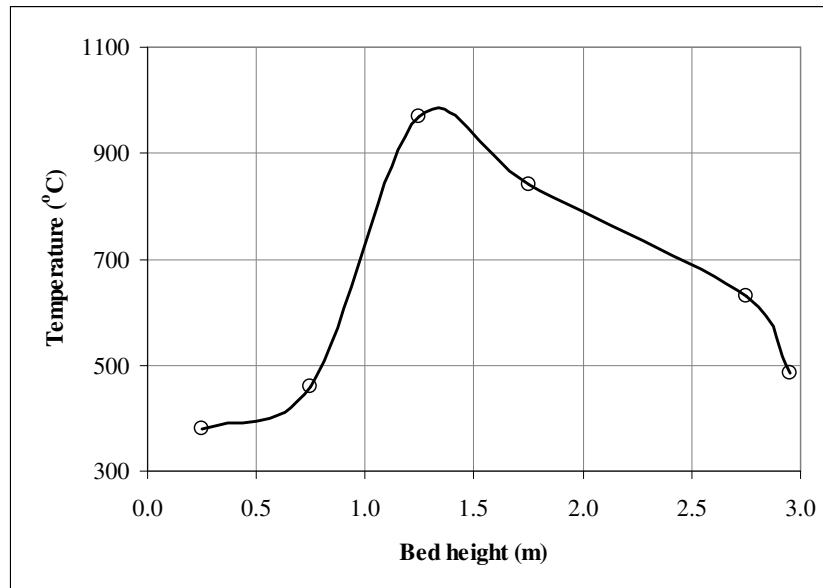
## B.4.3 Air blown simulation

The temperature profile for the air blown gasifier model can be seen in figure B.6. The figure shows that the maximum temperature is lower and less well-defined than that for the oxygen blown gasifier model. The rate of reaction is therefore lower even though the oxygen to steam ratio is the same as that for the oxygen blown gasifier. This is due to the nitrogen introduced with the air that cools the reaction down. The oxygen therefore takes longer to be consumed resulting in a larger fire-bed and a less well-defined maximum temperature.

## B.5 Dynamic model description

### B.5.1 Initial conditions

The initial conditions for the dynamic simulation is listed in table B.1. The dynamic simulation started with a full bed of coal in the reaction chamber at a temperature of 25 °C with no inlet and outlet flows.

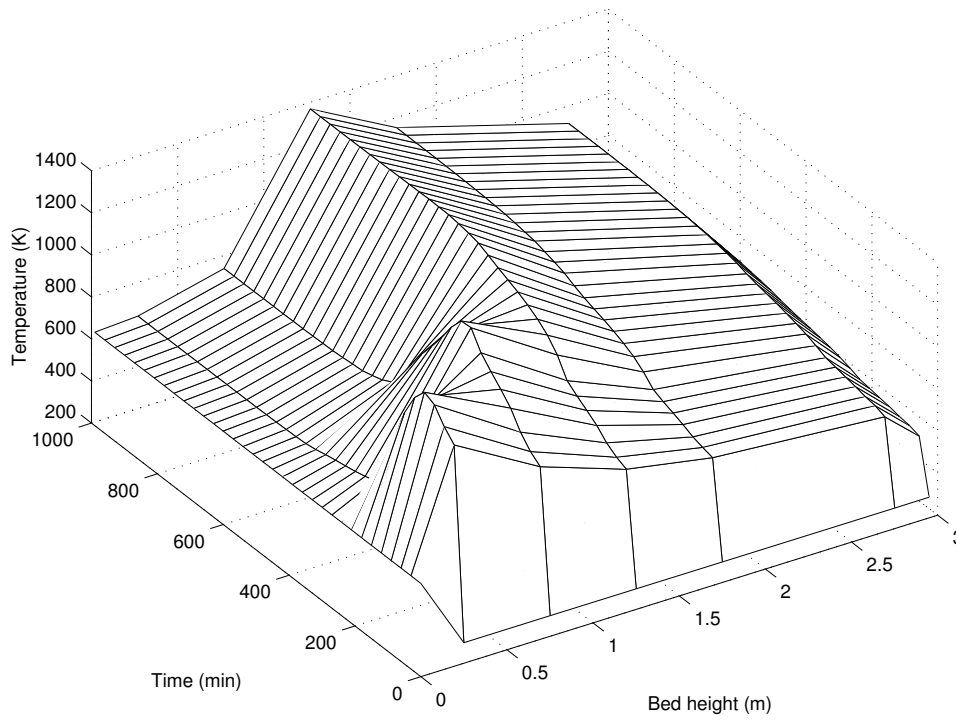


**Figure B.6:** Steady state temperature profile of the air blown model (2000 minutes)

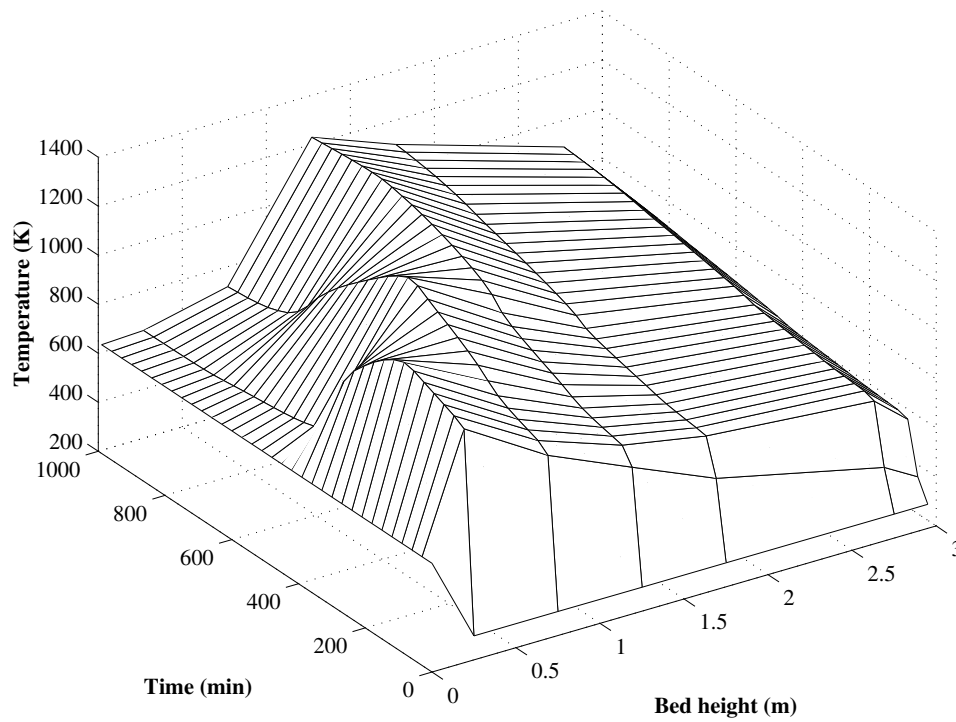
### B.5.2 Dynamic profiles

The dynamic temperature profiles for an oxygen and a air blown gasification unit can be seen in figures B.7 and B.8.

The development and stabilization of the ash layer can clearly be seen as the fire-bed moves upwards in the packed bed. The fire-bed takes a longer time to develop and is less well-defined for the air blown gasifier. This is due to the lower rate of reaction, as discussed earlier.



**Figure B.7:** Dynamic temperature profile of the oxygen blown gasifier



**Figure B.8:** Dynamic temperature profile of the air blown gasifier

---



---

# APPENDIX C

---

## Software description

### C.1 Gasifier modelling

The software dependencies of the different m-functions, s-functions and simulink blocks can be seen in figure C.1.

#### C.1.1 m-functions

The different m-functions defined as well as their descriptions are listed in table C.1. More information can be obtained by typing “help function name.m” in the main MATLAB window.

**Table C.1:** m-file description

Function name	Description
CpAtTemp	Determines the $C_p$ value at the given temperature.
EquilibriumReaction	Obtains the equilibrium concentrations of a mixture.
FlowSum	Determines the mixing flow of the de-volalization and gas streams. The heat removed due to drying is also included.
GasifDerivatives	Calculates the differentials (i.e. $\dot{x} = Ax + Bu$ ) of GASIFMODEL.
GasifOutputs	Generates the outputs (i.e. $y = Cx + Du$ ) for GASIFMODEL.
GasifReaction	Generates the reaction rates for the different components for GASIFMODEL.
WaterGasShiftReaction	Calculates the equilibrium concentrations of the water-gas shift reaction.



Figure C.1: Function dependencies

### C.1.2 s-functions

The two s-functions defined as well as their descriptions are listed in table C.2. More information can be obtained by typing “help function name.m” in the main MATLAB window.

**Table C.2:** s-function descriptions

Function name	Description
GasifModel	Calculates the dynamic response of the gas–solid phase reactions.
GasifPress	Calculates the dynamic response of the pressure model.

### C.1.3 Simulink implementation

The implementation of the different m-functions and s-functions discussed in the previous section can be seen in figure C.2.

The inputs and outputs are defined in table C.3. The dynamic model was divided into two sub-systems, the first (dry bed) is used to define the gas and gas–solid reactions in the reactor and is connected in counter currently in order to simulate the counter current flow of the gas and solid phases. The second sub-system defined (de-volatilization) is used to describe the de-volatilization and drying of the coal at the top of the reactor.

**Table C.3:** Input and output block definitions

Name	Definition	Type
<i>Inputs</i>		
Gas: composition in	Molar flow of the different gas phase components into the reactor. The component description is: $\mathbf{u} = [O_2, H_2O, CO_2, CO, H_2, CH_4, N_2]$	Vector
Gas: Flow in	Volumetric ( $m^3/min$ ) flow rate of the gas through the bed.	Scalar
Gas: Temperature	Inlet gas phase temperature.	Scalar
Coal: Total in	Molar coal flow into the reactor.	Scalar
Solid: Flow in	Volumetric solid flow rate ( $m^3/min$ ) through the reactor.	Scalar
<i>Outputs</i>		
Gas: Total flow out	Exit gas flow. The the vector description is: $\mathbf{u} = [O_2, H_2O, CO_2, CO, H_2, CH_4, N_2, G_{gas}, T_{gas}, P]$	Vector
Solid: Total flow out	Solid exit flow with the vector description: $\mathbf{u} = [Char, Ash, G_{solid}T_{solid}]$	Vector

#### Dry bed sub-system

The dry bed sub-system is described in figure C.3. The and models the gas–solid (rate limited) and gas (WGS) phase reactions. The gas–solid phase reactions are calculated with the



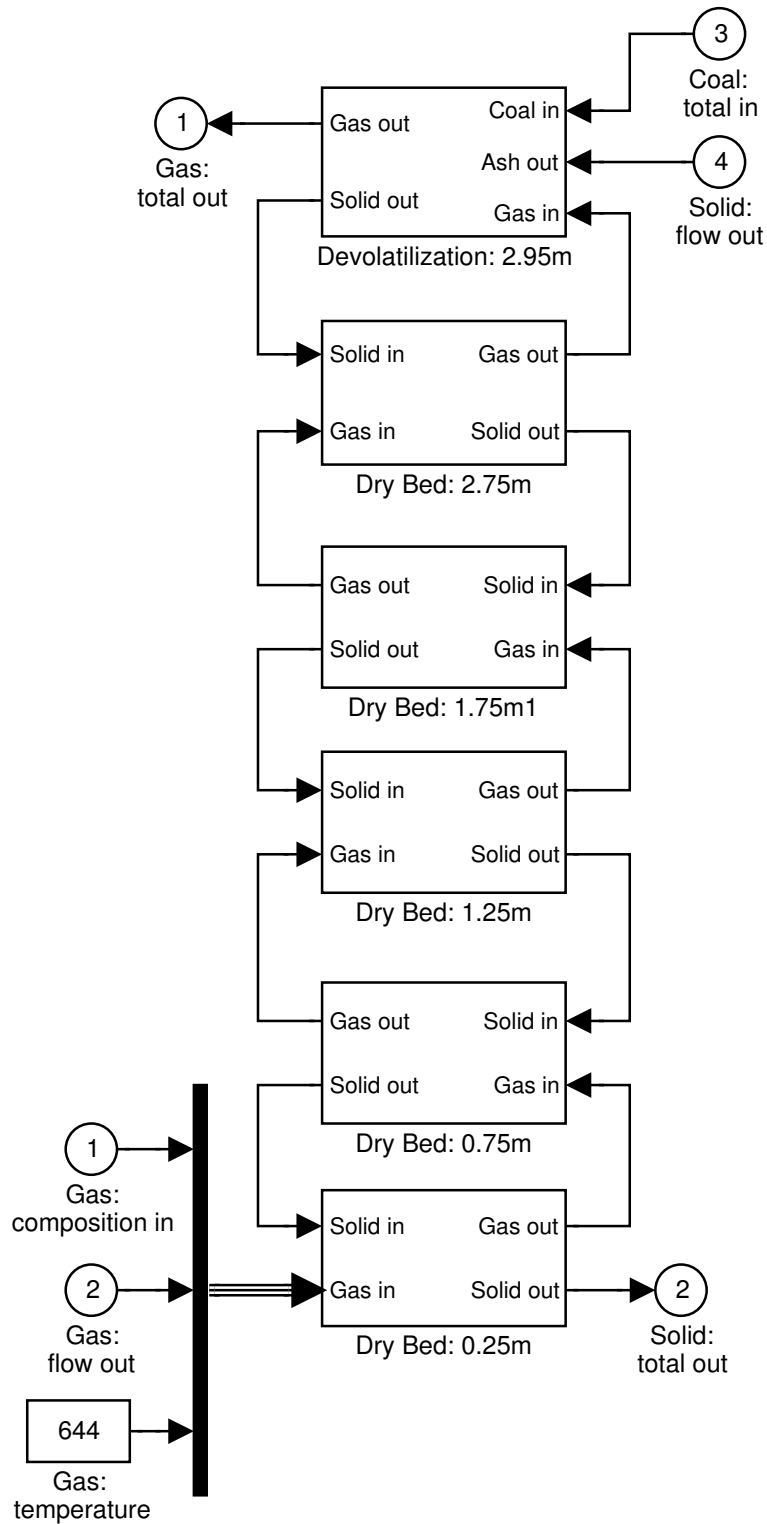
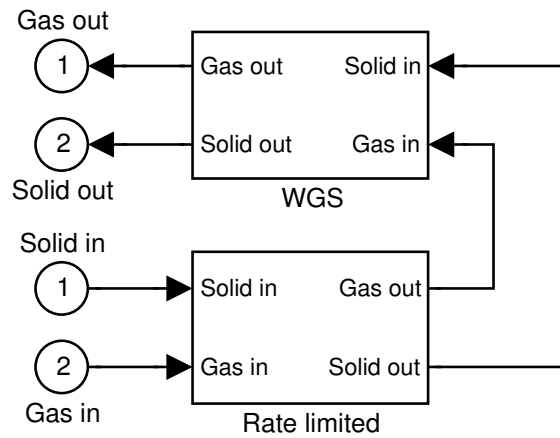


Figure C.2: Simulink dynamic bed implementation

GASIFMODEL s–function while the gas phase reactions are described with the WATER-GASSHIFTREACTION m–function.



**Figure C.3:** Simulink dry bed sub–system

The input and output definitions of the different flows are the same and is defined for the gas flows ( $O_2$  is the molar oxygen flow rate etc.):

$$\mathbf{u}_{gas} = [O_2, H_2O, CO_2, CO, H_2, CH_4, N_2, G_{gas}, T_{gas}]$$

and for the solid flows:

$$\mathbf{u}_{solid} = [Char, Ash, G_{solid}, T_{solid}]$$

### De-volatilization sub–system

The block definition used to describe the de–volatilization subsystem can be seen in figure C.4. The figure shows how the three different components interact. The different components can be listed:

- i) The coal composition block divides the incoming coal into volatile and solid components.
- ii) The flowsum block is used to model the de–volatilization and heating of the coal at the top of the bed.
- iii) The pressure block calculates the pressure of the top part of the gasifier using the ideal gas law.

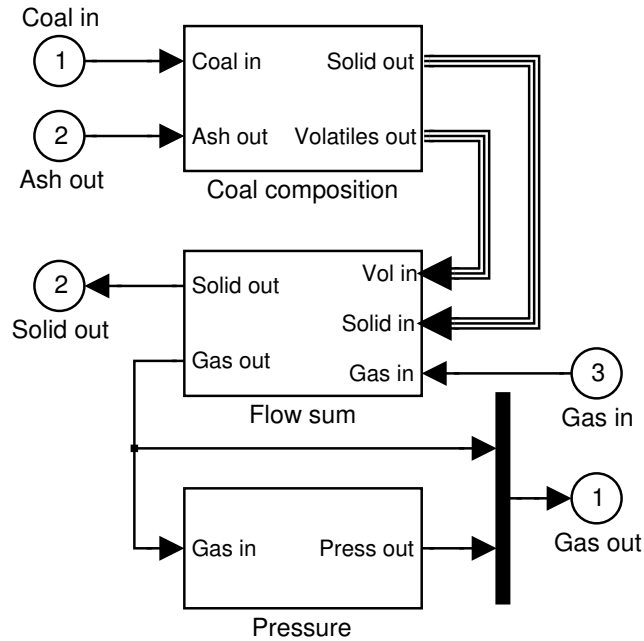


Figure C.4: Simulink de-volatilization sub-system

## C.2 Grafcet

### C.2.1 Overview

The dependencies of the Simulink blocks, s-functions, m-functions and graphical unit interfaces (GUI) are shown in figure C.5.

### C.2.2 m-function

The different m-functions defined as well as their descriptions are listed in table C.2.2. More information can be obtained by typing “help function name.m” in the main MATLAB window.

### C.2.3 s-function

The s-functions defined and their descriptions are listed in table C.2.3. More information can be obtained by typing “help function name.m” in the main MATLAB window.

### C.2.4 Algorithms

The different algorithms defined for the generic Grafcet components are listed below as well as the recursive rSrcSearch function that is used by the *convergence bar* to determine the *steps* associated with this operation.

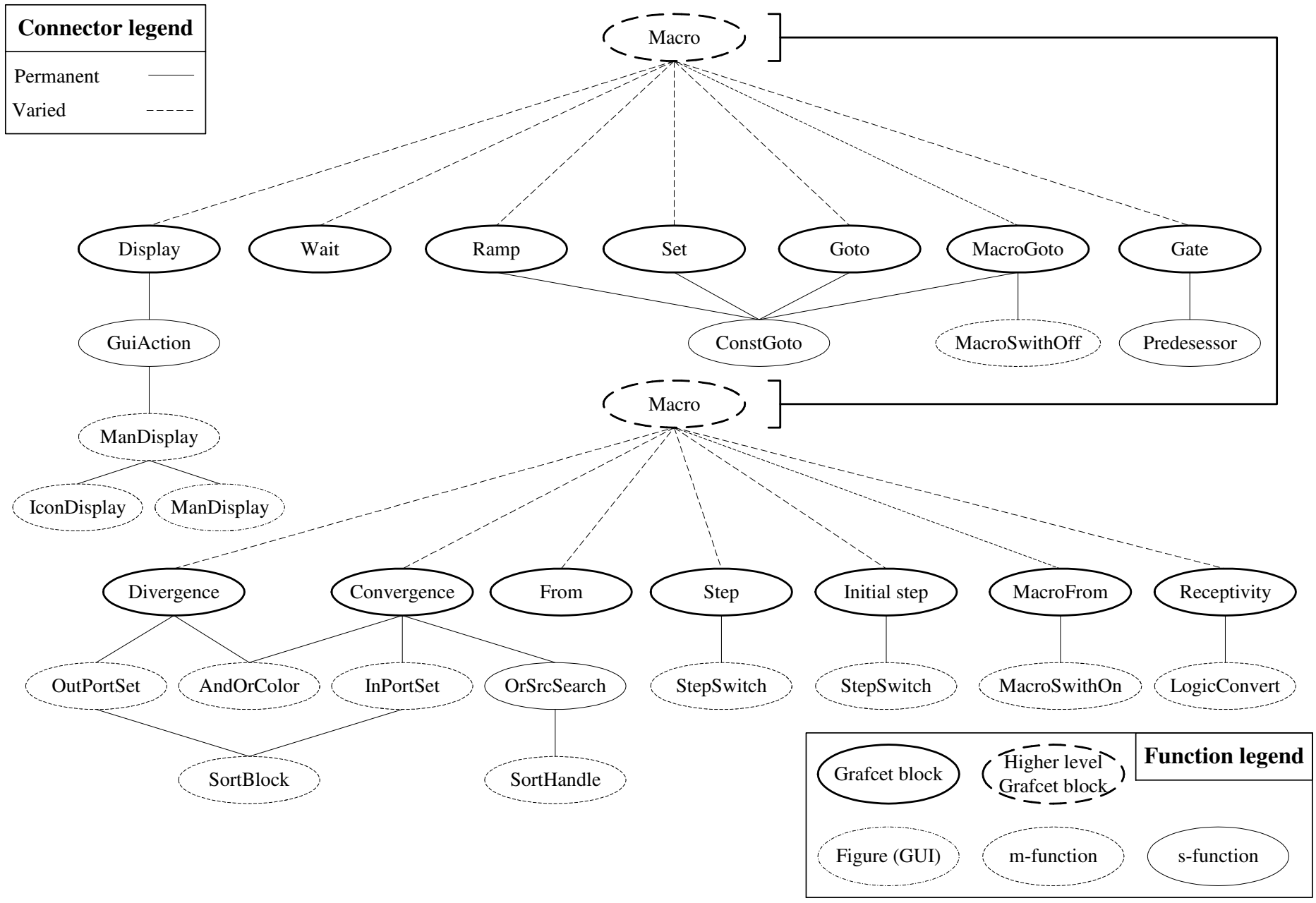


Figure C.5: Function and simulink block description

Function name	Description
AndOrColor	Changes the background color of the converge and diverge bars.
IconDisplay	Adds an icon to the current figure.
InPortSet	Sets the specified amount of input ports for the convergence block.
OutPortSet	Sets the specified amount of output ports for the divergence block.
LogicConvert	Outputs a logic operator given a numeric value.
ManDisplay	Executes the ManDisplay.fig graphical unit interface (GUI).
SortBlock	Sort the different Simulink blocks of a sub-system according to their type.
SortHandle	Finds and sorts the handles of a Simulink block in a sub-system.

Function name	Description
MacroSwitchOn	Changes the macro block (current system) background color and re-initialise all the steps.
MacroSwitchOff	Changes the macro colour.
StepSwitch	Changes the step block (current system) background color and reset the switch to reset the step.
ConstGoto	Custom goto block function, sets the value of the corresponding from block.
GuiAction	Calls the specified GUI when activated.
OrSrcSearch	Switches all the predecessor blocks from an convergence block.
Predecessor	Switches the predecessor block.

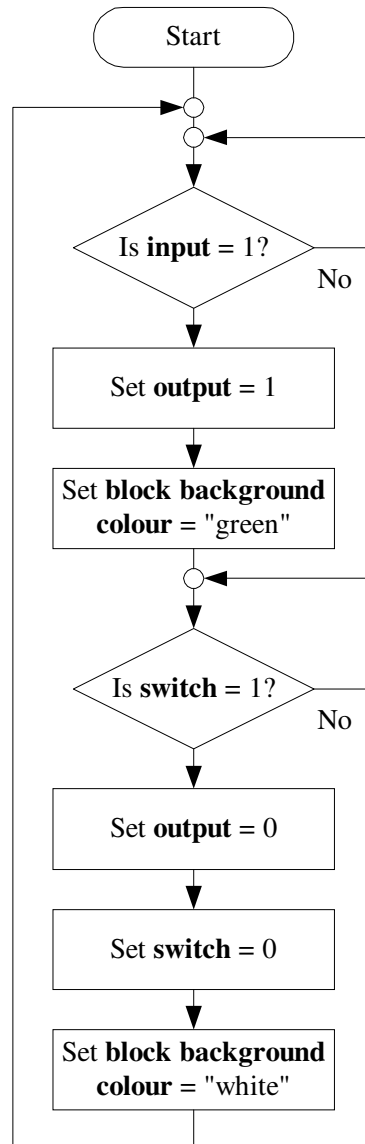


Figure C.6: Step algorithm

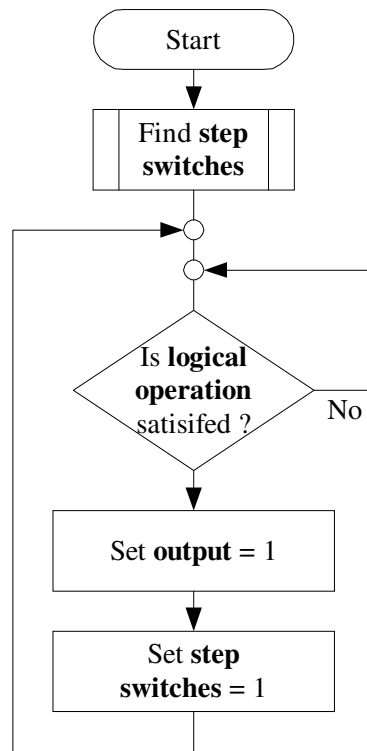


Figure C.7: Convolution bar algorithm

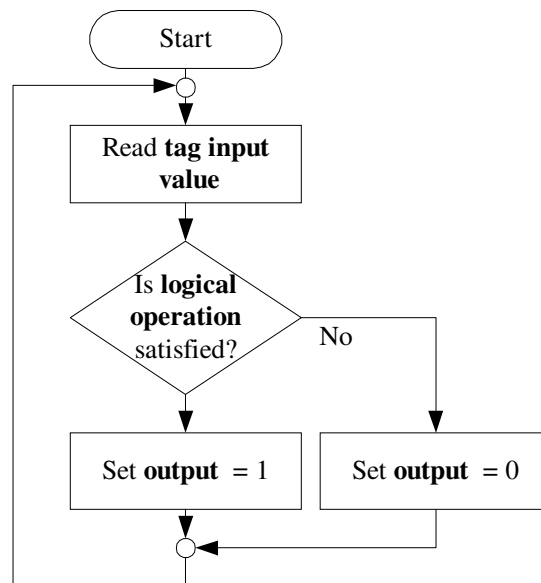


Figure C.8: Receptivity algorithm

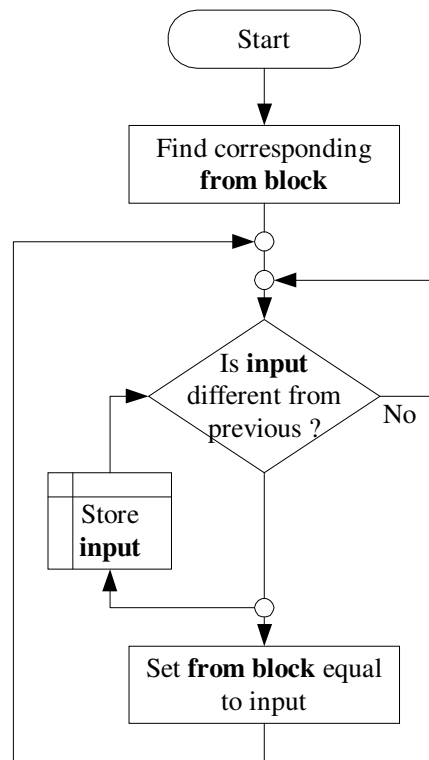


Figure C.9: Connection post algorithm

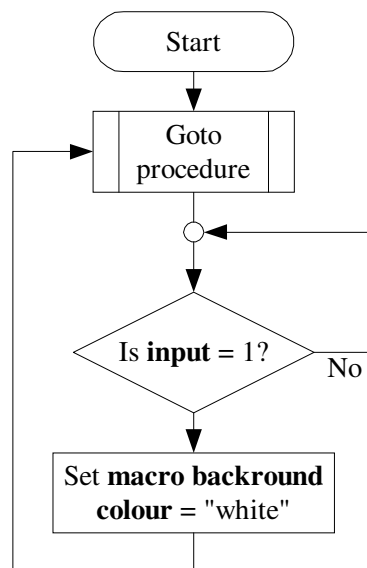


Figure C.10: Macro goto connection post algorithm



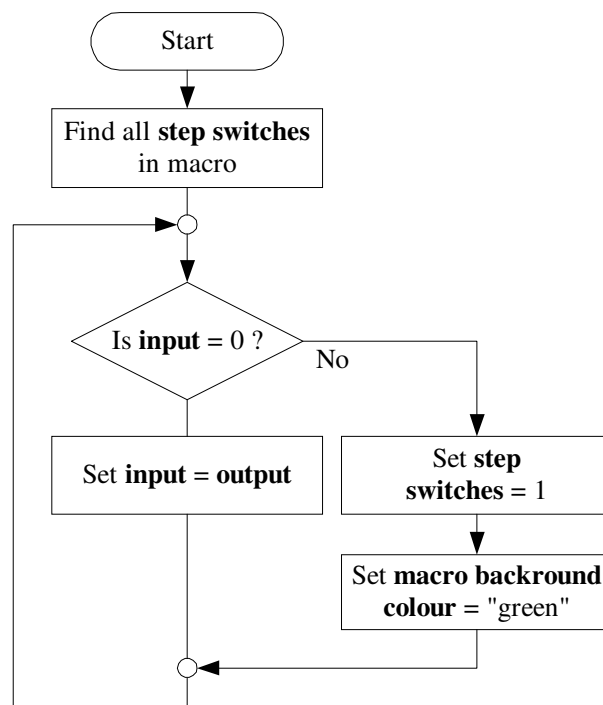


Figure C.11: Macro from connection post algorithm

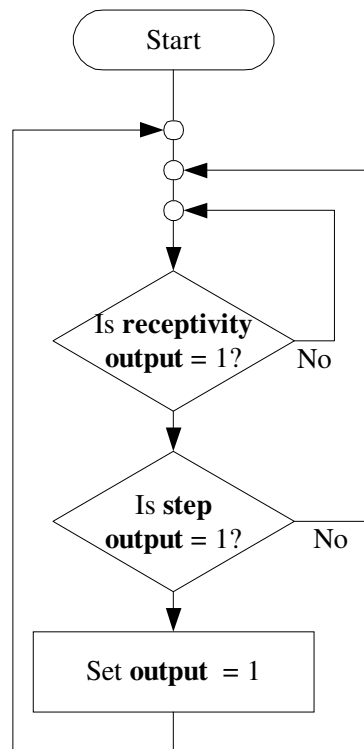


Figure C.12: Gate algorithm

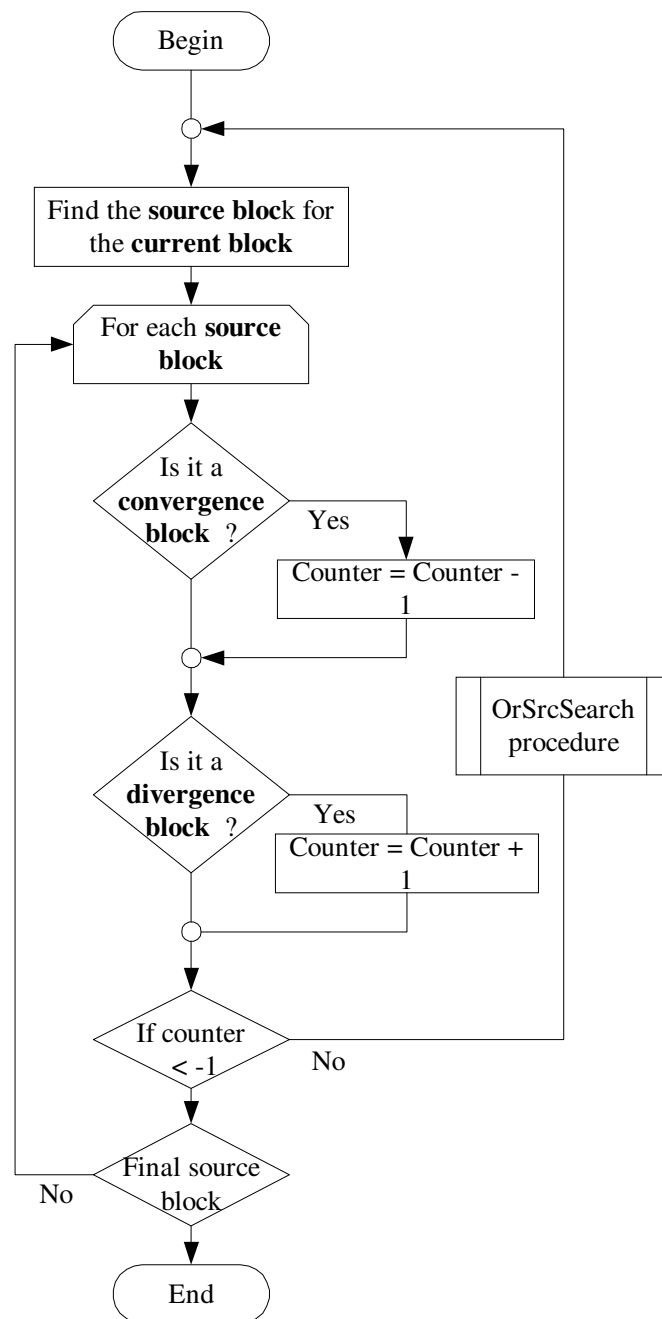


Figure C.13: OrSrcSearch algorithm

---



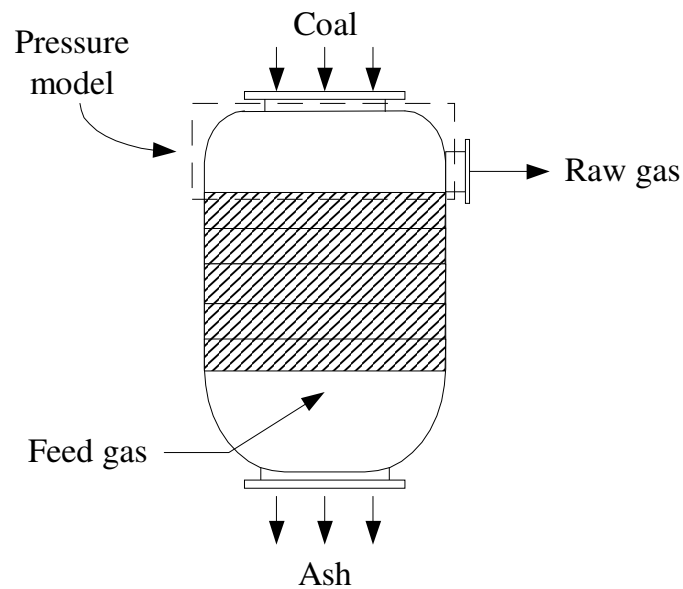
---

# APPENDIX D

---

## Pressure model

A time dependent model describing the pressure in the reaction chamber was developed and can be seen in figure figure D.1. The inlet gas flow conditions was taken as the steady state composition and temperature obtained with the oxygen blown model of the gasification unit.

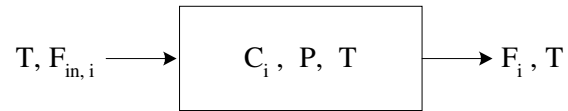


**Figure D.1:** Dynamic pressure model description

### D.1 Equation development

The gas phase component balances for the defined system (figure D.2) for the well-mixed gas volume ( $V$ ) is written:

$$\frac{dC_i}{dt} = \frac{F_{i,in} - F_i}{V} \quad (D.1)$$



**Figure D.2:** Dynamic pressure calculation description

The pressure can be related to the composition using the ideal gas law:

$$P = \sum C_i RT \quad (\text{D.2})$$

The outlet flow rate as a function of the volumetric outlet raw gas flow is:

$$F_i = GC_i \quad (\text{D.3})$$

It can be seen that the pressure is a non-linear function of the volumetric flow rate:

$$P = \frac{\sum C_i RT}{G} \quad (\text{D.4})$$