CHAPTER 9: CONTROL IMPLEMENTATION

9.1 INTRODUCTION

The aim of this chapter is to illustrate how the controller is implemented on the full non-linear plant model, with the use of custom written software. In the implementation it will be necessary to perform a final fine-tuning of the controller. A brief explanation of the implementation software structure is also necessary. The final controller is evaluated against manual control, as shown in the simulation in Chapter 5.

During the controller design phase, the objective was to obtain a controller with acceptable performance. Once this objective is reached, the next step is to test the controller on the "plant" for which it was designed and to fine-tune it. Fine-tuning or "on-line tuning" is essentially done using trial-and-error. To make it effective however, a strategy is devised to determine how tuning parameters should be utilised in the process. The strategy is based on findings from the previous chapter. There are three rules that will be applied:

- If a limited output does not come close to its limit, decrease its cost function weight;
- If a limited output exceeds its limit, increase its cost function weight and/or take its setpoint further away from (further lower than) the limit, so as to regulate it better;
- If the relative pressure can be regulated better, increase its cost function weight, unless it causes undesirable oscillatory or unstable behaviour.

In order to do meaningful tests, a fundamental understanding of the simulation system is necessary. To illustrate the chronological order of the program implementation, a program flowchart [50] is used. Program flowcharts are also used to facilitate the discussions of the MPC and the quadratic programming (QP) algorithms.

In Section 9.2 the structure of the non-linear closed loop simulation is explained. Section 9.3 gives the structure of the MPC algorithm. Section 9.4 gives the structure of the QP optimisation algorithm. Section 9.5 gives the structure of the plant simulation. In Section 9.6 the fine-tuning of the controller is discussed and illustrated. The evaluation of each successive improvement is done and the results are shown. In Section 9.7 the achievement of the control objectives are discussed.

9.2 SIMULATION STRUCTURE

The structure of the simulation is illustrated in the program flowchart in Fig.9.1, where the blocks with bold edges indicate algorithms that are discussed in more detail. The disturbance inputs and initial conditions are the same as those discussed in Chapter 5. The internal model prediction

matrices are calculated in advance, since the assumption of superposition is used with the internal model predictions. This assumption is valid for the internal model, since it is a linear system. For matrix manipulations such as matrix inversion see [51,52] and (programmed in C) [53].

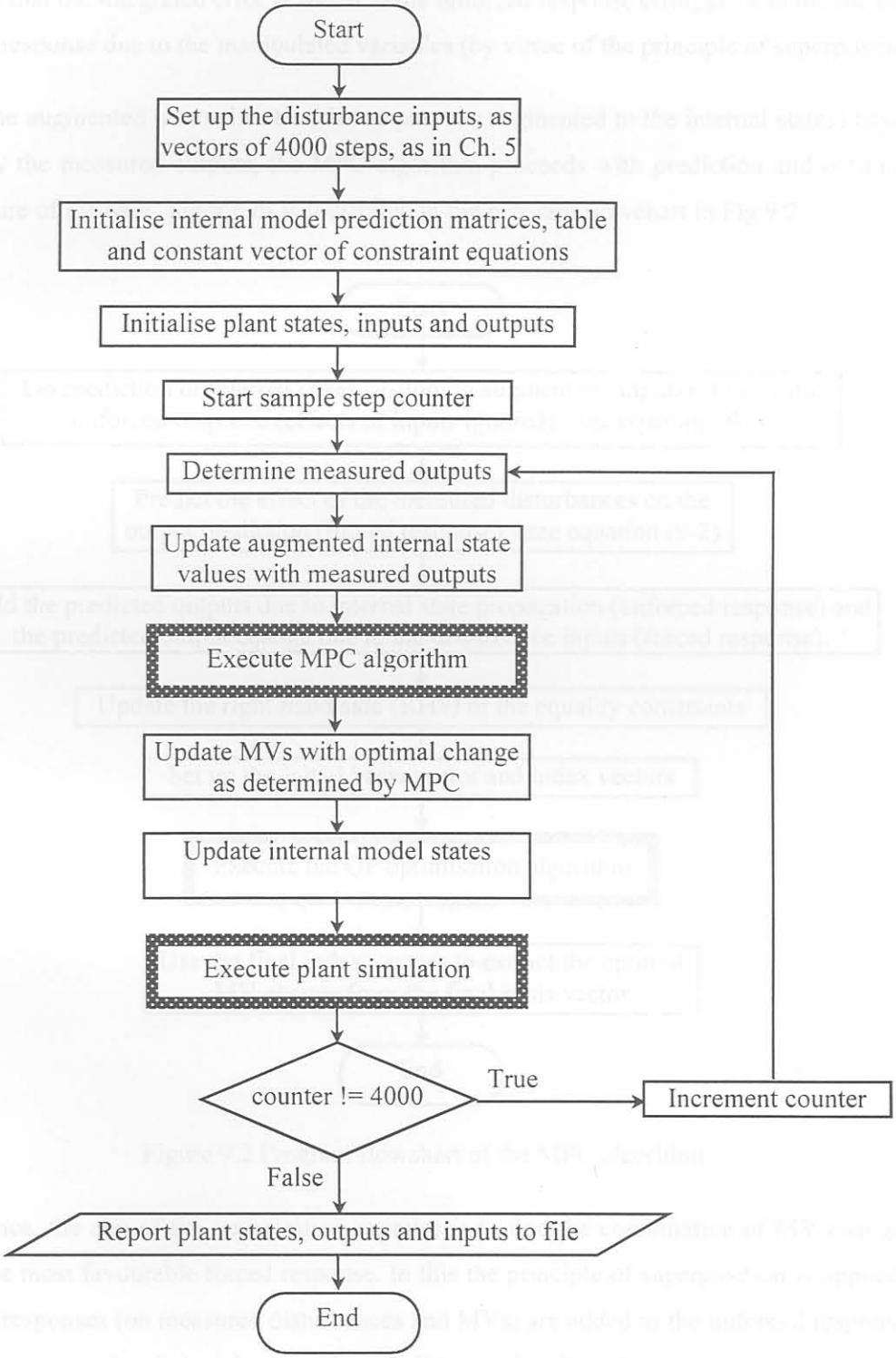

Figure 9.1 Program flowchart of non-linear simulation structure

9.3 MPC ALGORITHM

For the basic MPC algorithm there is no integrated error term and since it was added only later (Section 8.4) to improve the regulation capability of the controller, it is not shown here. It suffices to mention that the integrated error is added to the unforced response error, prior to the inclusion of the forced response due to the manipulated variables (by virtue of the principle of superposition).

After the augmented internal states (the outputs are augmented to the internal states) have been updated by the measured outputs, the MPC algorithm proceeds with prediction and optimisation. The structure of the MPC algorithm is illustrated in the program flowchart in Fig.9.2.

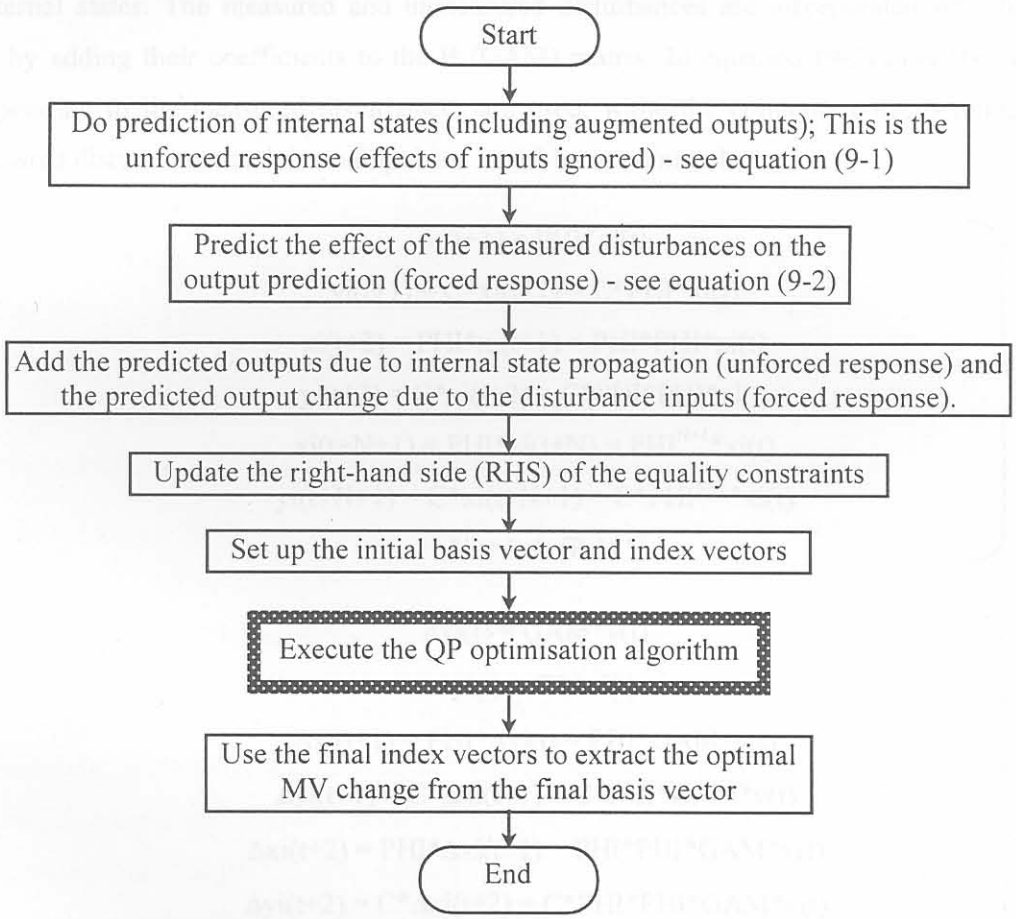

Figure 9.2 Program flowchart of the MPC algorithm

In essence, the aim of the optimisation exercise is to find the combination of MV changes that produce the most favourable forced response. In this the principle of superposition is applied since the forced responses (on measured disturbances and MVs) are added to the unforced response. The unforced response is shown in equation (9-1) and the forced response (on the measured disturbances) is shown in equation (9-2).

The following parameters appear in equations (9-1) and (9-2):

xi – internal model states, augmented with the outputs;

yi – internal model outputs;

v – measured disturbances.

Δxi and Δyi are the forced responses due to the measured disturbances, and are added to xi and yi respectively, by virtue of the priciple of superposition.

The PHI and GAM matrices are the discrete time versions of the A and B matrices of a continous time-invariant linear system model. The PHI matrix has a dimension equal to the sum of the number of states and the number of outputs. This is due to the augmentation of the outputs to the internal states. The measured and unmeasured disturbances are incorporated into the linear model by adding their coefficients to the B (GAM) matrix. In equation (9-2) only the collumns corresponding to the measured disturbances are used, while the collumns corresponding to the unmeasured disturbances and the manipulated variables are omitted.

$$
\begin{aligned}
xi(t+1) &= PHI*xi(t) \\
yi(t+1) &= C*xi(t+1) = C*PHI*xi(t) \\
xi(t+2) &= PHI*xi(t+1) = PHI*PHI*xi(t) \\
yi(t+2) &= C*xi(t+2) = C*PHI*PHI*xi(t) \\
xi(t+N+1) &= PHI*xi(t+N) = PHI^{N+1}*xi(t) \\
yi(t+N+1) &= C*xi(t+N+1) = C*PHI^{N+1}*xi(t) \\
N &= 1 \rightarrow (P-1)
\end{aligned}
\qquad (9\text{-}1)
$$

$$
\begin{aligned}
\Delta xi(t) &= GAM*v(t) \\
\Delta yi(t) &= C*\Delta xi(t) \\
\Delta xi(t+1) &= PHI*\Delta xi(t) = PHI*GAM*v(t) \\
\Delta yi(t+1) &= C*\Delta xi(t+1) = C*PHI*GAM*v(t) \\
\Delta xi(t+2) &= PHI*\Delta xi(t+1) = PHI*PHI*GAM*v(t) \\
\Delta yi(t+2) &= C*\Delta xi(t+2) = C*PHI*PHI*GAM*v(t) \\
\Delta xi(t+N+1) &= PHI*\Delta xi(t+N) = PHI^{N+1}*GAM*v(t) \\
\Delta yi(t+N+1) &= C*\Delta xi(t+N+1) = C*PHI^{N+1}*GAM*v(t) \\
N &= 1 \rightarrow (P-1)
\end{aligned}
\qquad (9\text{-}2)
$$

9.4 QUADRATIC PROGRAMMING ALGORITHM

George B. Dantzig originally formulated the QP algorithm used by Morari and Ricker [12] in the MPC toolbox for Matlab®. Boot [47], Dano [54] and Peressini *et al* [55] discuss this algorithm, and [56,57,58] discuss related theory in optimisation and optimality in mathematical programming.

A brief summary of the algorithm will suffice. The algorithm is also known as the "Simplex" method or the "Dantzig-van de Panne" method, and is preferred to the other methods (Wolfe's method and Beale's method) as it has proven to be far more efficient [54].

The simplex method is intended for linear systems with a quadratic cost (performance-) function and linear inequality constraints. The QP algorithm is guaranteed (for a linear system) to progress monotonically [47] towards an optimal solution. To apply the QP algorithm, a linear system model is necessary, so that the principle of superposition may be used. This implies that if there is a feasible optimal solution, then the algorithm will find it in a finite number of iterations.

If there is no feasible solution (due to severe constraints that cannot possibly be satisfied), then the algorithm returns without a solution. Therefore, if there is a (any) feasible solution, the algorithm will return with an optimal feasible solution. The objective with the simplex method is to obtain a table in standard form, while the solution has all the basis (containing the primal variables) elements non-negative. To do this, a set of rules is followed, which is explained in [47,54,55] and briefly described in Fig.9.3 that summarises the main procedures of the algorithm.

The following variables are mentioned in Fig.9.3: The dual variables are Lagrangian multipliers for the inequality constraint equations, while the primal variables are the actual input or output variables that feature in the constraint equations.

The common simplex method is used in linear programming as well as quadratic programming. In the common simplex method a variable is selected for replacement as follows:

- From the column (q) of the variable that is to be introduced to the basis, form the quotients of the basis element B_i and the table element TAB_{iq} for all i, (the available rows);
- Test which quotient is the smallest non-negative, let
  $Q_{min} = B_p/TAB_{pq} <= B_i/TAB_{iq}$ for all i, **and** $Q_{min} >= 0$;
- Then the variable in row (p) of the basis is the one to be replaced;
- In other words, TAB_{pq} is to be the pivot element (pivoting exactly as in Gaussian elimination).

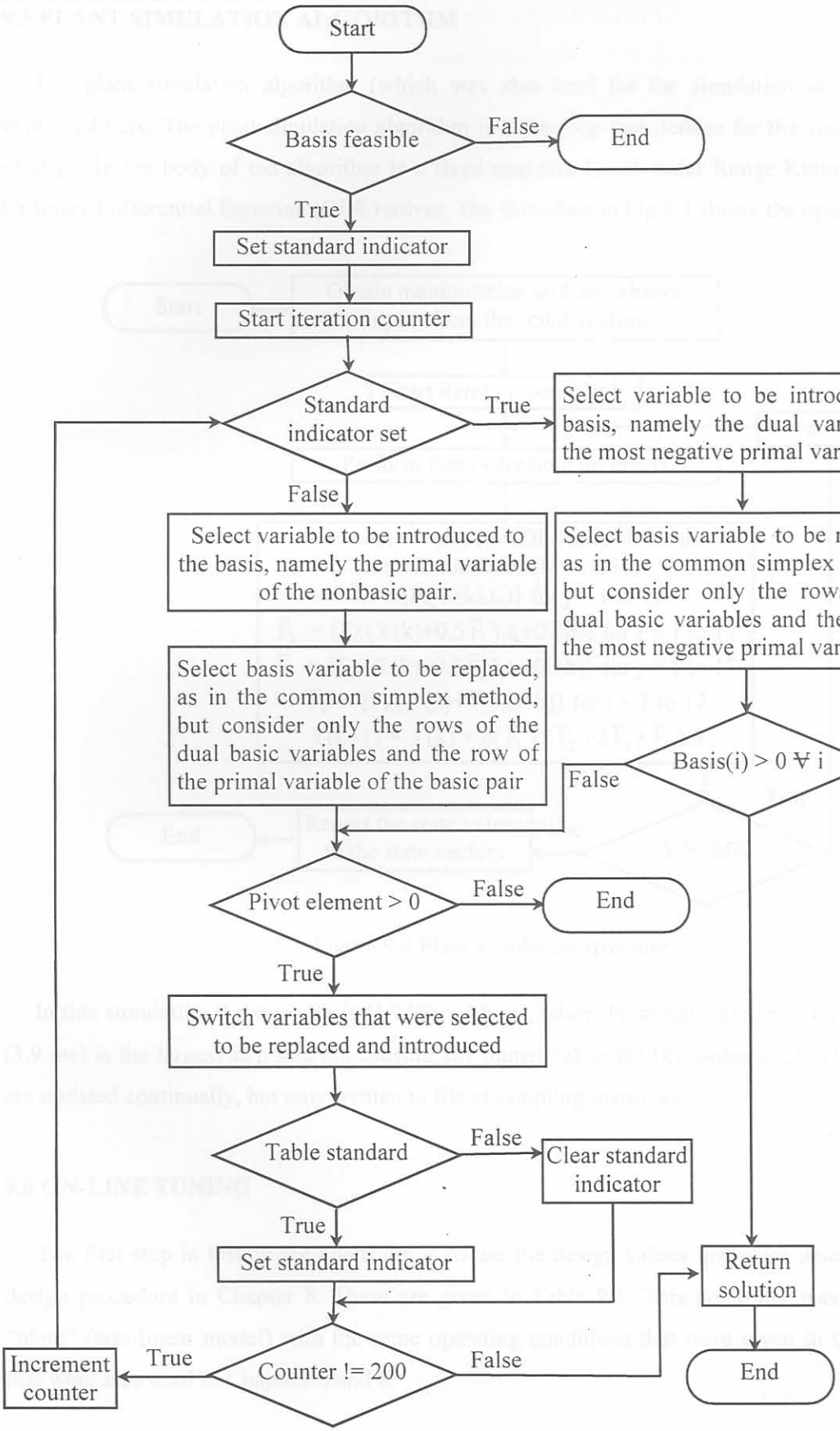

Figure 9.3 Quadratic programming algorithm

9.5 PLANT SIMULATION ALGORITHM

The plant simulation algorithm (which was also used for the simulation in Chapter 5) is explained here. The plant simulation algorithm is a **for**-loop that iterates for the specified number of steps. In the body of the algorithm is a fixed step-size fourth-order Runge-Kutta multivariable Ordinary Differential Equation (ODE) solver. The flowchart in Fig.9.4 shows the operation:

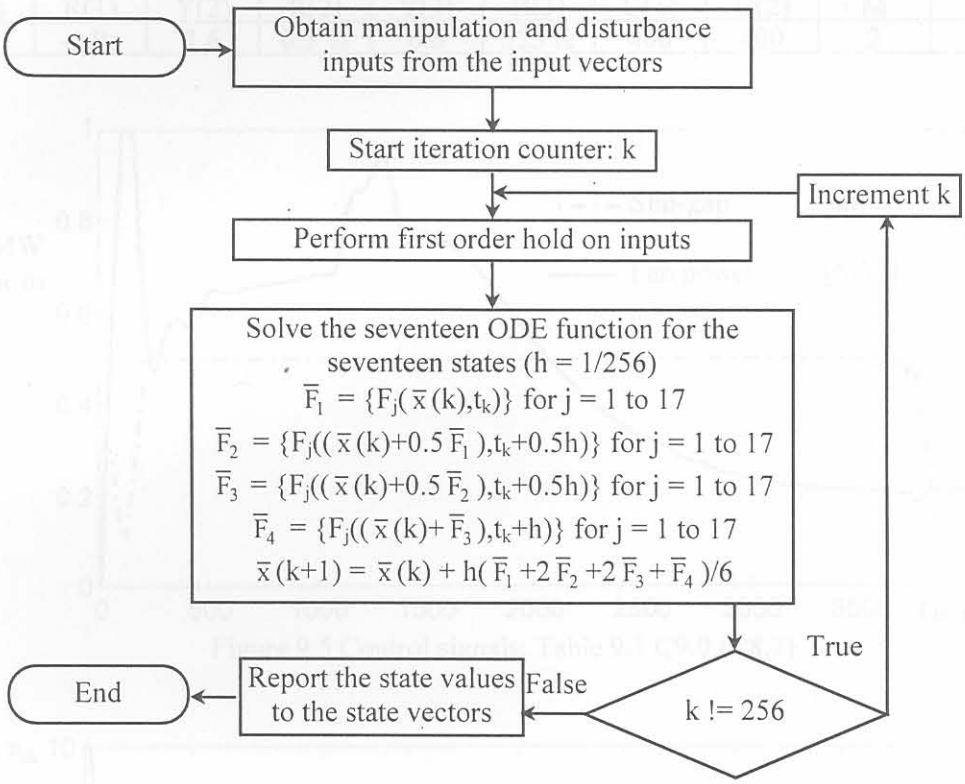

Figure 9.4 Plant simulation structure

In this simulation the step size is (1/256) = 3.9 ms, while the sampling time is 1 s. The step size (3.9 ms) is the largest step size not causing any numerical instability whatsoever. The state values are updated continually, but only written to file at sampling instances.

9.6 ON-LINE TUNING

The first step in testing the controller is to use the design values that were determined by the design procedure in Chapter 8. These are given in Table 9.1. This controller was tested on the "plant" (non-linear model) with the same operating conditions that were given in Chapter 5, and that were also used in Chapters 6 and 8.

The control signals are shown in Fig.9.5, and the three outputs in Fig.9.6, Fig.9.7 and Fig.9.8. After several experiments with the non-linear closed-loop simulation it was determined that the setpoint for the off-gas temperature needs to be made at least 100 K below the off-gas temperature limit, otherwise an infeasible QP problem results. To remedy this the setpoint for the off-gas temperature was set at 523 K. The other settings are also shown in Table 9.1.

Table 9.1: Controller settings from linear design (C9.0 = = C8.7)

Y(1)	R(1)	Y(2)	R(2)	Y(3)	R(3)	U(1)	U(2)	M	P
4.5	-5 Pa	1.6	0.5 %	0.6	523 K	400	400	2	6

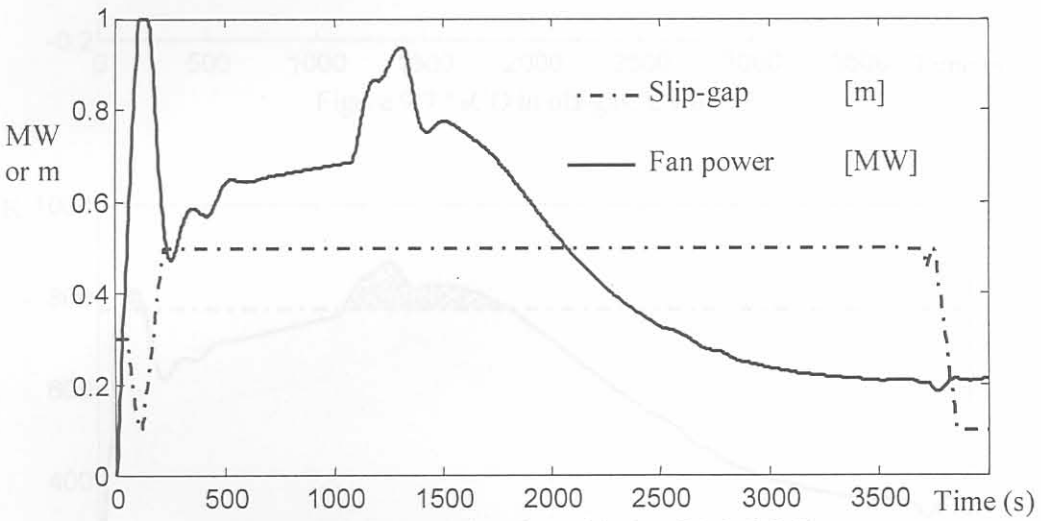

Figure 9.5 Control signals: Table 9.1 C9.0 (C8.7)

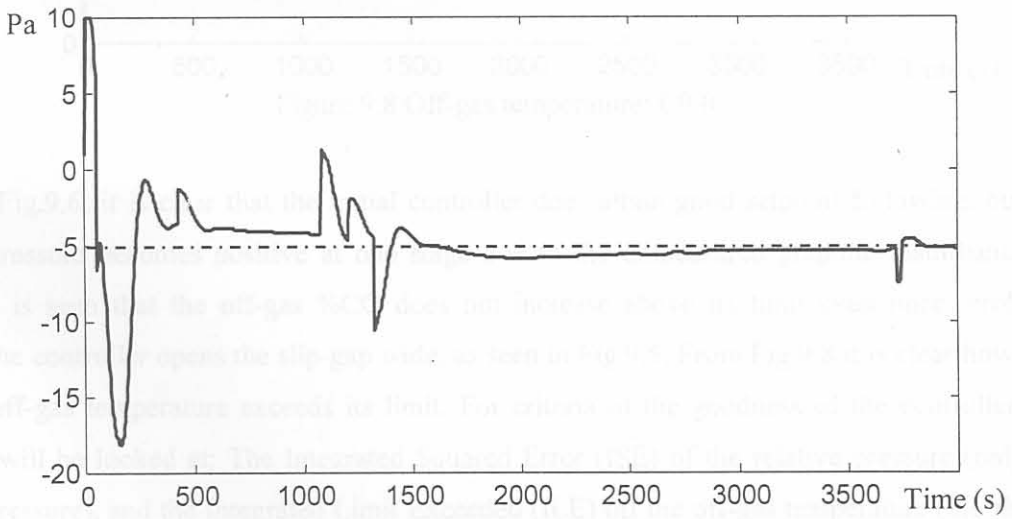

Figure 9.6 Relative pressure: C9.0

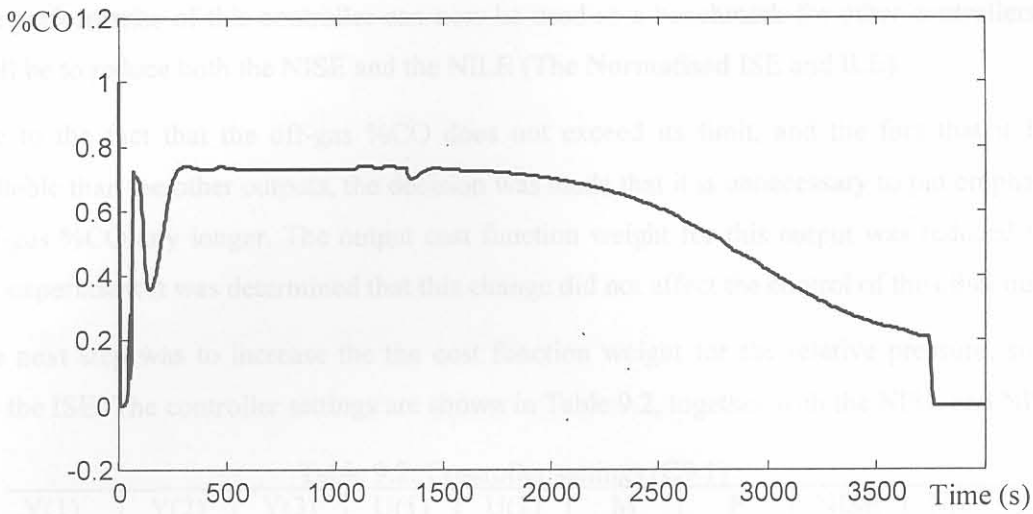

Figure 9.7 %CO in off-gas: C9.0

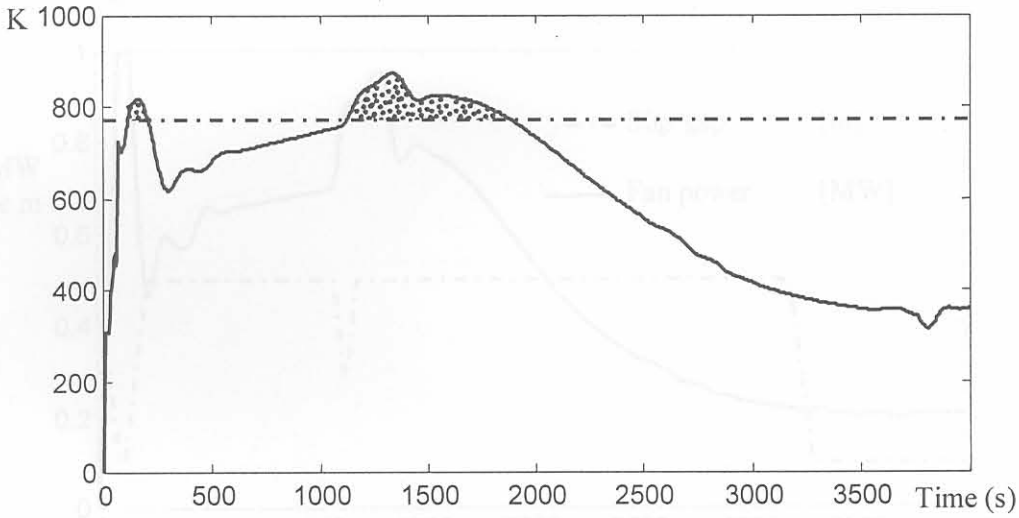

Figure 9.8 Off-gas temperature: C9.0

From Fig.9.6, it is clear that the initial controller does attain good setpoint following, but the relative pressure becomes positive at one stage due to the unmeasured graphite disturbance. In Fig.9.7 it is seen that the off-gas %CO does not increase above its limit even once, probably because the controller opens the slip-gap wide, as seen in Fig.9.5. From Fig.9.8 it is clear however, that the off-gas temperature exceeds its limit. For criteria of the goodness of the controller two numbers will be looked at: The Integrated Squared Error (ISE) of the relative pressure (only the relative pressure), and the Integrated Limit Exceeded (ILE) off the off-gas temperature (the shaded part in Fig.9.8). For the initial controller the values were as follows:

ISE(C9.0) = 32593 if normalised it is NISE = 1;

ILE(C9.0) = 41724 if normalised it is NILE = 1.

The performance of this controller can now be used as a benchmark for other controllers. The aim will be to reduce both the NISE and the NILE (The **Normalised** ISE and ILE).

Due to the fact that the off-gas %CO does not exceed its limit, and the fact that it is less controllable than the other outputs, the decision was made that it is unnecessary to put emphasis on the off-gas %CO any longer. The output cost function weight for this output was reduced to 1.0, and by experiment it was determined that this change did not affect the control of the other outputs.

The next step was to increase the the cost function weight for the relative pressure, so as to reduce the ISE. The controller settings are shown in Table 9.2, together with the NISE and NILE:

Table 9.2: Controller settings (C9.1)

Y(1)	Y(2)	Y(3)	U(1)	U(2)	M	P	NISE	NILE
5.5	1.0	0.6	400	400	2	6	0.9198	1.0574

The control signals of C9.1 are shown in Fig.9.9 and the relative pressure in Fig.9.10.

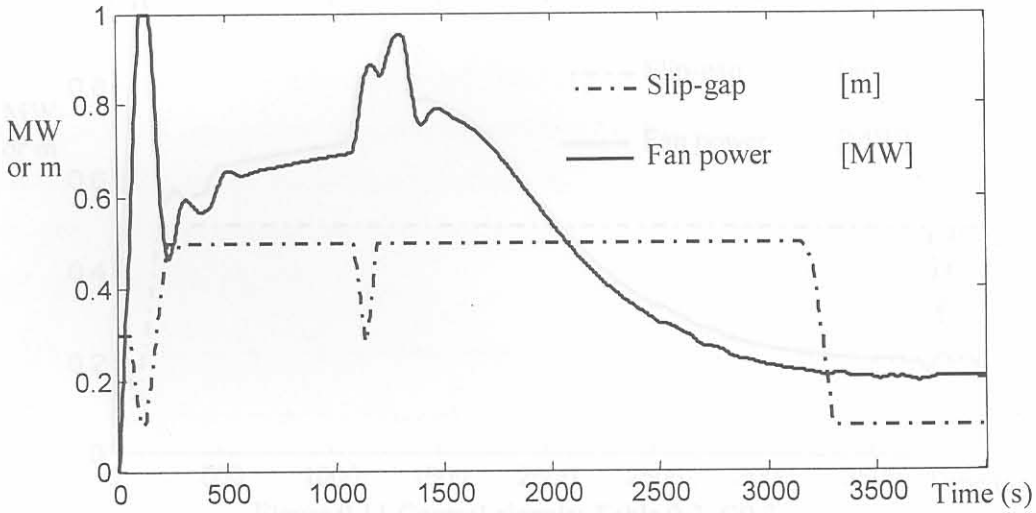

Figure 9.9 Control signals: Table 9.2: C9.1

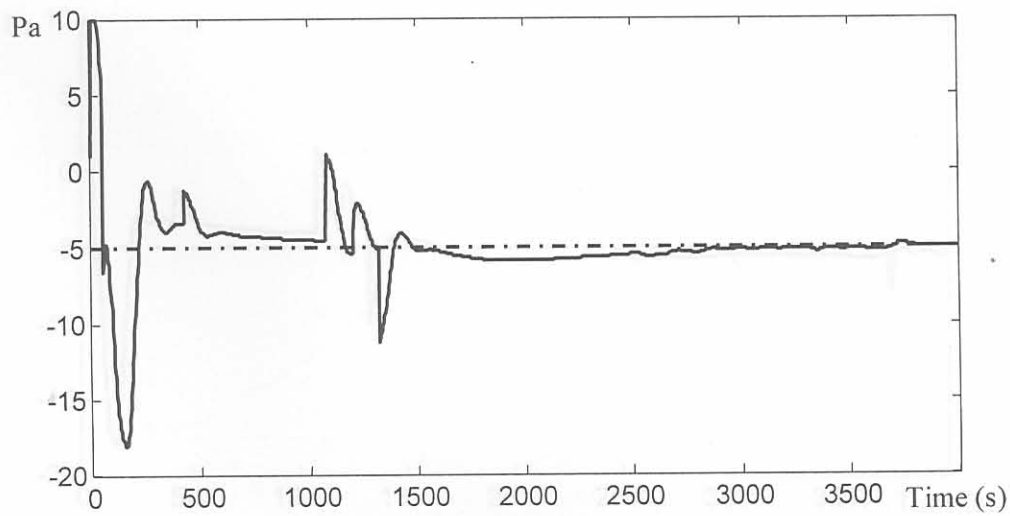

Figure 9.10 Relative pressure: C9.1

The performance of C9.1 relative to C9.0 is obtained by normalising the ISE and the ILE of C9.1 with respect to that of C9.0. The normalised ISE (NISE) and the normalised ILE (NILE) will be used as performance criteria from now on. For C9.1 these are:

NISE = 0.9198

NILE = 1.0574

The next step is to increase the cost function weight for the off-gas temperature so as to decrease the ILE. The controller settings are shown in Table 9.3, together with the NISE and NILE:

Table 9.3: Controller settings (C9.2)

Y(1)	Y(2)	Y(3)	U(1)	U(2)	M	P	NISE	NILE
5.5	1.0	**0.8**	400	400	2	6	0.9531	0.9413

The control signals of C9.2 are shown in Fig.9.11 and the relative pressure in Fig.9.12. The increased effort to reduce the off-gas temperature compromised the relative pressure regulation.

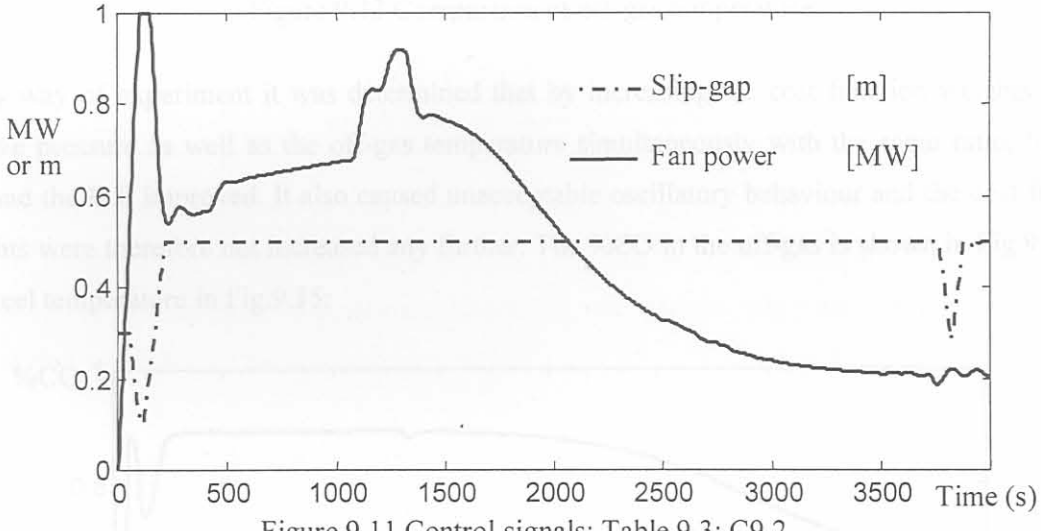

Figure 9.11 Control signals: Table 9.3: C9.2

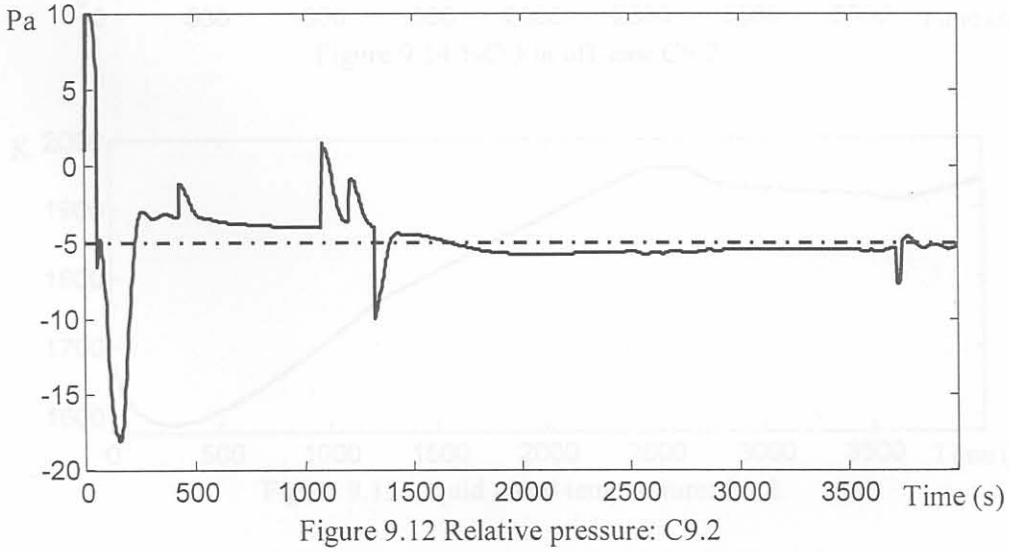

Figure 9.12 Relative pressure: C9.2

At the same time, the off-gas temperature was brought slightly out of the exceed limit area so as to reduce the NILE, as is shown in Fig.9.13:

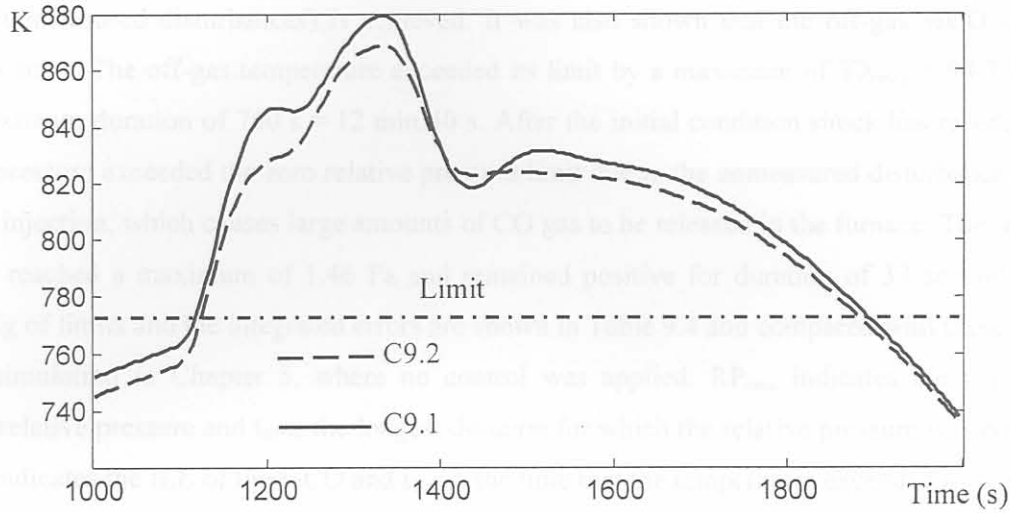

Figure 9.13 Comparison of off-gas temperature

By way of experiment it was determined that by increasing the cost function weights for the relative pressure as well as the off-gas temperature simultaneously with the same ratio, both the ISE and the ILE improved. It also caused unacceptable oscillatory behaviour and the cost function weights were therefore not increased any further. The %CO in the off-gas is shown in Fig.9.14 and the steel temperature in Fig.9.15:

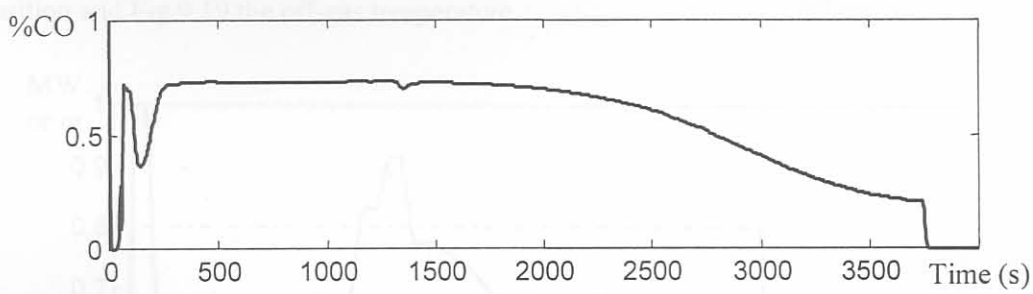

Figure 9.14 %CO in off-gas: C9.2

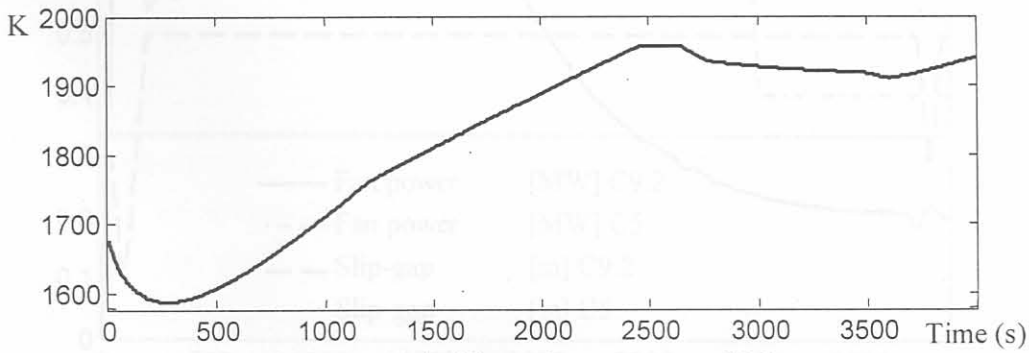

Figure 9.15 Liquid metal temperature: C9.2

9.7 CONTROL EVALUATION

It was illustrated in the previous section that acceptable regulation of the relative pressure (despite unmeasured disturbances) is achieved. It was also shown that the off-gas %CO is kept below its limit. The off-gas temperature exceeded its limit by a maximum of TX_{max} = 94.7 K and for a maximum duration of 760 s = 12 min:40 s. After the initial condition shock has receded, the relative pressure exceeded the zero relative pressure limit due to the unmeasured disturbance of the graphite injection, which causes large amounts of CO gas to be released in the furnace. The relative pressure reached a maximum of 1.46 Pa and remained positive for duration of 37 seconds. The exceeding of limits and the integrated errors are shown in Table 9.4 and compared with those of the similar simulation in Chapter 5, where no control was applied. RP_{max} indicates the maximum positive relative pressure and $t_{RP>0}$ the longest duration for which the relative pressure was positive. $ILE_{\%CO}$ indicates the ILE of the %CO and $t_{T>limit}$ the time that the temperature exceeded its limit.

Table 9.4: Controller evaluation (C9.2) against manual control (Chapter 5)

Controller	NISE	RP_{max}	$t_{RP>0}$	$\%CO_{max}$	$ILE_{\%CO}$	NILE	TX_{max}	$t_{T>limit}$
None	3.2461	4.10 Pa	263 s	3.54 %	1894	1.5682	92.4 K	1140 s
C9.2	0.9531	1.46 Pa	37 s	0.74 %	0	0.9413	94.7 K	760 s

To visualise the comparison in Table 9.4 the following four figures for comparison are included: Fig.9.16 compares the manipulated variables for C9.2 against the manual control of Chapter 5. Fig.9.17 compares the relative pressure between the two options, Fig.9.18 compares the off-gas composition and Fig.9.19 the off-gas temperature.

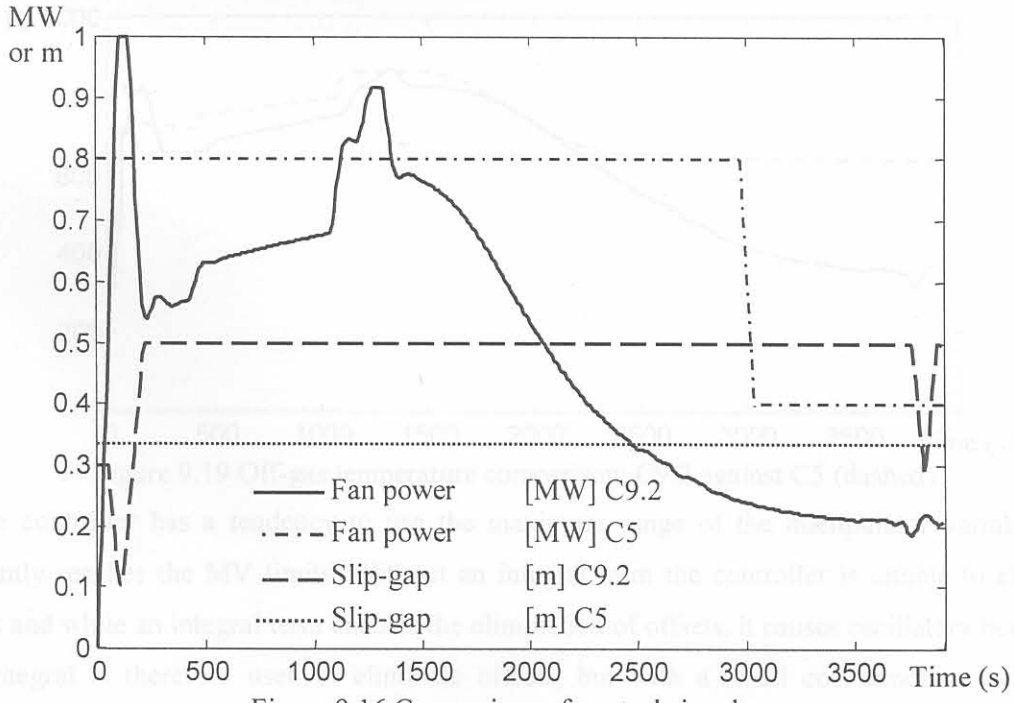

Figure 9.16 Comparison of control signals

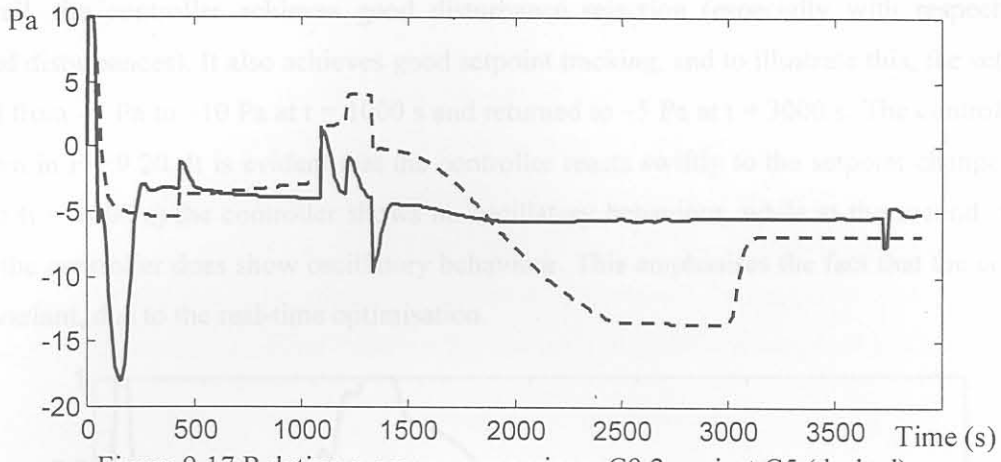

Figure 9.17 Relative pressure comparison: C9.2 against C5 (dashed)

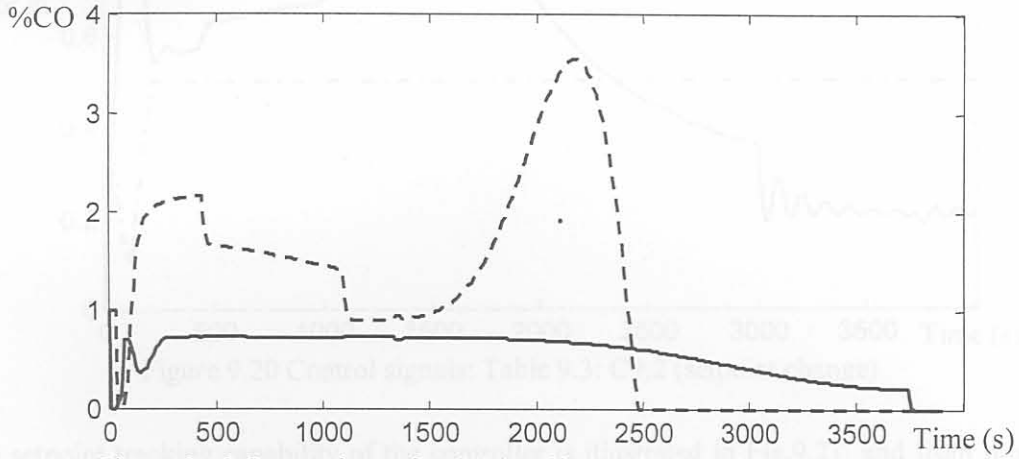

Figure 9.18 Comparison of %CO in off-gas: C9.2 against C5 (dashed)

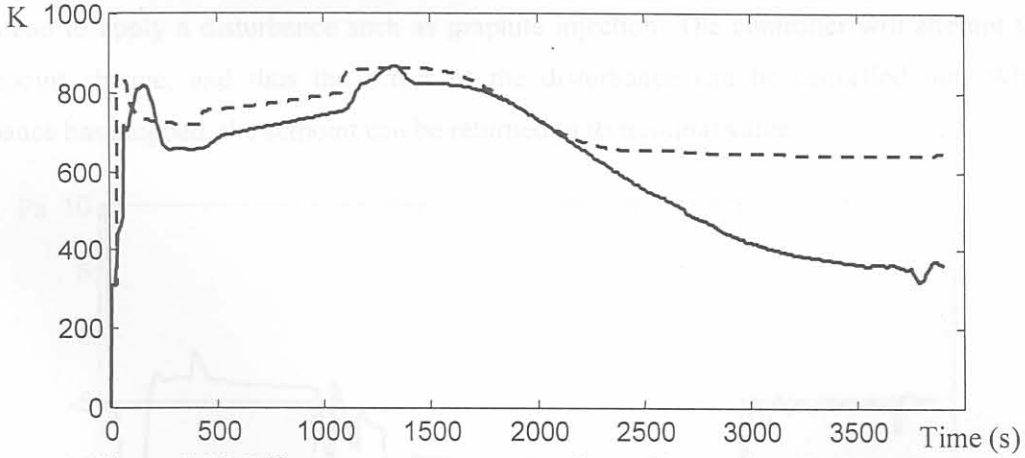

Figure 9.19 Off-gas temperature comparison: C9.2 against C5 (dashed)

The controller has a tendency to use the maximum range of the manipulator variables and frequently reaches the MV limits. Without an integral term the controller is unable to eliminate offsets and while an integral term enables the elimination of offsets, it causes oscillatory behaviour. The integral is therefore used to eliminate offsets, but with a small cost function weight, as determined in Chapter 8.

Overall, the controller achieves good disturbance rejection (especially with respect to the measured disturbances). It also achieves good setpoint tracking, and to illustrate this, the setpoint is changed from –5 Pa to –10 Pa at t = 1000 s and returned to –5 Pa at t = 3000 s. The control signals are shown in Fig.9.20. It is evident that the controller reacts swiftly to the setpoint change. At the first step (t = 1000 s) the controller shows no oscillatory behaviour, while at the second step (t = 3000 s) the controller does show oscillatory behaviour. This emphasises the fact that the controller is time-variant, due to the real-time optimisation.

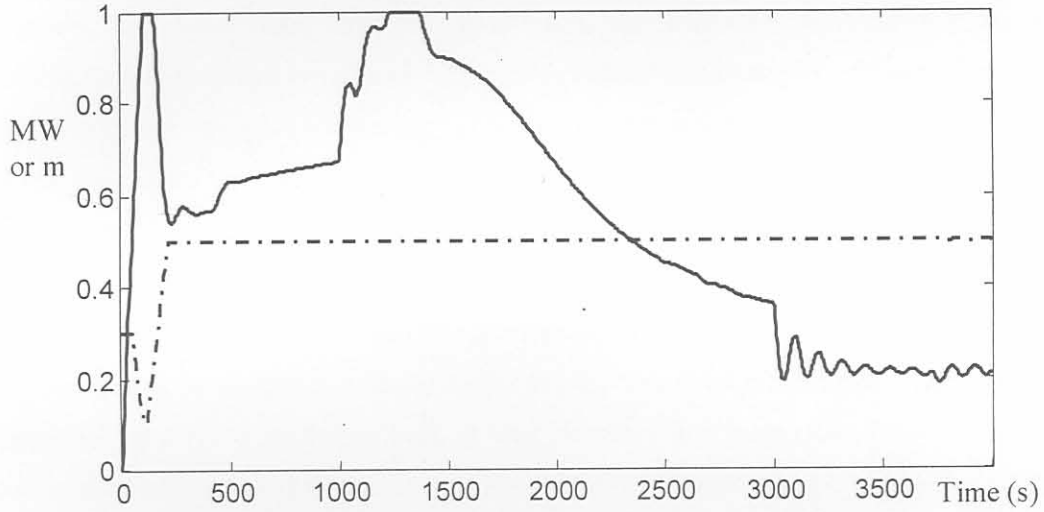

Figure 9.20 Control signals: Table 9.3: C9.2 (setpoint change)

The setpoint tracking capability of the controller is illustrated in Fig.9.21, and from this graph the deduction can be made that operators may temporarily make the setpoint more negative when they intend to apply a disturbance such as graphite injection. The controller will attempt to track the setpoint change, and thus the effect of the disturbance can be cancelled out. When the disturbance has stopped, the setpoint can be returned to its nominal value.

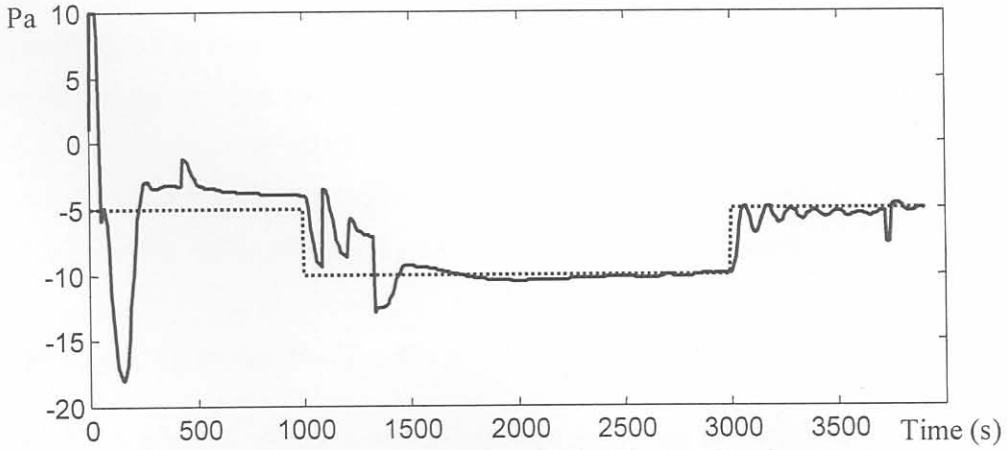

Figure 9.21 Relative pressure: C9.2 with setpoint changes

9.8 CONCLUSION

In Section 9.2 the structure of the non-linear closed loop simulation was discussed. Section 9.3 explained the structure of the MPC algorithm. In Section 9.4 the structure of the QP optimisation algorithm was explained. Section 9.5 discussed the structure of the plant simulation. In Section 9.6 the fine-tuning of the controller was discussed and illustrated. The evaluation of each successive improvement is performed as the results were shown. In Section 9.7 the achievement of the control objectives were discussed. The final conclusions and recommendations are given in Chapter 10.