Journal of Process Control, Volume 16, Issue 8, September 2006, Pages 795-808

# NONLINEAR MODEL PREDICTIVE CONTROL FOR THE ALSTOM GASIFIER

R K Al Seyab[‡] Y Cao[‡*]

[‡]Cranfield University, UK

**Keywords:** : Predictive control, Gasification, Wiener model, Feedforward neural networks, Linearisation.

**Abstract**

In this work a nonlinear model predictive control based on Wiener model has been developed and used to control the ALSTOM gasifier. The 0% load condition was identified as the most difficult case to control among three operating conditions. A linear model of the plant at 0% load is adopted as a base model for prediction. A nonlinear static gain represented by a feedforward neural network was identified for a particular output channel–namely, fuel gas pressure, to compensate its strong nonlinear behaviour observed in open-loop simulations. By linearising the neural network at each sampling time, the static nonlinear model provides certain adaptation to the linear base model at all other load conditions. The resulting controller showed noticeable performance improvement when compared with pure linear model based predictive control.

[*]To whom correspondence should be addressed (y.cao@cranfield.ac.uk).

# 1 Introduction

Model predictive control (MPC) has become a first choice of control strategy in industry because it is intuitive and can explicitly handle multivariable systems with constraints. The basic control strategy in MPC is the selection of a set of future control moves (control horizon) and minimise a cost function based on the desired output trajectory over a prediction horizon with a chosen length. This requires a reasonably accurate internal model, that captures the essential nonlinearities of the process under control and predict the dynamic behaviour multi–step ahead [1].

Until recently, industrial applications of MPC have relied on linear dynamic models even though most processes are nonlinear. Linear MPC (LMPC) is probably acceptable and sometime desirable when the process operates at a single set-point and the primary use of the controller is the rejection of small disturbances [2]. If the plant exhibits severe nonlinearities, the usefulness of MPC based on a linear model is limited, particularly if it is used to transfer the plant from one operating point to another as in the case under study. The obvious solution is to use a nonlinear model. However, the extension to nonlinear model based predictive control has not been always very successful. The main hurdle facing this extension is the significant computational burden associated with solving a set of nonlinear differential equations and a nonlinear dynamic optimisation problem online. In addition, the convexity of the optimisation problem in this case is not guaranteed, which is a serious drawback for online applications [3].

A number of researchers have developed NMPC approaches based on a linearisation of the plant model for the prediction phase. In this solution, the model equations are linearised around the operating point and solved within an efficient convex optimisation method to obtain the optimum control step. The linearisation step is performed once over the prediction horizon [3, 4], or further at a number of time steps inside the prediction horizon [5, 6]. This strategy has proved to be

highly successful in controlling mildly nonlinear processes [7].

The internal model in NMPC can be based on the physical laws governing the behaviour of the true system and often referred to as a first-principle model [8, 9]. Alternatively the model is derived from measurements of input and output data from the real plant. This method relies heavily on system identification and the resulting model is called an empirical or black–box model [10, 11]. First-principle models are valid globally and can predict system dynamics over the entire operating range. The development of a reliable first-principle model is, in general, a difficult and time consuming task. The nonlinear black-box models, on the other hand, have certain advantages over the first-principle models in terms of development time and efforts. If chosen wisely, it can simplify and accelerate the controller as well.

There are many different black-box nonlinear models utilised for NMPC include: Volterra models [12], Polynomial autoregressive moving average model with exogenous inputs (polynomial ARMAX) [13], Hammerstein and Wiener type models [14, 15, 7, 16], artificial neural networks [10], and others. Among these types of models Hammerstein and Wiener models have a special structure that facilitates their application to NMPC. Wiener model is particularly useful in representing the nonlinearities of process without introducing the complications associated with general nonlinear operators [7, 16]. This model consists of a linear dynamic element followed in series by a static nonlinear element. Hammerstein model contains the same elements in the reverse order.

These models have been shown to adequately represent many of the nonlinearities commonly encountered in industrial processes such as distillation and pH neutralisation [17]. Wiener models may be incorporated into MPC schemas in a unique way which effectively removes the nonlinearity from the control problem, preserving many of the favourable properties of linear MPC [15, 7, 18, 16].

An approach of identification and control using a Wiener model was proposed by Al Duwaish et al [15]. The authors proposed a Wiener model consisting of an autoregressive moving average (ARMA) model as a linear dynamic model in cascade with a multi-layer feedforward neural network (FFNN). A controller using the Wiener model is constructed by inserting the inverse nonlinearity of the FFNN in an appropriate loop location. The inverse of the static nonlinearity is modelled by another FFNN. A linear controller was designed for the ARMA model using linear control theory. Norquay et al [7] proposed another system identification and model predictive control approach using Wiener model. Two linear models were chosen for the linear dynamic element of the Wiener model: autoregressive with exogenous (ARX) model and the step-response model. While a low-order piecewise polynomial is used for the nonlinear static element of the model. The model is incorporated into an unconstrained LMPC algorithm by removing the nonlinear element from the control problem via using a static inverse nonlinearity. A similar NMPC approach is proposed in [18] but with input and output constraints. The nonlinear constraints of the outputs were linearised using the inverse static nonlinearity in that work.

Recently, Cervantes et al [16] presented a Wiener model based NMPC approach with an invertible piecewise linear gain. The inverse of the piecewise linear function is used to map the setpoint, output upper and lower bounds, and the measured outputs so that linear relationship is retained with these signals. Then a quadratic programming (QP) routine is used to solve the optimisation problem online.

In this paper, another Wiener model based NMPC approach is developed to control the ALSTOM gasifier. A Wiener structure consisting of a linear multi-input multi-output (MIMO) state-space part followed by a partially nonlinear static part is used to identify a black-box model of the gasifier plant. By linearising the static

part of Wiener model at each optimisation step, the nonlinear model becomes linear and the NMPC is simplified to a classical LMPC which keeps computation easier to perform.

The rest of this paper is organised as follows. Section 2 gives an introduction to the ALSTOM gasifier benchmark problem and an overview of the work. A short description of the nonlinear plant is presented in section 3. Section 4 discusses the partially nonlinear internal model. The formulation used for model predictive control is provided in section 5. Section 6 explains the procedures of nonlinear system identification and controller design for the gasifier. Section 7 presents the simulation results, and in section 8 some conclusions are drawn from this work.

## 2 The ALSTOM Gasifier

The coal gasifier is essentially a chemical reactor where coal reacts with air and steam to produce low calorific value fuel gas, which then can be burnt in a suitably adapted gas turbine, and char. Limestone (sorbent) is also added to the vessel to capture the majority of sulphur present in the coal [19]. In modern advanced power generating plants gasification helps burning coal in a new and environmentally friendly process.

The ALSTOM gasifier was issued as a benchmark problem by the ALSTOM Power Technology Centner [20]. This process involves several challenging issues, such as high order, high nonlinearity and strong interactions between process variables. Furthermore, this process has very stringent upper, lower and rate constraints on the process input and output variables because of safety and environmental issues and the physical nature of the variables themselves.

Based on an industrial scale gasifier, the ALSTOM Power Technology Centner issued a benchmark challenge in 1997 and a second round challenge in 2002. The

first challenge included three linear models representing three operating conditions of the gasifier at 0%, 50% and 100% load respectively. The challenge required the controller to control the gasifier at three load conditions to satisfy input and output constraints in the presence of step and sinusoidal disturbances (see Ref. [20]). An overview and comparison of various control approaches submitted to the first round challenge are given in [19].

None of the controllers proposed in the first round managed to satisfy all the performance criteria within specified constraints. The only MPC approach [21] proposed at the first round challenge involved the use of a LMPC with an additional inner loop to stabilise the process. The inner loop controller is supervised by an outer loop to handle the process constraints.

The second round of the challenge issued in 2002 extended the original problem by providing participants with a nonlinear simulation model of the gasifier in MATLAB/SIMULINK [22]. In addition to the original disturbance tests, two extra tests: load change and coal quality disturbance tests were added. Recently, a group of control solutions for the benchmark problem were presented at "Control-2004" at Bath University, UK in September 2004. Most controllers were reported as capable to control the system in the specified tests except in the coal quality disturbance test because of a char flow rate saturation behaviour. Among the solutions, a linear MPC employing Generalised Predictive Control (GPC) strategy was proposed [23].

In the previous work [23], the operating condition at 0% load point was considered to be the most difficult case to control and a linearised model around this load condition was adopted for the internal model. The controller was able to maintain all the required performance specifications within the input and output constraints at all load conditions. In this work, it is shown that the plant/model mismatch can further be reduced by developing a partially nonlinear Wiener type model instead

of a pure linear model. More specifically, a FFNN is developed as a nonlinear static gain for one of four output channels, fuel gas pressure (PGAS) to compensate its strong nonlinear behaviour observed in the open loop simulation (see Figure 1). The FFNN was then linearised at every sampling instance and used as constant over the prediction horizon to provide an adaptation to the main linear controller. A similar strategy can be used for the other output variables but this was found neither necessary nor very productive. The partially nonlinear model leads to considerable performance improvement compared with the pure linear MPC. Also, the proposed controller was able to control the plant without any constraints violation and satisfied all the benchmark challenge requirements.

# 3 Plant Description

A schematic of the plant [24] is shown in Figure 2. The gasifier is a nonlinear, multivariable system, having five controllable inputs (coal, limestone, air, steam and char extraction), and four outputs (pressure, temperature, bed–mass and gas quality) with a high degree of cross coupling between them. One of its inputs, the limestone feed (WLS) is used to absorb sulphur in the coal and its flow rate must be set to a fixed ratio of 1:10 against the coal feed (WCOL). This leaves effectively 4 degrees of freedom for the control design. The plant inputs and outputs with their limits are given in Tables 1 and 2, respectively. The gasifier is open–loop stable system, has a very complex dynamic behaviour with mixed fast and slow modes. All the output variables take approximately $10^4$ s to reach their steady–state values (see Figure 3). On the other hand, the rising time for gas pressure (PGAS) is very short comparing with other variables. The gasifier proved to be difficult to control as it is both multivariable and nonlinear with a significant cross–coupling between the input and output variables [20]. The full model of the gasifier has 25 states and

the aim of the benchmark challenge is to design a controller to work with the given SIMULINK model as 'the plant' to satisfy the control performance. The control specification includes sink pressure step and sinusoidal disturbance tests (at the three different operating points), load ramp change from 50% to 100%, and coal quality change by $\pm 18\%$. The specifications of these tests are given in details in [20].

# 4 Internal model description

In this work, the original linear MPC design [23] is extended to include some of the plant nonlinearities by developing a static nonlinear model in the form of Wiener configuration as shown in Figure 1. Linear static gains are used for three outputs, CVGAS, MASS, TGAS, while, a feedforward neural network model is created for the fourth output PGAS. The output selection is based on the open-loop step response (within the prediction horizon length) comparison between the linear and nonlinear simulation model (see Figure 4). The results showed that the linear model can almost correctly capture the dynamic behaviour in three of the four outputs for up to 20 s (the prediction horizon length) under all load conditions. However, the fourth output PGAS exhibits salient nonlinearities which cannot be predicted by the linear model. It is also observed that the effect of the unmeasured disturbance PSINK on the output variable PGAS is quite large, whilst the time constant of the response is very short compared to that of other outputs.

Assuming that the plant considered has manipulable input, $\tilde{\boldsymbol{u}} \in \mathbb{R}^4$ and measured output, $\tilde{\boldsymbol{y}} \in \mathbb{R}^4$, which have steady-state values, $\tilde{\boldsymbol{u}}_0$ and $\tilde{\boldsymbol{y}}_0$ at the nominal operating point respectively. Around the operating point, the dynamic behaviour of the plant can be approximated by the following partially nonlinear discrete-time

state-space equations:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}_L(k) &= \boldsymbol{C}_L\boldsymbol{x}(k) \\ y_{NL}(k) &= f_{NN}(\boldsymbol{x}(k)) \\ \boldsymbol{y}(k) &= \begin{bmatrix} \boldsymbol{y}_L^T(k) & y_{NL}(k) \end{bmatrix}^T \end{aligned} \tag{1}$$

where, $k$, stands for $k$th sampling time, $\boldsymbol{u}(k) = \tilde{\boldsymbol{u}}(k) - \tilde{\boldsymbol{u}}_0$, and $\boldsymbol{y}(k) = \tilde{\boldsymbol{y}}(k) - \tilde{\boldsymbol{y}}_0$, are deviation variables, and $\boldsymbol{x}(k)$, is the internal state of the model. Outputs are divided into two groups: $\boldsymbol{y}_L$(k) outputs vector corresponding to the linear variables CVGAS, MASS and TGAS, and $y_{NL}(k)$ corresponding to nonlinear output, PGAS. The matrix $\boldsymbol{C}_L$ represents the linear static gain, while $f_{NN}$ is the nonlinear function modelled by a neural network. Of course, other forms of nonlinear function may be used as the static gain. The choice of neural network here is motivated by its ability to model any nonlinear function to any desired accuracy [10, 25].

Initially, the plant is assumed to be at the nominal operating point with $\boldsymbol{x}(0) = \boldsymbol{0}$, $\boldsymbol{u}(0) = \boldsymbol{0}$, $\boldsymbol{y}(0) = \boldsymbol{0}$. The matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}_L$ are obtained by linearising the nonlinear plant model at 0% load condition. The neural network model consists of two hidden layers and one output layer. A FFNN with single layer is usually sufficient to capture the nonlinearity of the model for most applications. However, it was found that, for the gasifier case, a network with two layers was more capable to model the plant than a single layer network (see section 6.1 for more details). The transfer function of the hidden layers is a ***sigmoid–tanh*** nonlinear function while a ***linear*** transfer function is used for the output layer. The mathematical

form of the function $f_{NN}$ can be represented in a vector form as :

$$\begin{aligned} \boldsymbol{O}_1 &= \sigma_s(\boldsymbol{W}_1\boldsymbol{x}(k) + \boldsymbol{b}_1) \\ \boldsymbol{O}_2 &= \sigma_s(\boldsymbol{W}_2\boldsymbol{O}_1 + \boldsymbol{b}_2) \\ y_{NL} &= \boldsymbol{W}_3\boldsymbol{O}_2 + b_3 \end{aligned} \tag{2}$$

where $\boldsymbol{O}_1$, $\boldsymbol{O}_2$ and $y_{NL}$ are the output values of each layer. The values $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, and $\boldsymbol{W}_3$ are the weight parameters while $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, and $b_3$ are the bias parameters. The function $\sigma_s(\cdot)$ is the ***sigmoid–tanh*** function which is defined as,

$$\sigma_s(n) = \frac{2}{1 + e^{-2n}} - 1 \tag{3}$$

Because the model in (1) is nonlinear, the convexity of the optimisation problem is not guaranteed. In order to use efficient QP algorithm to solve the online optimisation problem, local linearisation of the static FFNN model around the current states is performed. Future predictions of output based on current measurement $y_{NL}(k)$ can be approximated by the first two terms of the Taylor series expansion:

$$y_{NL}(k+i) \cong n_k + \boldsymbol{C}_{NL}\boldsymbol{x}(k+i) \quad \text{for } i = 1, \ldots, P \tag{4}$$

where,

$$n_k = f_{NN}(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}(k)} - \boldsymbol{C}_{NL}\boldsymbol{x}(k) \tag{5}$$

$$\boldsymbol{C}_{NL} = \frac{\partial f_{NN}(\boldsymbol{x})}{\partial \boldsymbol{x}}|_{\boldsymbol{x}=\boldsymbol{x}(k)} \tag{6}$$

The value of the function $f_{NN}(\boldsymbol{x})$ and the partial derivative $\partial f_{NN}(\boldsymbol{x})/\partial \boldsymbol{x}$ can be efficiently calculated from the neural network structure in equation 2 using the

chain rule as:

$$\frac{\partial f_{NN}}{\partial \boldsymbol{x}} = \boldsymbol{W}_3 \operatorname{diag}(\boldsymbol{I} - \boldsymbol{O}_2\boldsymbol{O}_2^T)\boldsymbol{W}_2 \operatorname{diag}(\boldsymbol{I} - \boldsymbol{O}_1\boldsymbol{O}_1^T)\boldsymbol{W}_1 \tag{7}$$

This results in a time-varying linear state-space form to be used in the predictive controller. Note that, $\boldsymbol{C}_{NL}$ is treated as constant over the entire prediction horizon for an optimisation problem to be solved at each sampling time and updated only when new plant measurement available.

The static gain inversion method used by many researchers [15, 18, 7, 16] to linearised the Wiener model is not applicable here as the model is partially nonlinear. To use this method the linear static gain (*i.e.* matrix $\boldsymbol{C}_L$) should be invertible which is not the case since $\boldsymbol{C}_L$ is not square. On the other hand, the inverse neural network should map one output to 16 states. Training such a network is not trivial. For simplicity, the sequential linearisation approach is adopted here.

# 5 Predictive control formulation

The prediction model to be used can be represented by the following state-space equation:

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \\ \boldsymbol{y}(k) &= \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{d}(k) \end{aligned} \tag{8}$$

where, $\boldsymbol{y}(k) = [\boldsymbol{y}_L^T(k)\ \ y_{NL}(k)]^T$, $\boldsymbol{C} = [\boldsymbol{C}_L^T\ \ \boldsymbol{C}_{NL}^T]^T$, and $\boldsymbol{d}(k)$ is the virtual output disturbance estimated from the outputs measurement to reduce the plant-model mismatch. Note, the term $n_k$ in equation (4) is absorbed into $\boldsymbol{d}(k)$. At the kth sampling time, with currently measured outputs, $\boldsymbol{y}_m = \tilde{\boldsymbol{y}}_m(k) - \tilde{\boldsymbol{y}}_0$, and

the current states $\boldsymbol{x}(k)$, the future output within the prediction horizon, $P$ can be estimated from the future input $\boldsymbol{u}(k)$ to be determined within the moving horizon, $M$ as follows. Taking

$$\boldsymbol{d}(k+i) = \boldsymbol{d}_k = \boldsymbol{y}_m(k) - \boldsymbol{C}\boldsymbol{x}(k), \quad \text{for} \quad i = 1, \ldots, P \tag{9}$$

then

$$\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{U} + \boldsymbol{\Psi}\boldsymbol{x}(k) + \boldsymbol{L}\boldsymbol{d}_k \tag{10}$$

where

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}^T(k+1) & \cdots & \boldsymbol{y}^T(k+P) \end{bmatrix}^T$$

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}^T(k) & \cdots & \boldsymbol{u}^T(k+M-1) \end{bmatrix}^T$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{CB} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{CA}^{P-1}\boldsymbol{B} & \boldsymbol{CA}^{P-2}\boldsymbol{B} & \cdots & \sum_{i=M}^{P} \boldsymbol{CA}^{P-i}\boldsymbol{B} \end{bmatrix}$$

$$\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{CA} \\ \vdots \\ \boldsymbol{CA}^P \end{bmatrix}$$

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \end{bmatrix}^T$$

Future input, $\boldsymbol{U}$ is determined to follow the output reference, $\boldsymbol{y}_r(k)$, and the input reference $\boldsymbol{u}_r(k) = \boldsymbol{H}_0^{-1}(\boldsymbol{y}_r(k) - \boldsymbol{d}_k)$, where $\boldsymbol{H}_0 = \boldsymbol{C}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}$. Define input

and output reference vectors as

$$\boldsymbol{Y}_r = \begin{bmatrix} \boldsymbol{y}_r^T(k+1) & \cdots & \boldsymbol{y}_r^T(k+P) \end{bmatrix}^T$$
$$\boldsymbol{U}_r = \begin{bmatrix} \boldsymbol{u}_r^T(k) & \cdots & \boldsymbol{u}_r^T(k+M-1) \end{bmatrix}^T$$

Then, $\boldsymbol{U}_r = \boldsymbol{H}(\boldsymbol{Y}_r - \boldsymbol{L}\boldsymbol{d}_k)$, where

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_0^{-1} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_0^{-1} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{H}_0^{-1} & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix}$$

The optimisation problem is to minimize the performance cost:

$$\begin{aligned} J &= 0.5(\boldsymbol{Y} - \boldsymbol{Y}_r)^T \boldsymbol{Q}(\boldsymbol{Y} - \boldsymbol{Y}_r) + 0.5(\boldsymbol{U} - \boldsymbol{U}_r)^T \boldsymbol{R}(\boldsymbol{U} - \boldsymbol{U}_r) \quad (11) \\ \text{s.t.} \quad & \underline{\boldsymbol{u}} \leq \boldsymbol{u} \leq \overline{\boldsymbol{u}} \\ & |\boldsymbol{u}(k+1) - \boldsymbol{u}(k)| \leq \boldsymbol{\delta}_u \end{aligned}$$

where, output and input weighting matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive definite and $\underline{\boldsymbol{u}}$, $\overline{\boldsymbol{u}}$ and $\boldsymbol{\delta}_u$ are the lower, upper and maximum rate bounds of the input respectively.

Using the predictive equation (10), the optimisation problem is equivalent to a standard QP problem:

$$\begin{aligned} J &= 0.5\boldsymbol{U}^T \boldsymbol{S}\boldsymbol{U} + \boldsymbol{U}^T(\boldsymbol{X}_1 \boldsymbol{x}(k) - \boldsymbol{X}_2(\boldsymbol{Y}_r - \boldsymbol{L}\boldsymbol{d}_k)) \\ \text{s.t.} \quad & \boldsymbol{U} \leq \overline{\boldsymbol{U}} \quad (12) \\ & -\boldsymbol{U} \leq -\underline{\boldsymbol{U}} \\ & \boldsymbol{E}\boldsymbol{U} \leq \boldsymbol{\Delta}_u + \boldsymbol{F}\boldsymbol{u}(k-1) \\ & -\boldsymbol{E}\boldsymbol{U} \leq \boldsymbol{\Delta}_u - \boldsymbol{F}\boldsymbol{u}(k-1) \end{aligned}$$

where, $\boldsymbol{u}(k-1)$ is the previous input, and other variables are defined as follows:

$$
\begin{aligned}
\boldsymbol{S} &= \boldsymbol{\Phi}^T \boldsymbol{Q} \boldsymbol{\Phi} + \boldsymbol{R} \\
\boldsymbol{X}_1 &= \boldsymbol{\Phi}^T \boldsymbol{Q} \boldsymbol{\Psi} \\
\boldsymbol{X}_2 &= \boldsymbol{\Phi}^T \boldsymbol{Q} + \boldsymbol{R}\boldsymbol{H} \\
\overline{\boldsymbol{U}} &= \begin{bmatrix} \overline{\boldsymbol{u}}^T & \cdots & \overline{\boldsymbol{u}}^T \end{bmatrix}^T \\
\underline{\boldsymbol{U}} &= \begin{bmatrix} \underline{\boldsymbol{u}}^T & \cdots & \underline{\boldsymbol{u}}^T \end{bmatrix}^T \\
\boldsymbol{\Delta}_u &= \begin{bmatrix} \boldsymbol{\delta}_u^T & \cdots & \boldsymbol{\delta}_u^T \end{bmatrix}^T \\
\boldsymbol{E} &= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{I} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & -\boldsymbol{I} & \boldsymbol{I} \end{bmatrix} \\
\boldsymbol{F} &= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix}^T
\end{aligned}
$$

Note, in the above formulation, output constraints are neglected to simplify the algorithm and to fully use the plant capability. The QP problem (12) is efficiently solvable by off-the-shelf software. The only tunable parameters in the above formulation are $\boldsymbol{Q}$, $\boldsymbol{R}$, $P$, $M$ and the sampling time. Thus, the control strategy can be easily implemented and tuned to satisfy required performance.

In vector $\boldsymbol{U}$, only the first $n_u$ rows, corresponds to $\boldsymbol{u}(k)$ are applied to the plant. The whole procedure is repeated at the next sampling instance.

It should be noted here, the initial state vector required to solve the optimisation problem at every sampling time is estimated by a state updated using the linear dynamic part of model (8). The remain estimation error is further corrected via the output disturbance model $d_k$ where in the model nonlinearity is included (see

equation (9)). Note, this is a standard treatment in offset-free MPC and has been recognised as a sufficient alternative of traditional state estimation approaches in MPC [26].

For the unconstrained case, the optimal solution, corresponding to a state feedback control law, can be obtained analytically:

$$\boldsymbol{U} = -\boldsymbol{K}_1\boldsymbol{x}(k) + \boldsymbol{K}_2(\boldsymbol{y}_r - \boldsymbol{d}_k) \tag{13}$$

where $\boldsymbol{K}_1 = \boldsymbol{S}^{-1}\boldsymbol{X}_1$ and $\boldsymbol{K}_2 = \boldsymbol{S}^{-1}\boldsymbol{X}_2$. Let $\boldsymbol{K}$ be the first $n_u$ rows of $\boldsymbol{K}_1$, then the nominal stability (perfect model without input saturation) of the closed-loop can be checked by calculating the eigenvalue of the matrix, $\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}$.

# 6 Gasifier control using NMPC

## 6.1 Nonlinear system identification

The first step to implement the above algorithm is to get an internal model in the form of (8). Three operating conditions are specified in the gasifier benchmark problem: 0%, 50%, and 100% load conditions. The performance requirements at 50% and 100% load conditions were found relatively easier to achieve. It was then decided to use the 0% load point as the nominal point to get the linearised state space model. The resulting model was then reduced to 16 states via pole-zero cancellation (using Control System Toolbox functions, `ssbal` and `minreal`). The 16 states model is discretized with the sampling time selected.

For the FFNN static model of PGAS, the number of nodes in the first or second hidden layers was 10 with one node in the output layer. Note for comparison, a FFNN with a single hidden layer and 18 hidden nodes is also trained to capture the static nonlinearity of PGAS. Data fitting using the network with two hidden layers was considerably better than use the single layer network. In addition, the number

of parameters need to be trained (weights and biases) in the two-layers network (*i.e.* $(10\times16+10)+(10\times10+10)+11 = 291$) is smaller than the number of parameters for the single layer network, which is $(16 \times 18 + 18) + 19 = 325$). Therefore, the two-layer network was selected as the internal model. Training, validating and testing data were generated through applying a sequence of zero mean normalised random pulses to input channels. The periods and amplitudes of these pulses vary according to their expected maximum and minimum variations under different load conditions. Data sets over different loads were then linked together and used in training and validation of the FFNN. The performance of the trained PGAS model at 0% load is shown in Figure 4, while Figure 5 shows the output pressure response at the three load conditions, where a significant improvement in model accuracy is observed.

### 6.2 Predictive controller design

Normally, the sampling time should be less than one tenth of $2\pi/\omega_b$, where $\omega_b$ is the required bandwidth of the closed-loop. The benchmark requires to reject a sinusoidal disturbance with a period of 25 seconds (0.04 Hz). Therefore, the sampling time should be less than 2.5 seconds. Further, the sampling time should not be too large so that in step disturbance tests, the output variables do not deviate from setpoints more than the specified constraints before the controller can start to response. Several open-loop tests for a step disturbance of PSINK at three load conditions were performed. The outputs response results are shown in Figure 6. The results show that, the worst response case is the 0% condition, where, without control, the output PGAS is within specified range for only 1.2 seconds. Hence, the sampling time is selected as 1 second. This satisfies the requirements of both disturbance tests.

The predictive controller is implemented in MATLAB as a SIMULINK s-

function to replace the control block in the nonlinear simulation model provided in the benchmark suit. The QP problem is solved by calling `quadprog` of the Optimisation Toolbox at each sampling time. This is the major computation burden in the above algorithm and is solely determined by the control horizon, $M$. The prediction horizon, $P$ has little effect on computation time, thus can be selected relatively large to benefit stability.

To tune $M$ and $P$, initially let $P = M$. By varying $M$ from 1 s to 12 s, a stable performance is obtained which satisfies all control specifications for $5$ s $\leq M \leq$ $10$ s. When $M \geq 10$ s, the improvement on the system performance is negligible but computation time increases significantly. Therefor $M = 7$ s is selected, which gives a good performance in all tests. To choose a suitable prediction horizon $P$, a reasonable range from the minimum value ($P = M = 7$ s) to $P = 25$ s has been tested. A stable response without any constraint violation is found within the range $15$ s $\leq P \leq 20$ s. No performance improvement can be observed when $P \geq 20$ s. Therefore $P = 20$ s (the maximum value of the range) is chosen to ensure that both the system stability and satisfactory control performance achieved within a reasonable computation time.

The weighting matrix, $\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{Q}_0, \cdots, \boldsymbol{Q}_0)$, where $\boldsymbol{Q}_0$ is diagonal and initially set to the inverse of the output error bounds. After online tuning, the final values are:

$$\boldsymbol{Q}_0 = \begin{bmatrix} 0.15 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 2.1 & 0 \\ 0 & 0 & 0 & 2 \times 10^6 \end{bmatrix} \tag{14}$$

Also, the input weighting matrix $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{R}_0, \cdots, \boldsymbol{R}_0)$, where $\boldsymbol{R}_0$ is diagonal

and set to the following value after online tuning:

$$\boldsymbol{R}_0 = \begin{bmatrix} 10^5 & 0 & 0 & 0 \\ 0 & 5 \times 10^3 & 0 & 0 \\ 0 & 0 & 5 \times 10^3 & 0 \\ 0 & 0 & 0 & 10^4 \end{bmatrix} \tag{15}$$

Using the above configuration, nominal stability is achieved at all three load conditions, *i.e.* the magnitudes of all eigenvalues of $\boldsymbol{A}_i - \boldsymbol{B}_i\boldsymbol{K}$ are less than 1. Where, $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are the discrete states and control matrices at different load conditions.

One of the advantages of MPC is that future setpoint change information is incorporable into the QP optimisation problem to improve setpoint tracking performance. This is implemented in the gasifier controller.

# 7 Simulation Results

## 7.1 Disturbance Tests

The following two disturbance tests are performed for three load conditions for 300 seconds:

1. step change in sink pressure of $-0.2$ bar at 30s;
2. 0.04 Hz sinusoidal variation in sink pressure of amplitude 0.2 bar beginning at 30s.

All the results to follow are compared with the linear MPC. The maximum and minimum values as well as the peak rate change of the input variables of two disturbance tests under different load conditions are shown in Table 4. The maximum absolute error between output variables and their setpoints and the integral of absolute error (IAE) of these variables are calculated in Table 3. To quantify

the performance improvement, the IAE ratio, defined as IAE(NMPC)/IAE(LMPC) is calculated for all output variables. The mean value of these ratios is given at the bottom of Table 3. These ratios indicate that the most significant improvement are relating to the step tests for all three load conditions and sinusoidal test for 0% load condition. Response for step test under 50% load condition and sinusoidal test under 0% load condition are shown in Figures 7 and 8, respectively. In the step disturbance test, the results are plotted for $t \leq 100$s to present the control performance in more details. After this time period, all the outputs response remained constant. The results in Table 4 and 3 however are calculated until t=300 s as required in the original challenge issue.

For 0% load sinusoidal test, results with extra simulation time (until t=600 s) are provided to confirm the satisfactory performance of output in meeting the given specifications. The results show that both linear and nonlinear controllers are capable of maintaining the output variables within the limits for the tests specified by ALSTOM.

The mean IAE ratio given in Table 3 shows that the system performance, comparing with LMPC results, is significantly improved (overall IAE index is reduced by more than 10%) by using the NMPC. In fact, most of the improvement happened during the step disturbance test. In the sinusoidal disturbance test a little improvement in the system response is observed. In addition, the linear controller was the best at 50% and 100% load conditions. This result can be explained as follow. A Wiener model mainly captures static nonlinearity of the system by introducing a static nonlinear gain. In the step disturbance test, signal change of the system is relatively slow at the most of time. Therefore, the nonlinearity associated with such a dynamic mode can be well captured by the identified Wiener model. This leads to the performance improvement in all load conditions. However, in the sinusoidal disturbance test, signals fluctuate all the time and relatively faster than

in the step test. Nonlinearity associated with such dynamic behaviour is relatively difficult to be captured by the Wiener model, hence it results some sort of performance deterioration in these test particularly for 50% and 100% load conditions, where the model mismatch is relatively large because the linear model is based on the 0% load condition. To further demonstrate this, extra simulation is performed by reducing the frequency of the sinusoidal disturbance to the half of the specified value. A performance improvement in all load conditions is obtained using the NMPC approach (the results are not included here to save space).

Also, due to the strong interactions between the system variables, the improvement in other output variables sometimes is even larger than that in PGAS itself (see figures 7 and 8). This is explained as follows. The response of PGAS, particularly to disturbance PSINK is much faster than other output variables (Figure 6). The improvement of nonlinear model is mainly in long term prediction (Figure 4). Hence, it has more effect on slow-response variables than on PGAS, which is a fast-response variable. Moreover, the maximum drop of PGAS in the step disturbance test is a response to the disturbance before the controller can take any action, hence is not able to be reduced by changing internal model only.

The partially nonlinear model does not only improve the output performance, it also results smaller excursion behaviour in NMPC than that in LMPC as shown in figures 7 and 8.

### 7.2 Load Change Test

In this test, the load is required to increase from 50% to 100% within time from 100 s to 700 s. The actual response is collected from the simulation and compared with the results when using LMPC controller. For both controllers, good setpoint tracking performance is obtained. The outputs results in Figure 9 show approximately similar behaviours for the two controllers, with a small improvement in

the bed–mass response when using the NMPC approach. It can be seen that with the exception of the bed–mass the outputs track their demanded levels reasonably well. It takes significant time (beyond the length of simulation shown) for the bed–mass to return to the setpoint. This is due to the saturation of the coal feed input ( Figure 9 (g)). Note that, the coal flow-rate saturation on its upper limit is not avoidable as explained by other researchers [24, 27, 28]. Therefore, the achievable improvement in the bed mass response is limited due to the inherent characteristics of the process. However, the manipulated variables response appears smoother in this test as shown in (e)–(h) of Figure 9.

# 8 Conclusion

A nonlinear predictive controller has been developed to control the ALSTOM gasifier benchmark process. The LMPC approach employing GPC strategy of [23] is modified to use a partially nonlinear model as the internal model. A nonlinear Wiener model is used to identify one of the process output variables (PGAS) which has strong nonlinearity while a linear model at 0% load condition is adopted for the other output variables. A multi-layer feedforward neural network is used as the nonlinear static element of the Wiener model. To regain the convex feature of the QP optimisation problem, the nonlinear static gain of PGAS model is linearised at every sampling time to update the linear model for each new optimisation problem. Thus, the resulting internal model is a linear time-varying model. The new controller meets all the required performance specifications within given input and output constraints during sink pressure disturbance and load change tests and the results show a significant improvement in the system performance compared with the results obtained when only linear time-invariant model is used.

The proposed method is useful if only part of a MIMO system exhibits a strong

nonlinearity. In this case, only this part need to be modelled using a nonlinear Wiener model. This will reduce the efforts and time to identified a much more complicated MIMO nonlinear model which may not be necessary for the overall system.

## References


[1] Pearson R. K., "Selecting nonlinear model structures for computer control: review", *Journal of Process Control*, vol. 13, pp. 1-26, 2003.

[2] Piche S., Sayyar-Rodsari B., Johnson D., and Gerules M., "Nonlinear model predictive control using neural networks". *IEEE Control System Magazine*, vol. 20(3), pp. 53–62, 2000.

[3] Maciejowski J. M., "Predictive control with constraints", *Prentice Hall*, Harlow, England, 2002.

[4] Khadir M. T. and Ringwood J. V., "Linear and nonlinear model predictive control design for a milk pasteurization plant", *Control and Intelligent Systems*, vol. 31, pp. 1–8, 2003.

[5] Gattu G. and Zafiriou E., "Nonlinear quadratic dynamic matrix control with state estimation", *Indus. Eng. Chem. Res.*, vol. 31, pp. 1096–1104, 1992.

[6] Huzmezan M. and Maciejowski J. M., "Reconfiguration and scheduling in flight using quasi–LPV high–fidelity models and MBPC control", *In Proc. American Control Conference*, Philadelphia, 1998.

[7] Norquay S. J., Palazoglu A., and Romagnoli J. A., "Model predictive control based on Wiener models", *Chemical Engineering Science*, vol. 53(1), pp. 75-84, 1998.

[8] Brengle D. D. and W. D. Seider., “Multistep nonlinear predictive controller”, *Ind. Eng. Chem. Res.*, vol. 28, pp. 1812–1822, 1989.

[9] Patwardhan A. A. and Edgar T. F., “Nonlinear model predictive control of packed distillation column”, *Ind. Eng. Chem. Res.*, vol. 32, pp. 2345–2356, 1993.

[10] Su H. T. and McAvoy (1997), “Artificial neural networks for nonlinear process identification and control”, In: Henson, M. A. & Seborg, D.E. (Eds). “Nonlinear Process Control”, Chap. 7, pp. 371–428, Englewood Cliffs, NJ: Prentice–Hall.

[11] Sentoni G. B., Biegler L. T., Guiver J. B., and Zhao H., “State–space nonlinear process modelling”, *AIChE J.*, vol. 44(10), pp. 2229–2239, 1998.

[12] Maner B. R., Doyle F. J., Ogunnaike B. A., and Pearson R. K., “Nonlinear model predictive control of a simulated multivariable polymerization reactor using second–order Volterra models”, *Automatica*, vol. 32, pp. 1285–1301, 1996.

[13] Sriniwas G. R. and Arkun Y. , “A global solution to the nonlinear predictive control algorithms using polynomial ARX models”, *Comput. Chem. Eng.*, vol. 21, pp. 431–439, 1997.

[14] Dumont G., Fu Y., and Lu G., ”Nonlinear adaptive generalized predictive control and applications ”, *Advances in model-based predictive control, Oxford University Press, Oxford*, 1994.

[15] Al–Duwaish H., Karim M. N., and Chandrasekar V., “Use of multilayer feedforward neural network in identification and control of Wiener model”, *IEE Proc–Control Theory Appl.*, vol. 143(3), pp. 255–258, 1996.

[16] Cervantes A. L.,Agamennoni O. E., and Figueroa J. L., ”A nonlinear model predictive control system based on Wiener piecewise linear models”, *Journal of Process Control*, vol. 13, pp. 655–666, 2003.

[17] Zhao H., Guiver J., and Sentoni G., “An identification approach to nonlinear state space model for industrial multivariable model predictive control”, *Proceeding of the American Control Conference, Philadelphia, Pennsylvania*, 1998.

[18] Gerksic S., Juricic D., Strmcnik S., and Matko D., “Wiener model based nonlinear predictive control”, *International Journal of Systems Science*, vol. 31(2), pp. 189–202, 2000.

[19] R. Dixon, “Advanced gasifier control,” *IEE Computing and Control Engineering Journal*, vol. 10, no. 3, pp. 93–96, 1999.

[20] R. Dixon, A. Pike, and M. Donne, “The ALSTOM benchmark challenge on gasifier control,” *Proc. IMechE Part I, Journal of Systems and Control Engineering*, vol. 214, pp. 389–394, 2000.

[21] M. Rice, J. Rossiter, and J. Schurmans, “An advanced predictive control approach to the ALSTOM gasifier problem,” *Proc. IMechE Part I, Journal of Systems and Control Engineering*, vol. 214, pp. 405–413, 2000.

[22] R. Dixon, “Alstom benchmark challenge ii: Control of a nonlinear gasifier model,” available from http://www.iee.org/OnComms/PN/controlauto/Specification_v2.pdf, 2002.

[23] Al Seyab R. K., Cao Y., and Yang S. H., ”A case study of predictive control for the ALSTOM gasifier problem”. *Control 2004, University of Bath*, UK, 2004.

[24] R. Dixon and A. W. Pike, “Introduction to the $2^{ND}$ ALSTOM benchmark challenge on gasifier control ”, *Control 2004, University of Bath*, UK, 2004.

[25] Behat N. and McAvoy, “Use of neural nets for dynamic modeling and control of chemical process systems”, *Computers Chem. Engng.*, vol. 14(4-5), pp. 573–582, 1990.

[26] Seki H., Ogawa M., Ooyama S., Akamatsu K., Ohshima M., and Yang W., “Industrial application of a nonlinear model predictive control to polymerization reactors”, *Control Engineering Practice*, vol. 9, pp. 819–828, 2001.

[27] Farag A. and Werner H., “Multivariable PID–controller design for a gasifier plant using penality–based–multi–objective GA”, *Control 2004, University of Bath*, UK, 2004.

[28] Simm A. and Liu G. P.,“Improving the performance of the ALSTOM baseline controller using multiobjective optimisation”, *Control 2004, University of Bath*, UK, 2004.

# List of Figures

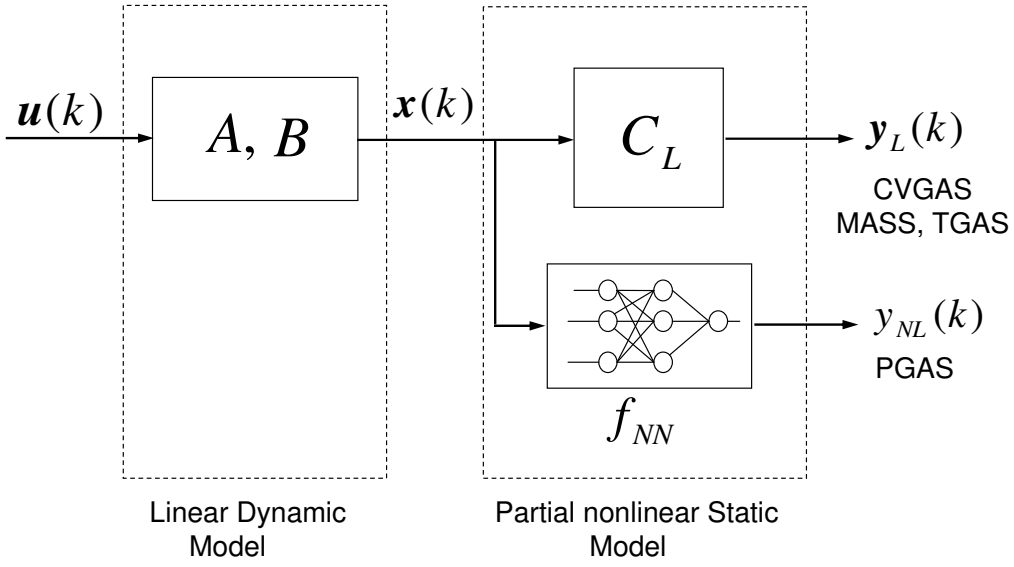


Figure 1: Nonlinear internal model.

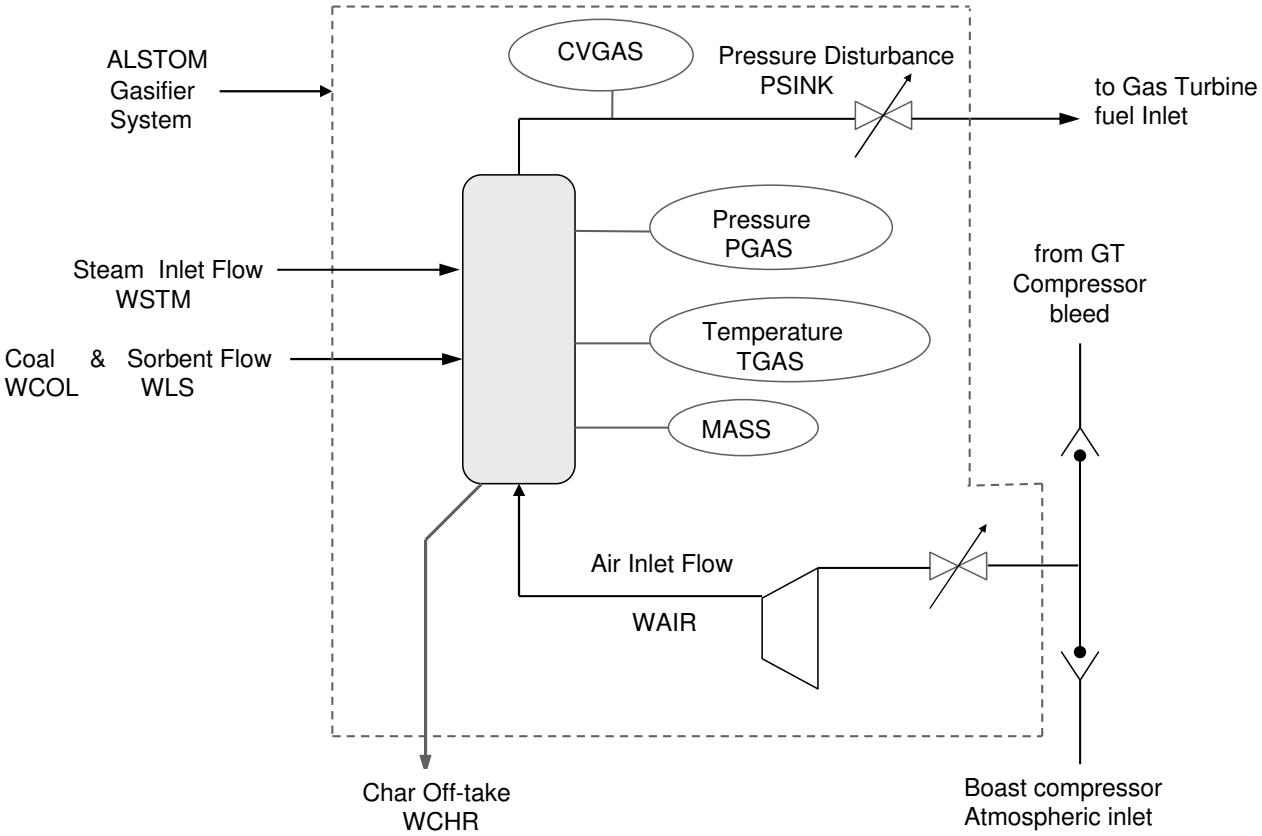


Figure 2: Gasifier Schematic [24].

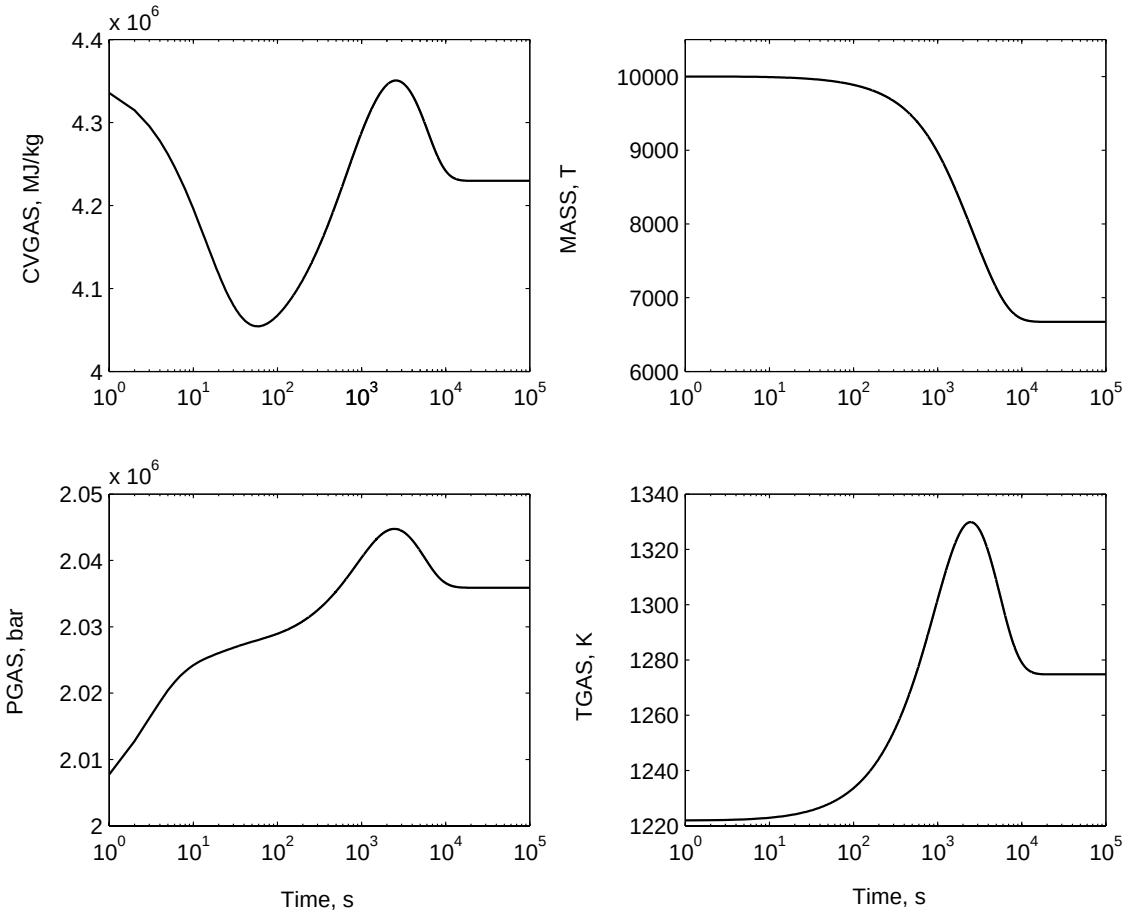


Figure 3: Open–loop response of the ALSTOM gasifier for a +20% step change in the air flowrate .

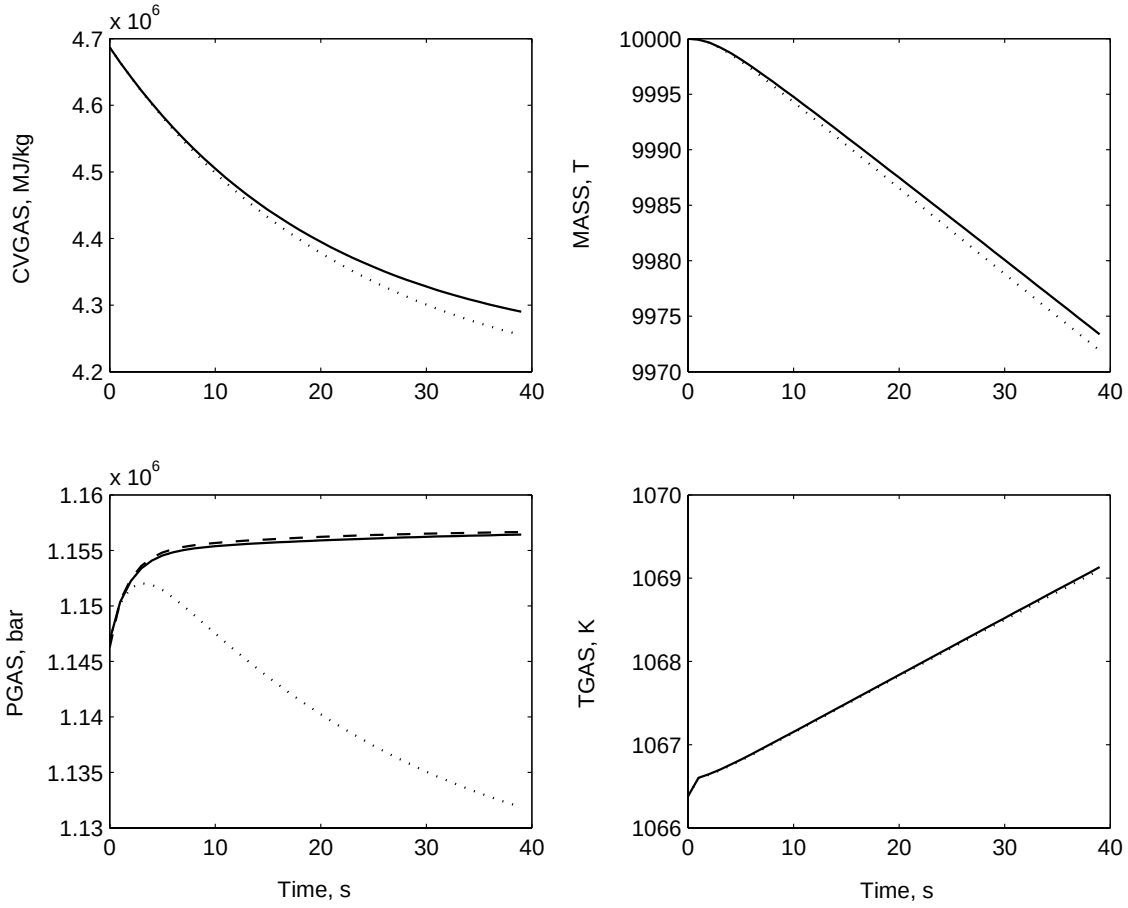


Figure 4: Open-loop response of gas pressure at 0% load condition to a step change in WAIR. Plant output (solid), Wiener model (dashed), linear model (dotted).

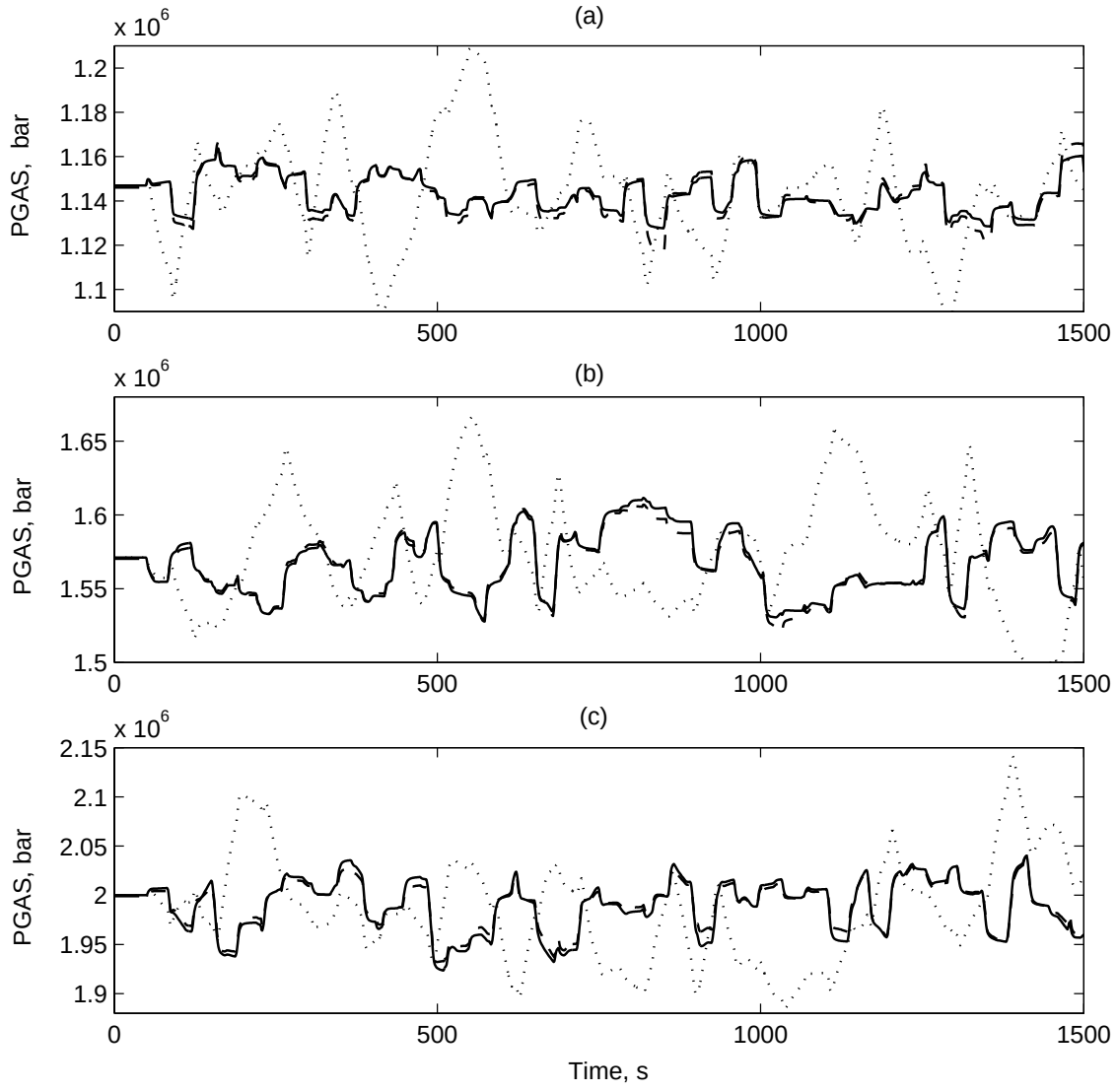


Figure 5: Validation data, plant output (solid), Wiener model (dashed), linear model (dotted) : (a) 0% load, (b) 50% load, (c) 100% load.

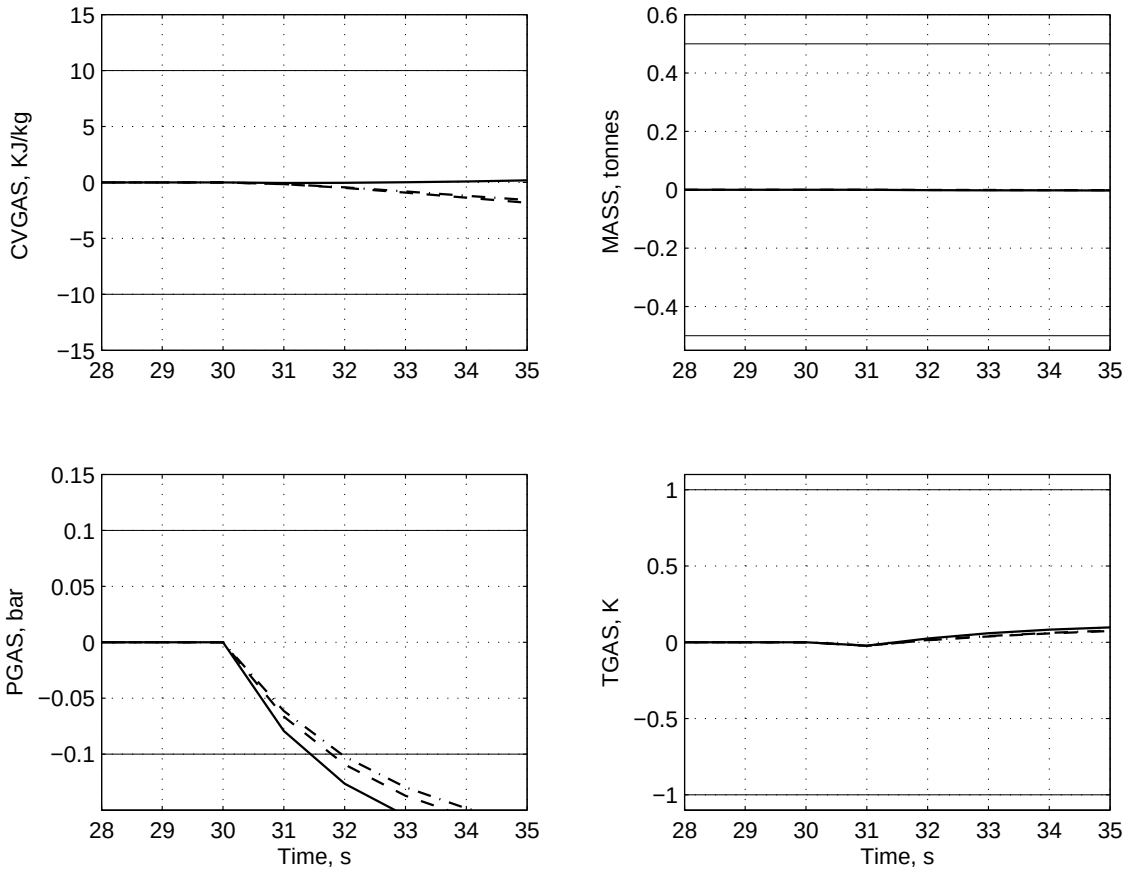


Figure 6: Open-loop output response to a step disturbance at 30s at 0% (solid), 50% (dashed) and 100% (dash-dotted) load conditions.

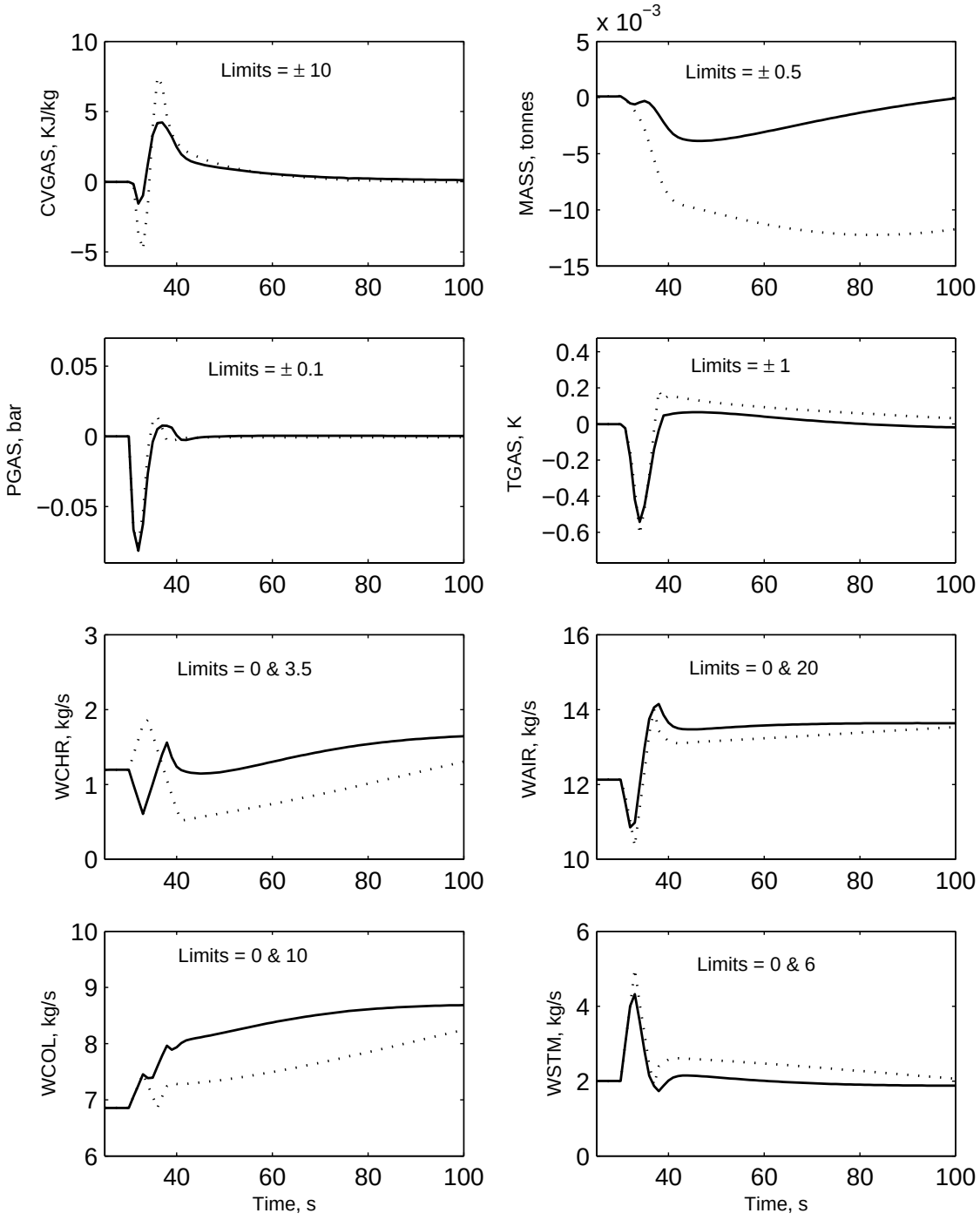


Figure 7: The gasifier response at 50% load condition, NMPC (solid), LMPC (dotted).

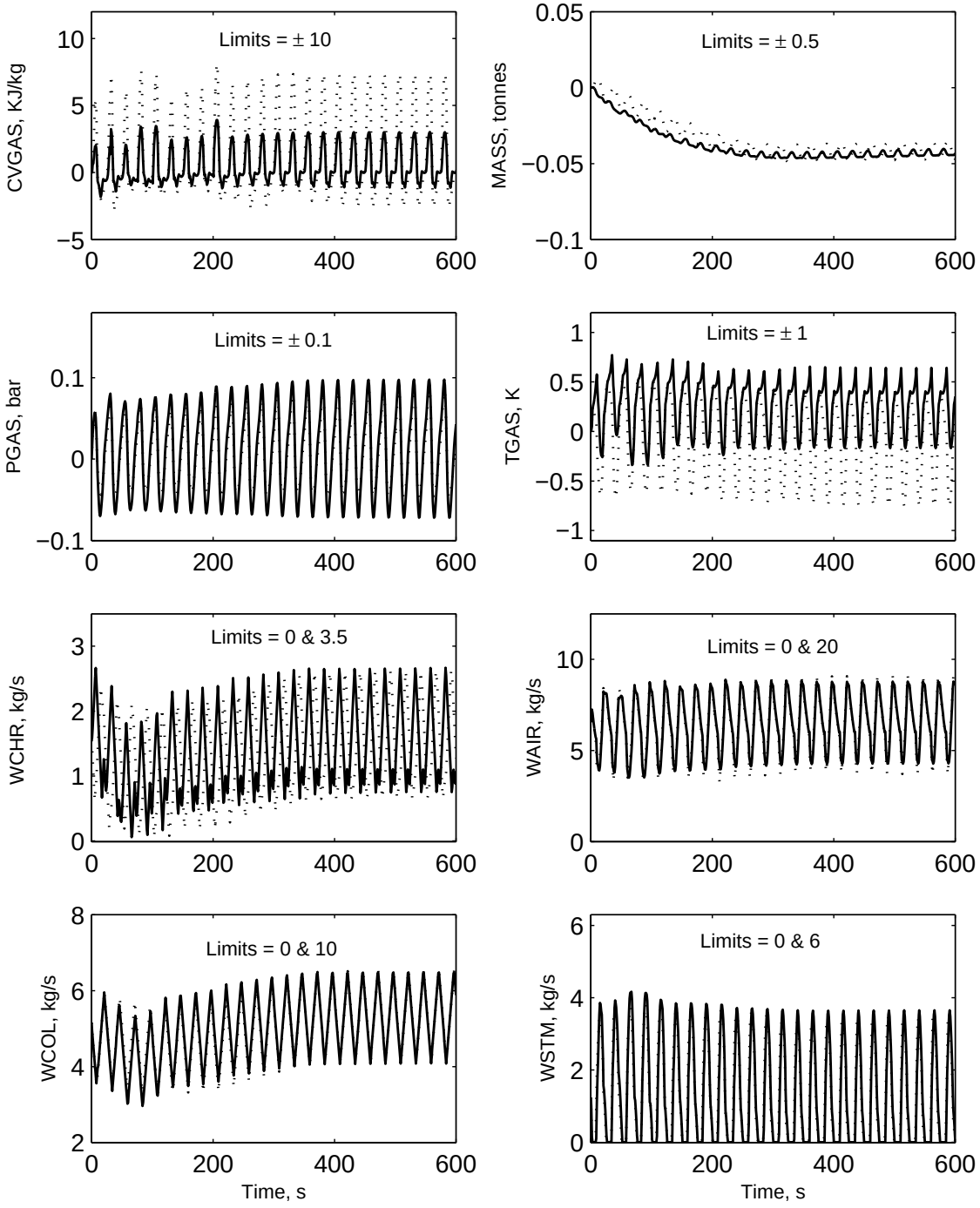


Figure 8: The gasifier response at 0% load condition, NMPC (solid), LMPC (dotted).

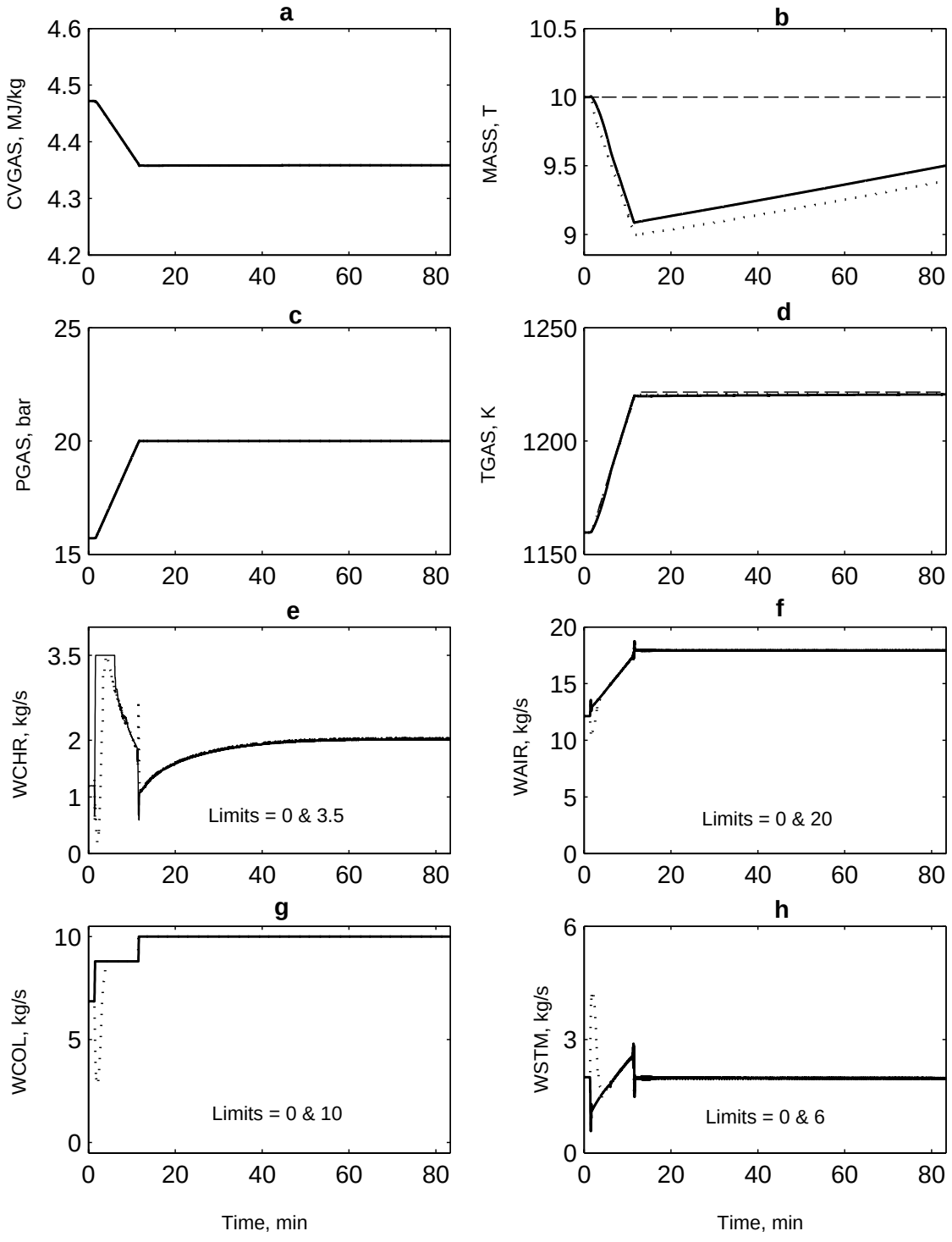


Figure 9: The gasifier response at setpoint ramp test, NMPC (solid), setpoint (dashed), LMPC (dotted). (a)–(d) Outputs, (e)–(f) Inputs and limits.

Table 1: Outputs variables and limits

| Outputs | Description | Allowed fluctuations |
|---|---|---|
| CVGAS | Fuel gas calorific value | $\pm 0.01$ MJ/kg |
| MASS | Bed mass | $\pm 500$ kg |
| PGAS | Fuel gas pressure | $\pm 0.1$ bar |
| TGAS | Fuel gas temperature | $\pm 1$ K |

Table 2: Inputs variables and limits

| Inputs | Description | Maximum value | Peak rate |
|---|---|---|---|
| WCHR | Char extraction rate | 3.5 kg/s | 0.2 kg/s$^2$ |
| WAIR | Air flow rate | 20 kg/s | 1.0 kg/s$^2$ |
| WCOL | Coal flow rate | 10 kg/s | 0.2 kg/s$^2$ |
| WSTM | Steam flow rate | 6 kg/s | 1.0 kg/s$^2$ |

Table 3: Output results

| Step, 100% load | Maximum Absolute Error | | IAE | | |
|---|---|---|---|---|---|
| Output | LMPC | NMPC | LMPC | NMPC | Ratio |
| CVGAS | 7.33 | 5.69 | 96.275 | 84.986 | 0.8827 |
| MASS | 16.00 | 10.613 | 3148 | 1711 | 0.5435 |
| PGAS | 0.067 | 0.0735 | 0.262 | 0.266 | 1.0153 |
| TGAS | 0.5194 | 0.5193 | 12.619 | 10.842 | 0.8592 |
| Step, 50% load | Maximum Absolute Error | | IAE | | |
| Output | LMPC | NMPC | LMPC | NMPC | Ratio |
| CVGAS | 7.447 | 4.231 | 75.112 | 64.775 | 0.8624 |
| MASS | 12.42 | 4.054 | 1569 | 372.686 | 0.2089 |
| PGAS | 0.076 | 0.081 | 0.336 | 0.3128 | 0.9310 |
| TGAS | 0.611 | 0.542 | 10.813 | 9.819 | 0.9081 |
| Step, 0% load | Maximum Absolute Error | | IAE | | |
| Output | LMPC | NMPC | LMPC | NMPC | Ratio |
| CVGAS | 8.98 | 2.943 | 106.12 | 47.054 | 0.4434 |
| MASS | 29.26 | 8.090 | 5587 | 1505 | 0.2694 |
| PGAS | 0.0954 | 0.1006 | 0.471 | 0.449 | 0.9533 |
| TGAS | 0.525 | 0.6005 | 31.13 | 22.325 | 0.7172 |
| Sine, 100% load | Maximum Absolute Error | | IAE | | |
| Output | LMPC | NMPC | LMPC | NMPC | Ratio |
| CVGAS | 5.065 | 5.348 | 858.052 | 898.07 | 1.0466 |
| MASS | 3.588 | 3.1789 | 452.835 | 434.991 | 0.9606 |
| PGAS | 0.0315 | 0.0354 | 5.146 | 5.245 | 1.0192 |
| TGAS | 0.302 | 0.321 | 43.805 | 45.957 | 1.0491 |
| Sine, 50% load | Maximum Absolute Error | | IAE | | |
| Output | LMPC | NMPC | LMPC | NMPC | Ratio |
| CVGAS | 4.317 | 4.720 | 695.382 | 760.38 | 1.0935 |
| MASS | 6.663 | 6.191 | 948.47 | 1010.7 | 1.0656 |
| PGAS | 0.034 | 0.040 | 5.7903 | 6.417 | 1.1082 |
| TGAS | 0.349 | 0.440 | 57.327 | 66.176 | 1.1544 |
| Sine, 0% load | Maximum Absolute Error | | IAE | | |
| Output | LMPC | NMPC | LMPC | NMPC | Ratio |
| CVGAS | 8.0745 | 3.916 | 649.430 | 295.29 | 0.4547 |
| MASS | 48.642 | 46.207 | 7510 | 9292 | 1.2373 |
| PGAS | 0.0774 | 0.099 | 9.899 | 13.616 | 1.3755 |
| TGAS | 0.7138 | 0.7705 | 111.17 | 101.388 | 0.9120 |
| The Mean IAE Ratio | | | | | 0.8780 |

Table 4: Input results

| Step, 100% load | Maximum | | Minimum | | Peak Rate | |
|---|---|---|---|---|---|---|
| Input | LMPC | NMPC | LMPC | NMPC | LMPC | NMPC |
| WCHR | 1.6899 | 1.366 | 0.3075 | 0.582 | 0.2 | 0.2 |
| WAIR | 19.109 | 19.322 | 16.175 | 16.161 | 1.0 | 1.0 |
| WCOL | 10.00 | 10.00 | 8.62 | 8.616 | 0.2 | 0.2 |
| WSTM | 5.0702 | 4.744 | 2.492 | 2.347 | 1.0 | 1.0 |
| Step, 50% load | Maximum | | Minimum | | Peak Rate | |
| Input | LMPC | NMPC | LMPC | NMPC | LMPC | NMPC |
| WCHR | 1.9405 | 1.673 | 0.520 | 0.606 | 0.2 | 0.2 |
| WAIR | 14.004 | 14.151 | 10.356 | 10.856 | 1.0 | 1.0 |
| WCOL | 8.987 | 8.688 | 6.844 | 6.844 | 0.2 | 0.2 |
| WSTM | 4.997 | 4.322 | 1.719 | 1.733 | 1.0 | 1.0 |
| Step, 0% load | Maximum | | Minimum | | Peak Rate | |
| Input | LMPC | NMPC | LMPC | NMPC | LMPC | NMPC |
| WCHR | 2.153 | 1.947 | 0.227 | 0.962 | 0.2 | 0.2 |
| WAIR | 8.385 | 8.799 | 4.715 | 4.714 | 1.0 | 1.0 |
| WCOL | 7.558 | 7.207 | 5.157 | 5.156 | 0.2 | 0.2 |
| WSTM | 4.236 | 4.236 | 1.040 | 1.191 | 1.0 | 1.0 |
| Sine, 100% load | Maximum | | Minimum | | Peak Rate | |
| Input | LMPC | NMPC | LMPC | NMPC | LMPC | NMPC |
| WCHR | 1.3616 | 1.523 | 0.473 | 0.3908 | 0.174 | 0.1857 |
| WAIR | 18.902 | 18.933 | 15.75 | 15.735 | 0.569 | 0.625 |
| WCOL | 9.724 | 9.788 | 7.293 | 7.244 | 0.2 | 0.2 |
| WSTM | 3.604 | 3.616 | 1.571 | 1.626 | 0.602 | 0.4922 |
| Sine, 50% load | Maximum | | Minimum | | Peak Rate | |
| Input | LMPC | NMPC | LMPC | NMPC | LMPC | NMPC |
| WCHR | 1.834 | 1.738 | 0.416 | 0.548 | 0.2 | 0.2 |
| WAIR | 13.896 | 13.916 | 10.222 | 10.15 | 0.6502 | 0.733 |
| WCOL | 8.122 | 8.108 | 5.441 | 5.381 | 0.2 | 0.2 |
| WSTM | 3.292 | 3.387 | 0.635 | 0.4905 | 0.6802 | 0.606 |
| Sine, 0% load | Maximum | | Minimum | | Peak Rate | |
| Input | LMPC | NMPC | LMPC | NMPC | LMPC | NMPC |
| WCHR | 2.427 | 2.665 | 0.066 | 0.059 | 0.2 | 0.2 |
| WAIR | 8.992 | 8.876 | 3.252 | 3.511 | 1 | 1 |
| WCOL | 6.264 | 6.346 | 3.172 | 2.961 | 0.2 | 0.2 |
| WSTM | 3.829 | 4.166 | 0 | 0 | 1 | 1 |