Nonlinear Model Predictive Control of a Laboratory Gas Turbine Installation

The feasibility of model predictive control (MPC) applied to a laboratory gas turbine installation is investigated. MPC explicitly incorporates (input and output) constraints in its optimizations, which explains the choice for this computationally demanding control strategy. Strong nonlinearities, displayed by the gas turbine installation, cannot always be handled adequately by standard linear MPC. Therefore, we resort to nonlinear methods, based on successive linearization and nonlinear prediction as well as the combination of these. We implement these methods, using a nonlinear model of the installation, and compare them to linear MPC. It is shown that controller performance can be improved, without increasing controller execution-time excessively.

Introduction

Gas turbines can be found, for instance, in jet engines and in the field of power generation. The laboratory installation is a scale model (400 kW thermal power input) of a true gas turbine and differs from a gas turbine in the traditional sense, in the fact that compressor and expander are separated by a buffer-tank, as will be depicted in Fig. 2. This setup is used to validate general purpose dynamic simulation models of turbomachines, and to investigate the control of such systems.

The gas turbine’s operation is limited by some undesirable effects, which we will incorporate as output constraints. Furthermore, limits on actuator operation impose input constraints. It is especially these constraints which have resulted in choosing MPC for a control strategy. Apart from explicit constraint handling, the advantages of MPC include anticipation to future setpoint changes (and possibly known disturbances), and the ability to handle deadtime responses and interactions in multivariable systems.

MPC uses a model of the process to predict future system response. Through minimizing the difference between desired and predicted outputs, the future inputs (or manipulated variables) can be determined. Standard linear MPC incorporates linear models both for prediction and optimization. Although it is successful in controlling linear and mildly nonlinear processes, performance degradation and instability often occur in the presence of strong nonlinearities. During the past decade, the number of nonlinear predictive control algorithms has increased significantly. A comprehensive review of nonlinear control is provided by Bequette (1992). However, the excessive computational requirements of such methods remain a serious obstacle to industrial implementation in spite of the advances made in developing efficient algorithms. Also, little progress has been made in understanding their stability and performance properties (Gattu, 1992).

For our purposes, we need an algorithm which handles nonlinearities adequately, but is guaranteed not to require computation times in excess of that which is allowable for real-time implementation. This limit is set at approximately 1 sec. per sample. Furthermore, a strategy must be developed to handle infeasibilities during the optimization phase.

This paper addresses three different approaches, which are then compared to linear MPC. The first approach we will look into, uses a linearization of the nonlinear model, obtained at each sampling time. This linear model is used to compute the effect of previous control moves, as well as to optimize future manipulated variables. A second approach uses a nonlinear model for prediction and again a linear model for optimization but does not repeat the linearization. Combining nonlinear prediction and successive linearization results in the third approach. A major advantage of these approaches is that they allow the use of proven linear optimization algorithms to be used on a nonlinear problem. The third approach was utilized earlier by Garcia (1984) and has been refined by Gattu (1992) and Lee (1993) to include an Extended Kalman Filter. We do not linearize repeatedly within the prediction horizon, something which is done, e.g., by Bregel (1989), Li (1989), and De Oliveira (1995). These methods are expected to increase computational demands substantially, since they incorporate an iterative (SQP) optimization problem instead of a single QP problem.

The main objective of this paper is to assess the feasibility (stability, real-time performance, robustness) and benefits of MPC on a system as fast as the gas turbine installation. We evaluate situations where our nonlinear model perfectly describes the plant, as well as situations where parametric plant/model mismatch is introduced. In a larger context, one of the goals is to implement the control algorithm in a real-time situation. This paper only describes results from simulations, which were obtained using PRIMACS, the MPC-implementation developed by TNO-TPD at Delft. Experiments will be performed shortly and reported in a successor of this paper. Initial experimental results are encouraging.

The paper is organized as follows: It starts with a short description of the gas turbine installation and some key aspects of the model we developed. In the following section, we will elucidate the concept of MPC as it is implemented in PRIMACS and describe the nonlinear methods that were newly developed and added to the package. The concluding sections present and discuss simulation results, respectively.

The Gas Turbine Installation

A gas turbine produces mechanical (shaft) power by expansion of compressed gas—usually air—through an expander. The required pressure ratio is provided by a compressor. Compressor and expander are mounted on the same shaft and the expander powers the compressor. To overcome losses and develop useful power, energy has to be added by raising the temperature of the compressed air prior to expansion. This is accomplished in a combustion chamber which is positioned between compressor and expander.
Of special interest to us are the restrictions to the operation of the gas turbine. Operating points are determined by the combination of the characteristics of the comprising components. Operation restrictions are therefore related to these components. Furthermore, since the compressor appears to be the most critical component, the operating points of the overall gas turbine are usually represented in the compressor characteristic, which is depicted in Fig. 1.

In this characteristic, curved lines represent the relationship between pressure ratio over, and mass flow through the compressor for various rotational speeds. The extrema of these lines are connected by what we refer to as the surge line.

Being the most severe restriction, surge denotes the phenomenon associated with violent limit cycle oscillations of mass flow and pressure rise, which originate in the compressor and are transmitted throughout the gas turbine. Surge is likely to happen when the mass flow through the compressor decreases enough to enter the part of the compressor characteristic where a decrease in mass flow will be accompanied by a fall of delivery pressure, i.e., where the lines of constant speed exhibit a positive slope. In the compressor characteristic this corresponds to passing the surge line, which more or less separates the stable zone from the unstable zone.

The second limitation we consider is the expander inlet temperature which may not exceed some maximum value. Beyond this temperature the life-time of the rotor decreases dramatically. In practice, this is the limitation we encounter most, especially when the fuel valve is opened rather quickly. Doing so causes the inlet temperature of the expander to rise instantaneously, only to decrease again after mass flows—being governed by larger timescales—have risen also.

The Laboratory Setup. Figure 2 schematically depicts the laboratory installation. It comprises a BBC turbocharger, which consists of a single stage radial compressor and a single stage axial expander. Also a combustion chamber and a buffer tank can be recognized. The main function of the buffer tank is to decouple the flows out of the compressor and into the expander, enabling the isolation of physical phenomena in both components. Other components are the blow-off valve, the throttle valve and the fuel valve, which are all powered by electric drives. The blow-off valve directly influences the ratio of mass flows through compressor and expander, while the throttle valve influences the ratio of compressor delivery and expander inlet pressure. The fuel valve controls the power supplied to the gas turbine. These three valves are used to reach different operating points of the installation. A compressed air facility is employed in start-up procedures. For detailed information about the laboratory installation, the reader is referred to Essen (1995).

Control Objectives. In a practical gas turbine application, objectives (other than surge avoidance) one can think of are operating at constant rotational speed for generator purposes, or, if the gas turbine is to be used as a power supply for some external load it might be important to deliver constant power. Of course, this should still be true if disturbances, such as varying ambient conditions, are present. Also, operating at maximum efficiency can be of significant importance, so as to minimize fuel consumption.

In this paper, we will not primarily seek to meet these real-life control objectives. Instead we will specify objectives that in one way or another “challenge” the controller. We will focus on setpoint control and constraint handling.

The outputs we wish to control are formed by the three variables which determine the position of the operating point of the gas turbine in the compressor characteristic. These are the pressure ratio over the compressor, the mass flow through the expander and the rotational speed. Of course, we will never seek to control all these three variables at the same time, because this would result in an inherently over-determined problem: two out of these three variables fully determine the position in the compressor characteristic.

In order to specify the surge line constraint in MPC, we define a dummy output. To avoid surge, the distance (in terms of mass flow) between operating point and surge line must be kept greater than zero. The second output constraint stems from the expander inlet temperature, which is not allowed to (continuously) exceed the bound of 900° K.

Furthermore, valves should be constrained not to operate beyond their saturations. Just as importantly, maximum moving rates should be specified, since real-life actuators cannot simply move with arbitrarily high speed. In MPC this is realized by specifying the maximum move size per sample period.

Modeling the Gas Turbine. We developed a model in which components of the installation like compressor, valves, and expander are represented by a set of (nonlinear) quasi-steady, algebraic equations. This implies that the component characteristics are used as static impulse balances to determine the mass flow through the components. This type of modeling of turbomachines has been widely accepted, see, e.g., Botros (1991).

Two successive components are coupled by a volume, in which the dynamic behaviour is lumped. In our model three volumes, corresponding to the compressor plenum, the buffer tank, and the combustion chamber volume, are applied for each of these, instantaneous mass and energy conservation laws are used to obtain state equations for pressure \(p\) and temperature \(T\):

\[
\frac{dp}{dt} = \frac{\gamma R}{V} [\dot{m}_i T_i - \dot{m}_o T_o]
\]

\[
\frac{dT}{dt} = \frac{RT}{pV} [\gamma (\dot{m}_i T_i - \dot{m}_o T_o) - T (\dot{m}_i - \dot{m}_o)],
\]

with \(\dot{m}_i\) and \(\dot{m}_o\) mass flows in and out of the volume \((V)\), and \(T_i\) the inlet temperature, \(\gamma\) and \(R\) denote the quotient of specific heat...
Model Predictive Control

Model predictive control uses a model of the process to predict the outputs up to a certain time instant, based on the inputs to the system and the most recent process measurements. For example, consider Fig. 3. At the present time \( k \), the response of the output \( y(k) \) to changes in the manipulated variables \( u(k) \) is predicted over the prediction horizon \( p \). The manipulated variables are allowed to vary over the control horizon \( m \). The manipulated variables are to be controlled to change inadmissibly large. Of the computed optimal control moves, only the values for the first sample are actually implemented. This way the most recent process measurements can be used to calculate a new sequence of control moves. This mechanism is known as a moving (or receding) horizon. A comprehensive treatment on linear MPC is provided by Morari (1991).

To minimize future deviations of the controlled variables from their setpoints, while preventing the inputs from changing inadmissibly fast, we use the quadratic objective function

\[
\min_{\Delta u(k+1), \ldots \Delta u(k+m)} \left\{ \sum_{i=2}^{p} \| \Gamma_i^e [y(k+il) - r(k + l)] \|^2 + \sum_{i=1}^{m} \| \Gamma_i^r [\Delta u(k+i)] \|^2 \right\},
\]

where weights are included to express the relative importance of outputs following their reference trajectory on the one hand and trading off with reducing the action of manipulated variables on the other. In this notation \( \Gamma_i^e \) and \( \Gamma_i^r \) represent the output and input-weightings, respectively. Furthermore, \( y(k+il) \) denotes the estimate of \( y(k+l) \) obtained at \( k \), taking into account all measurement information up to \( k \). Note that, at the current sample \( k \), input moves \( \Delta u \) are optimized starting from sample \( k+1 \), so as to minimize predicted output deviations from their setpoints, from sample \( k+2 \) onwards. This way, the computational delay is accounted for, with the inherent implication that \( y(k + 1l) \) can never be influenced at sample \( k \).

The solution to this constrained problem can be obtained analytically with relatively little effort. In general, though, constraints will always be present at one time or another, be it output constraints, input constraints or input move constraints. Incorporating these constraints leads to a quadratic program (QP) of the following form:

\[
\min \frac{1}{2} \Delta \mathbf{u}^T(k + 1) \mathbf{Q} \mathbf{u}(k + 1) - \mathbf{e}^T(k + 2) \Delta \mathbf{u}(k + 1)
\]

such that

\[
\mathbf{e}^T(k + 1) \geq \mathbf{e}(k + 2l),
\]

where \( \mathbf{Q} \) contains the control moves \( \Delta u(k + 1) \ldots \Delta u(k + m) \), whereas \( \mathbf{e}^* \) and \( \mathbf{e}(k + 2l) \) are the Hessian matrix and gradient vector. \( \mathbf{e}^* \) and \( \mathbf{e}(k + 2l) \) depend on the constraints on manipulated variables, change in manipulated variables, and outputs.

**Constraint Handling.** The QP may not always have a solution, which is why a strategy must be specified to handle these infeasibilities. In PRIMACS, a strategy is adopted where constraint violations are included into the optimization criterion, once infeasibility occurs. This way, violations are kept from becoming inadmissibly large. It is implemented by simply adding the \( \ell_2 \)-norm of the (suitably weighted) constraint violations \( e \) \((e \geq 0)\) to the optimization criterion, which will then look like

\[
J_{\text{modified}} = J + \rho \| e \|_2,
\]

where \( J \) denotes the unmodified optimization criterion, and \( \rho \) the vector with weights on the constraint violations. The \( \ell_2 \)-norm is chosen over the \( \ell_1 \)-norm for its guaranteed open-loop stability when the unconstrained system is stable.

The total constraint handling strategy can now be summarized as follows:

1. Solve the unconstrained problem. If constraints are violated, go to step 2. Else, implement the controller moves and return to step 1.
2. Find a solution to the QP problem. If infeasible, go to step 3. Else, implement the controller moves and return to step 1.
3. Include constraint violations into the QP, and solve this modified problem. Implement the controller moves and return to step 1.

**Internal Model.** We assume that the internal model (the model MPC uses for its computations) is expressed through the following nonlinear differential equations:

\[
\dot{x}(t) = f(x(t), u(t))
\]

\[
y(t) = g(x(t), u(t)) + d(t),
\]

where \( x(t) \) are the model states, \( u(t) \) the manipulated variables, \( y(t) \) the outputs, and \( d(t) \) unmeasured disturbances. We adopted the same output disturbance model Garcia (1984) uses, where disturbances are added directly to each output.

In order to handle nonlinearities adequately, we will look into three nonlinear extensions to linear MPC, which we will denote as successive linearization, nonlinear prediction and the combination of these, nonlinear prediction, and successive linearization. Eventually, we will compare the results of these methods to those obtained with linear MPC.

**Successive Linearization.** This method linearizes the nonlinear model at each sampling instant. The linear model thus obtained
is expected to describe the dynamics better than the one, obtained from linearizing only once.

Using the first-order Taylor expansion of (2)–(3), we obtain the following set of linear equations:

\[
\dot{x}(t) = x_0 + \dot{x}(t) = f(x_0, u_0) + A\dot{x}(t) + B\dot{u}(t)
\]

\[
y(t) = y_0 + \dot{y}(t) = g(x_0, u_0) + C\dot{x}(t) + D\dot{u}(t) + d(t),
\]

where \(x_0, u_0\) are the nominal states and inputs around which we linearize, and \(\dot{x}(t), \dot{u}(t)\) the deviations from these nominal values. Furthermore,

\[
A := \frac{\partial f(x, u)}{\partial x} \bigg|_{x=x_0, u=u_0}
\]

\[
B := \frac{\partial f(x, u)}{\partial u} \bigg|_{x=x_0, u=u_0}
\]

\[
C := \frac{\partial g(x, u)}{\partial x} \bigg|_{x=x_0, u=u_0}
\]

\[
D := \frac{\partial g(x, u)}{\partial u} \bigg|_{x=x_0, u=u_0}
\]

Let \(\Phi, \Gamma \) \((C \text{ and } D)\) denote the discrete-time versions of these matrices. Similarly, \(f(x_0, u_0)\) denotes the discrete-time version of \(f(x_0, u_0)\), which is obtained by integrating (2) for one sample interval with the initial condition of \(x = x_0\) and constant inputs \(u_0\).

This yields the discrete-time model representation

\[
\begin{bmatrix}
  x(k+1) \\
  d(k+1)
\end{bmatrix} =
\begin{bmatrix}
  \Phi & 0 \\
  0 & I
\end{bmatrix}
\begin{bmatrix}
  x(k) \\
  d(k)
\end{bmatrix} +
\begin{bmatrix}
  \Gamma \\
  0
\end{bmatrix} u(k)
\]

\[
= + \begin{bmatrix}
  0 \\
  K_F
\end{bmatrix} q(k) +
\begin{bmatrix}
  \Phi(x_0, u_0) & 0
\end{bmatrix}
\begin{bmatrix}
  f(x_0, u_0)_x \\
  0
\end{bmatrix}
\]

\[
y_m(k+1) =
\begin{bmatrix}
  C & I
\end{bmatrix}
\begin{bmatrix}
  x(k+1) \\
  d(k+1)
\end{bmatrix} + Du(k) + g(x_0, u_0)
\]

\[
q(k) = q_y(k) - y_m(k),
\]

which includes an output disturbance filter. The output disturbances \(d(k+1)\) are obtained from a first-order integrating filter \(K_F\) with \(q(k)\) as input, with \(K_F\) taken to be a diagonal matrix. Furthermore, \(y_p\) are the process output measurements, and \(y_m\) the corresponding model outputs. All of our regulated variables are assumed to be measured, and no system or measurement noise is assumed to be present.

The elements of \(K_F\) should be chosen somewhere between 0 and 1. Taking these filter constants close to 1 is useful only if the process measurements are quite accurate. We chose to take the filter gains equal to 0.1, to allow for some difference between model and process.

We obtain \(x_0\) and \(u_0\) from \(x(k)\) and \(u(k)\), but one could also choose to reconstruct \(x\) (and \(u\)) from process measurements \(y\), naturally in conjunction with a noise filter.

**Nonlinear Prediction.** This approach uses a nonlinear model to predict the output response to previous control moves, but is no different from linear MPC otherwise. Though formally not correct, this approach (and the following) relies on the premise that the superposition theorem—which is valid for linear systems—still holds to a reasonable extent. The method requires execution-times, which—depending on the efficiency of the nonlinear integration algorithm—more or less severely exceed those of the method of successive linearization.

**Nonlinear Prediction and Successive Linearization.** Computationally even more demanding is the combination of nonlinear prediction and successive linearization. This approach should combine the benefits of using an updated linear model to perform the optimizations on the one hand, and obtaining output predictions from the nonlinear model on the other.

**Results**

In this section we present results obtained with the various methods we described. We compare these results to those obtained with linear MPC, and state execution times. In all figures, 1 X lin stands for linearizing once (=linear MPC), sl is short for successive linearization, nlp denotes the case where nonlinear prediction is used, and nlp + sl refers to the combination of nonlinear prediction and successive linearization. Furthermore, the solid (step-shaped and straight) lines correspond to setpoints and temperature constraint. We assume this to become clear from the accompanying text. Throughout the simulations, the prediction and control horizon were chosen to be 15 and 4 samples, respectively. Finally, we mention the fact that valve positions can vary between 0 (closed) and 1 (fully opened).

**Simple Setpoint Change.** Figure 4 shows the response to setpoint changes in compressor pressure from 1.62 to 2.00 (bar), and in compressor mass flow from 0.464 to 0.640 (kg/s), both at \(t = 20\) (s). Corresponding valve-positions are shown in Fig. 5. All approaches give satisfactory results, although the pressure response obtained with linear MPC displays quite some overshoot. Indeed, all controllers anticipate to future set-point changes. Note, however, that all controllers make the outputs move initially in the wrong direction, so as to make the rises as steep as possible. This effect has been seen to decrease if larger control horizons are used. Also note that these setpoints can be attained with different combinations of inputs. Due to blow-off, the mass flows through...
Unreachable Setpoint. We next specify a combination of setpoint changes, which simply cannot be realized. Mass flow is supposed to drop to 0.330 (kg/s), while keeping a constant pressure. This point would not only result in a temperature constraint violation, but also in surge. Figure 6 shows the resulting responses. Of course, in all cases pressure does not retain its initial value, whereas mass flow does not decrease as far as we wanted it to. This is exactly what we expected from the controller, in order not to violate constraints. However, linear MPC encounters serious trouble, as it cannot keep expander inlet temperature from reaching values in excess of 930° K. The surge constraint is almost reached (not shown), but is kept from doing so by the controller actions which serve to prevent temperature constraint violations. Again, sl and nlp + sl are hardly distinguishable.

Simple Setpoint Change With Model Errors. Finally, we repeat the "simple setpoint change" simulation, but different in the

Table 1 Maximum controller execution-time per sample

<table>
<thead>
<tr>
<th>execution-time [s]</th>
<th>1 x lin</th>
<th>sl</th>
<th>nlp</th>
<th>nlp+sl</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1.2</td>
<td>2.3</td>
<td>3.0</td>
<td></td>
</tr>
</tbody>
</table>
(t) the fact that filter actions cannot adequately compensate for the in this time interval corresponded to a (nearly) stationary situation.

balances we already experienced in the previous simulations. MPC to successive linearization only, considering the close resemblance constraint is slightly violated for some time, which is due to some of the most important parameters (such as compressor efficiency) by 5 to 20 percent. Figure 7 shows that both linear and nonlinear yield results which are acceptable. Here we compare linear MPC to successive linearization only, considering the close resemblances we already experienced in the previous simulations.

Note that, even in case of successive linearization, the temperature constraint is slightly violated for some time, which is due to the fact that filter actions cannot adequately compensate for the differences between model and process, because controller actions in this time interval (r \in [10, 42]) are still quite severe (not shown). This situation would not occur if the expander temperature in this time interval corresponded to a (nearly) stationary situation.

**Discussion**

We presented three nonlinear extensions to linear model predictive control, the last of these resulting in the same approach García (1984) used. We also incorporated the same output disturbance model encountered in that paper, and stress that—though not especially relevant to our problem—this strategy may not perform well in controlling integrating processes and may lead to instabilities when applied to open-loop unstable processes. To prepare for real-time implementation, we are currently researching the incorporation of a Kalman filter in the on-line optimization, to allow for reconstruction of unmeasured variables as well as to handle measurement noise.

The three nonlinear methods all offer improvements over linear MPC, with results in fact almost identical most of the time. This conclusion would favour whichever approach is computationally the least demanding. Most likely, the method which incorporates successive linearization, but refrains from nonlinear prediction, requires the least computational effort. Nevertheless, the execution-times stated should be viewed with scepticism, since the various algorithms involved can probably all be made more efficient, one way or another. For instance, using a more efficient integration routine would speed up the methods using nonlinear prediction, whereas a more efficient QP solver would contribute to reducing the requirements for successive linearization.

Judging from the simulations, it is concluded that, if no measurement noise is present and if the model describes the process fairly well, linear MPC is suited to control the gas turbine installation most of the time, requiring execution-times within the pre-specified limit of 1 (s) per sampling period. The benefits MPC offers, such as anticipation, input/output constraint handling, were shown to hold for all approaches investigated. Still, some situations may cause linear MPC to perform significantly worse, situations where nonlinear extensions offer considerable improvements. However, for the time being, none of these nonlinear methods can be made to require execution-times within the limit of 1 (s), although sl is not far from achieving this goal. In practice, the somewhat arbitrarily set bound of 1 (s) might be slightly increased or, alternatively, a faster computer may be used to render at least sl "feasible".

It should still be investigated if the approaches mentioned can be guaranteed to improve controller performance at all times, especially in a real-time application, where measurement noise is present and robustness of the control algorithm becomes increasingly important. Although the various methods were shown to possess some kind of robustness, more rigorous tests should point out if this result holds for a larger class of model uncertainties.

**Acknowledgment**

The authors gratefully acknowledge TNO-TPD at Delft for their support and for providing the PRIMACS software.

**References**


