Nigerian Journal of Technology (NIJOTECH)
Vol. 31, No. 2, July, 2012, pp. 139–148.

MODEL PREDICTIVE CONTROL FUNDAMENTALS

P.E. Orukpe

[a]Department of Electrical/Electronic Engineering, University of Benin, Nigeria.
Email: *patience.orukpe01@imperial.ac.uk*

Abstract

Model Predictive Control (MPC) has developed considerably over the last two decades, both within the research control community and in industries. MPC strategy involves the optimization of a performance index with respect to some future control sequence, using predictions of the output signal based on a process model, coping with constraints on inputs and outputs/states. In this paper, we will present an introduction to the theory and application of MPC with Matlab codes written to simulate an example of a randomly generated system. This paper can serve as tutorial to anyone interested in this area of research.

Keywords: model predictive control, linear systems, discrete-time systems, constraints, quadratic programming

1. Introduction

Model Predictive Control (MPC), also known as Moving Horizon Control (MHC) or Receding Horizon Control (RHC), is a popular technique for the control of slow dynamical systems, such as those encountered in chemical process control in the petrochemical, pulp and paper industries, gas pipeline control, waste heat boiler [1], and active vibration control of railway vehicle [2]. Using the definition of Lee and Marcus 1967 - MPC is one technique for obtaining a feedback controller synthesis from knowledge of open-loop controllers to measure the current processes state and then compute very rapidly for this open-loop control function. The first portion of this function is then used during a short interval, after which a new value of the function is computed for this measurement.

The name MPC stems from the idea of employing an *explicit* model of the plant to be controlled which is used to *predict* the future output behavior. This prediction capability allows solving optimal control problems on line, where tracking error, namely the dierence between the predicted output and the desired reference, is minimized over a future horizon, possibly subject to constraints on the manipulated inputs, outputs and states. When the model is linear, then the optimization problem is quadratic if the performance index is expressed through the ℓ_1- norm, or linear if expressed through the ℓ_1/ℓ_∞ - norm [3]. The result of the optimization is applied according to the receding horizon philosophy: At time t only the first input of the optimal command sequence is actually applied to the plant. The remaining optimal inputs are discarded, and a new control problem is solved at time $t + 1$. As new measurements are collected from the plant at each time t, the receding horizon mechanism provides the controller with the desired feedback characteristics [4].

MPC has been successful, as it can handle constraints on the manipulated and controlled variables in a systematic manner, handles multivariable control problems naturally, it is an easy to tune method, it is a totally open methodology based on certain basic principles which allow for future extensions [5], [6]. Discussions on some of the design techniques based on MPC and their properties can be found in the review article [3]. Stability of MPC has been achieved by the use of terminal cost function, a terminal constraint set innite horizon, fake algebraic riccati equation and a local controller [7], [8]. An overview of commercially available Model Predictive Control technology as given by vendors for both linear and non-linear MPC is presented in a survey [9].

Although MPC has been successful in industries with slow processes, computational issues with respect to fast processes are still open to further research. In this paper, we will shed light on the basics of MPC with or without constraint and this work was motivated by the work in [10]. This paper is organized as follows: section 2 describes what MPC involves; section 3 presents the problem formulation, section 4 outlines ways of solving predictive control problems, section 5 presents the simulation results and section 6 provides conclusion.

2. What is in MPC?

In this section, we will look at what MPC involves.

2.1. MPC Components

All the MPC algorithms possess common elements and different options can be chosen for each element giving rise to different algorithms. These elements are:

- Predictive model (i.e. process model and disturbance model),
- Objective function and
- Obtaining the control law.

The basic structure of MPC is as shown in figure 1.

The model is the cornerstone of MPC and can be either process model or disturbance model. Process model is of the following form:

- impulse response – used mostly in the industry
- step response
- transfer function – midway between industry and academic community
- state space – used mostly in the academic research community

while the disturbance model commonly used is Controlled AutoRegressive and Integrated Moving Average (CARIMA).

2.2. What makes MPC successful in industry?

Some of the features that have contributed to the widespread and successful use of MPC in industries are listed below:

- It handles multivariable control problems naturally.
- It can take account of actuator limitations.
- It allows operation closer to constraints, hence increased profit.
- It handles structural changes.
- It has plenty of time for on-line computations.
- It can handle non-minimal phase and unstable processes.
- It is an easy to tune method.

2.3. Characteristics of MPC

In this section, we will state the ideas that are used in MPC. The ideas, appearing in MPC are basically:

1. Moving horizon implementation
2. Performance oriented time domain formulation
3. Incorporation of constraints
4. Explicit system model used to predict future plant dynamics.

2.4. Types of MPC

There are two main classes of MPCs:

Linear MPC

1. Uses linear model: $\dot{x} = Ax + Bu$
2. Quadratic cost function: $F = x^T Qx + u^T Ru$
3. Linear constraints: $Hx + Gu < 0$
4. Quadratic program.

Nonlinear MPC

1. Nonlinear model: $\dot{x} = f(x, u)$
2. Cost function can be nonquadratic: $F = (x, u)$
3. Nonlinear constraints: $h(x, u) < 0$
4. Nonlinear program

2.5. Variations of MPC

In this section, we will state other classes of MPC. Other classes of MPCs include:

- Robust MPC – guarantees feasibility and stability,
- Feedback MPC – mitigate shrinkage of feasible region,
- Pre-computed MPC – Piecewise-linear solution stored in database or solved off-line using parametric (linear or quadratic) programming, and
- Decentralized MPC as used in autonomous air vehicle – speed up computation

2.6. Basics of MPC

MPC consist of the following three ideas [5]:-

1. Explicit use of a model to predict the process output along a future time horizon
2. Calculation of a control sequence to optimize a performance index
3. A receding horizon strategy, so that at each instant the horizon is moved towards the future, which involves the application of the first control signal of the sequence calculated at each step.

The methodology of all the controllers belonging to the MPC family is characterized by the following strategy represented in Figure 2:

Step 1: The predictive future outputs $\widehat{y}(t + k|t)$, $k = 1, \ldots, N$, for the prediction horizon are calculated at each instant t using the process model. These depend upon the known values up to instance t (past inputs and outputs), including the current output (initial condition) $y(t)$ and on the future control signals $u(t + k|t)$, $k = 0, \ldots, N - 1$, to be calculated.

Step 2: The sequence of future control signals is computed to optimize a performance criterion, often to minimize the error between a reference trajectory and the predicted process output. Usually the control effort is included in the performance criterion.

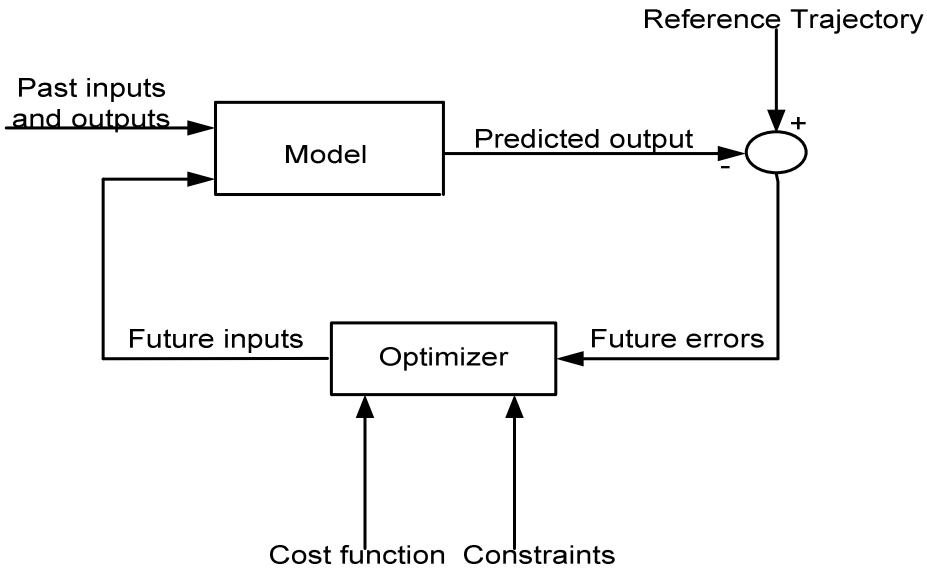

Figure 1: Basic structure of MPC.

Step 3: Only the current control signal $u(t|t)$ is transmitted to the process. At the next sampling instant $y(t+1)$ is measured and step 1 is repeated and all sequences brought up to date. Thus $u(t+1|t+1)$ is then calculated using the receding horizon concept, since the prediction horizon remains of the length as before, but slides along by one sampling interval at each step.

With the aid of increase speed in computer hardware MPC can now be applied to 'fast' processes instead of the confinement to slow processes. MPC can be tuned easily apart from the fact that it handles constraints properly. Constraints can be either on the output of the controlled processes (control variable) or on the control signals that is inputs to the process (manipulated variables). These constraints are in the form of saturation characteristics e.g. valves with a finite range of adjustment, control surface with limited deflection angles etc. Input constraints also appear in the form of rate constraints: valves and other actuators with limited slew rates.

3. Problem Formulation

In this section, we describe the type of system model used in this work, and give a brief formulation of the MPC problem. In this work a different method of formulation is used as opposed to that used in [11].

3.1. State space model

MPC can be used on any model, all that is needed to know is the step response at the coincidence points and to be able to compute the 'free response' at the coincidence points – i.e. one must have a simulation model capable of running faster than real time.

Model of the plant in state space is of the form

$$x(k+1) = Ax(k) + Bu(k) \tag{1}$$

$$y(k) = C_y x(k) \tag{2}$$

$$z(k) = C_z x(k) \tag{3}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^l$, $y \in \mathbb{R}^{m_y}$, $z \in \mathbb{R}^{m_z}$

x is the state vector, u is the input vector, y is the measured outputs and z is the vector of outputs which are to be controlled, either to particular set-points, or to satisfy some constraints, or both.

The type of system model used in this paper can be of the following types:

- Linear model, nonlinear plant – The plant in reality behaves in some complicated nonlinear fashion.
- First principles model and identification – The linearized model (1) – (3) used in predictive control can be obtained by performing test on the plant, which involves injecting known signals, such as steps, multi-sines, pseudo random, at the plant inputs, and recording the resulting outputs. Linearized models can then be obtained by using the techniques of system identification, which range from simple curve-fitting to sophisticated statistically based methods. A model in which the equations are obtained from a knowledge of the underlying physical, chemical and thermodynamic processes is available in nonlinear model of the plant. This is done for the purpose of

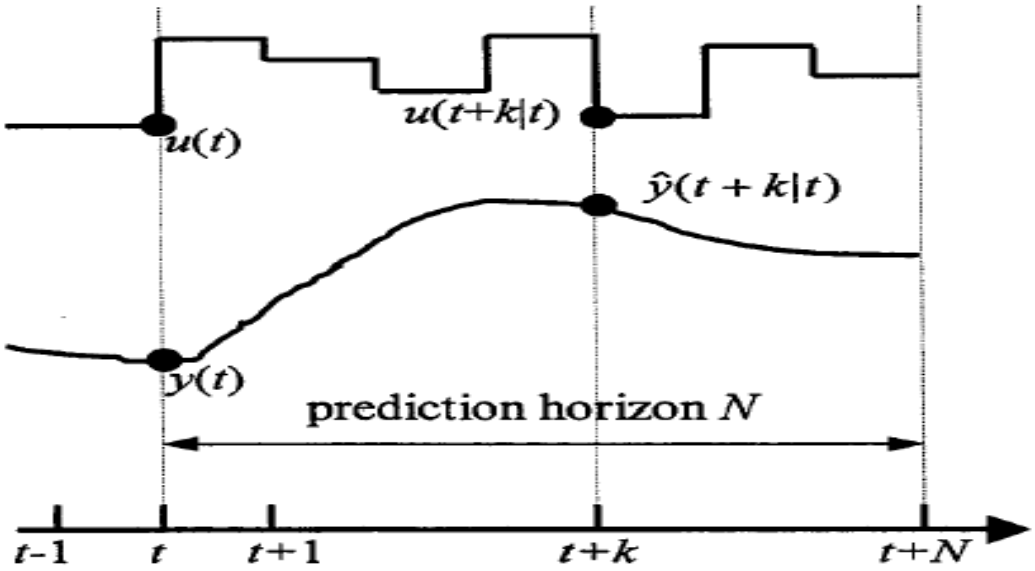

Figure 2: Model predictive control strategy.

operator training or safety certification. This is done by applying perturbations to the nonlinear model, and estimating the Jacobian matrices numerically. Another very effective way is to use the nonlinear model to generate simulation data for particular conditions, and then to apply system identification techniques to this data, as if it has been obtained from the plant itself [7].

3.2. A basic formulation

A formulation is done on the assumption that the plant model is linear, time invariant, the cost function is quadratic and the constraints are in the form of linear inequalities. The cost function does not penalize particular values of the input vector $u(k)$, but only changes of the input vector $\Delta u(k)$.

3.2.1. Cost function

The cost function V penalizes deviations of the predicted control outputs $\widehat{z}(k+i \setminus k)$ from a (vector) reference trajectory $\widehat{r}(k+i \setminus k)$.

The reference trajectory defines an important aspect of closed-loop behaviour of a controlled plant and it is frequently assumed that the reference trajectory approaches the set-point exponentially from the current output value, with the 'time constant' of the exponential, which shall be denoted T_{ref}. Sampling interval is denoted as T_s.

$$\begin{aligned}\widehat{r}(k+i \setminus k) &= s(k+i) - \epsilon(k+i) = s(k+i) - e^{iT_s/T_{ref}}\epsilon(k) \\ &= s(k+i) - \lambda^i \epsilon(k) \Longrightarrow \epsilon(k) = s(k) - y(k)\end{aligned}$$

where ϵ = current error and s = set-point.

$$\begin{aligned}V(k) = &\sum_{i=H_w}^{H_p} ||\widehat{z}(k+i \setminus k) - r(k+i \setminus k)||^2_{Q(i)} \\ &+ \sum_{i=0}^{H_u-1} ||\Delta\widehat{u}(k+i \setminus k)||^2_{R(i)}\end{aligned} \tag{4}$$

H_p - prediction horizon, H_u - control horizon and H_w - window parameter and is assumed to be one, $H_u \leq H_p$, $\Delta\widehat{u}(k+i \setminus k) = 0$ for $i = H_u$, thus $\widehat{u}(k+i \setminus k) = \widehat{u}(k+H_u-1)$ for all $i \geq H_u$.

The form of the cost function (4) implies that the error vector $\widehat{z}(k+i\setminus k) - r(k+i\setminus k)$ is penalized at every point in the prediction horizon, in the range $H_w \leq i \leq H_p$. This is indeed the most common situation in predictive control.

The tuning parameters H_p, H_u, H_w, $Q(i)$ and $R(i)$ affects the behaviour of the closed-loop combination of the plant and predictive controller, which are adjusted to give satisfactory dynamic performance.

3.2.2. Constraints and MPC

One of the major selling points of MPC is its ability to do on-line constraint handling in a systematic way, hopefully retaining to some extent the stability margins and performance of the unconstrained law. The algorithm does this by optimizing predicted performance subject to constraint satisfaction. There are various types of constraints on input, output or state, namely:

- Band constraints
- Overshoot constraints

- Monotonic behaviour
- Nonminimal phase behaviour
- Actuator nonlinearities
- Terminal state equality constraints and
- Terminal set constraints.

Constraints on the input are typically *hard* constraints, since they represent limitations on process equipment (such as valve saturation), and as such cannot be relaxed or softened. Constraints on the output (i.e. performance goals), are mainly due to safety reasons and must be controlled in advance because output variables are affected by process dynamics, and it is usually only required to make y_{max} and $y_{i,max}$ as small as possible, subject to the input constraints.

Constraints of the following form hold over the control and prediction horizons where $vec(0)$ denotes a column vector of elements zero.

$$E\ vec\left(\Delta\widehat{u}(k \setminus k), \ldots, \Delta\widehat{u}(k+H_u-1 \setminus k), 1\right) \leq vec\ (0) \tag{5}$$

$$F\ vec\left(\widehat{u}(k \setminus k), \ldots, \widehat{u}(k+H_u-1 \setminus k), 1\right) \leq vec\ (0) \tag{6}$$

$$G\ vec\left(\widehat{z}(k+H_w \setminus k), \ldots, \widehat{z}(k+H_p-1 \setminus k), 1\right) \leq vec\ (0) \tag{7}$$

E, F and G are matrices of suitable dimensions and the constraints represent actuator slew rates, actuator ranges and controlled variable constraints respectively.

3.3. Predictions

To solve the MPC problem, we must have a way of computing the predicted values of the control variables, $\widehat{z}(k+i \setminus k)$ from our best estimate of the current state $\widehat{x}(k \setminus k)$, and the assumed future inputs, or equivalently, the latest input $u(k-1)$ and the assumed future input changes, $\Delta\widehat{u}(k+i \setminus k)$. The choice of prediction is another 'tuning parameter', it has great effect on the performance of the closed-loop system.

The prediction of any system can be done base on the following:

1. No disturbance, full state measurement. In which (1) – (3) are iterated
2. Using an observer. In this case the state of the plant is unknown i.e. one cannot measure the full state vector. The use of Kalman filter come into play when the output of the plant is subjected to white noise disturbances with known covariance matrices. The gain of the observer is a 'tuning' parameter, which affects the behaviour of the predictive controller – particularly its response to disturbances, its stability margins etc [7].

3.4. Measured state, no disturbance

Assuming that the whole state is measured, so that $\widehat{x}(k \setminus k) = x(k) = y(k)$, so $(C_y = I)$, and nothing is known about the disturbance or measurement noise. Prediction is made by iterating the model (1), as follows:

$$\widehat{x}(k+1 \setminus k) = Ax(k) + B\widehat{u}(k \setminus k) \tag{8}$$

$$\widehat{x}(k+2 \setminus k) = A\widehat{x}(k+1 \setminus k) + B\widehat{u}(k+1 \setminus k) \tag{9}$$

$$= A^2x(k) + AB\widehat{u}(k \setminus k) + B\widehat{u}(k+1 \setminus k) \tag{10}$$

$$\vdots$$

$$\widehat{x}(k+H_p \setminus k) = A\widehat{x}(k+H_p-1 \setminus k) + B\widehat{u}(k+H_p-1 \setminus k) \tag{11}$$

$$= A^{H_p}x(k) + A^{H_p-1}\widehat{u}(k \setminus k), \ldots, B\widehat{u}(k+H_p-1 \setminus k) \tag{12}$$

$\widehat{u}(k \setminus k)$ was used instead of $u(k \setminus k)$, because at the time when we need to compute the predictions we do not know what $u(k)$ will be.

Assuming that the input will only change at times $k, k+1, \cdots, k+H_p-1$, and will remain constant after that, so we have $\widehat{u}(k+i \setminus k) = \widehat{u}(k+H_u-1)$ for $H_u \leq i \leq H_p-1$. Let the prediction be expressed in terms of $\Delta\widehat{u}(k+i \setminus k)$ rather than $\widehat{u}(k+i \setminus k)$, knowing that $\Delta\widehat{u}(k+i \setminus k) = \widehat{u}(k+i \setminus k) + \widehat{u}(k+i-1 \setminus k)$ and at the time k we already know $u(k-1)$. So we have

$$\widehat{u}(k \setminus k) = \Delta\widehat{u}(k \setminus k) + u(k-1) \tag{13}$$

$$\widehat{u}(k+1 \setminus k) = \Delta\widehat{u}(k+1 \setminus k) + \Delta\widehat{u}(k \setminus k) + u(k-1) \tag{14}$$

$$\vdots$$

$$\widehat{u}(k+H_u-1 \setminus k) = \Delta\widehat{u}(k+H_u-1 \setminus k) + \ldots + \Delta\widehat{u}(k \setminus k) + u(k-1) \tag{15}$$

Substituting (13) – (15) into (8) – (12) and rearranging, we obtain the following matrix-vector form:

$$\begin{bmatrix} \widehat{x}(k+1 \setminus k) \\ \vdots \\ \widehat{x}(k+H_u \setminus k) \\ \widehat{x}(k+H_u-1 \setminus k) \\ \vdots \\ \widehat{x}(k+H_p \setminus k) \end{bmatrix} = \underbrace{\begin{bmatrix} A \\ \vdots \\ A^{H_u} \\ A^{H_u+1} \\ \vdots \\ A^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} B \\ \vdots \\ \sum_{i=0}^{H_u-1} A^iB \\ \sum_{i=0}^{H_u} A^iB \\ \vdots \\ \sum_{i=0}^{H_p-1} A^iB \end{bmatrix} u(k-1)}_{\text{past}}$$

$$+ \underbrace{\begin{bmatrix} B & \cdots & 0 \\ AB+B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_u-1} A^iB & \cdots & B \\ \sum_{i=0}^{H_u} A^iB & \cdots & AB+B \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^{H_p-1} A^iB & \cdots & \sum_{i=0}^{H_p-H-u} A^iB \end{bmatrix} \begin{bmatrix} \Delta\widehat{u}(k \setminus k) \\ \vdots \\ \Delta\widehat{u}(k+H_u-1 \setminus k) \end{bmatrix}}_{\text{future}} \tag{16}$$

The predictions of z are now obtained simply as:

$$\begin{bmatrix} \widehat{z}(k+1 \setminus k) \\ \vdots \\ \widehat{z}(k+H_p \setminus k) \end{bmatrix} = \begin{bmatrix} C_z & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_z \end{bmatrix} \begin{bmatrix} \widehat{x}(k+1 \setminus k) \\ \vdots \\ \widehat{x}(k+H_p \setminus k) \end{bmatrix} \tag{17}$$

Table 1: Dimensions of matrices and vectors involved in computing the optimal moves, (plant has l inputs, n states and m controlled outputs).

Matrix	Dimensions
$\mathbb{Q}$	$m(H_p - H_w + 1) \times m(H_p - H_w + 1)$
$\mathbb{R}$	$lH_u \times lH_u$
Ψ	$m(H_p - H_w + 1) \times n$
Υ	$m(H_p - H_w + 1)$
Θ	$m(H_p - H_w + 1) \times lH_u$
ε	$m(H_p - H_w + 1) \times 1$
$\mathcal{G}$	$lH_u \times 1$
$\mathcal{H}$	$lH_u \times lH_u$

4. Solving Predictive Control Problems

In this section, we outline the solution of MPC problem for the unconstrained and constrained cases.

4.1. Unconstrained problems

In this section, we will look at measured state, no disturbances. The cost function to be minimized is

$$V(k) = \sum_{i=H_w}^{H_p} ||\widehat{z}(k+i \setminus k) - r(k+i \setminus k)||^2_{Q(i)} + \sum_{i=0}^{H_u-1} ||\Delta\widehat{u}(k+i \setminus k)||^2_{R(i)} \tag{18}$$

This is rewritten as

$$V(k) = ||Z(k) - T(k)||^2_{\mathbb{Q}} + ||\Delta U(k)||^2_{\mathbb{R}} \tag{19}$$

where

$$Z(k) = \begin{bmatrix} \widehat{z}(k+H_w \setminus k) \\ \vdots \\ \widehat{z}(k+H_p \setminus k) \end{bmatrix};\ T(k) = \begin{bmatrix} \widehat{r}(k+H_w \setminus k) \\ \vdots \\ \widehat{r}(k+H_p \setminus k) \end{bmatrix};$$

$$\Delta U(k) = \begin{bmatrix} \Delta\widehat{u}(k \setminus k) \\ \vdots \\ \Delta\widehat{u}(k+H_u-1 \setminus k) \end{bmatrix}$$

and the weighting matrices $\mathbb{Q}$ and $\mathbb{R}$ are given as

$$\mathbb{Q} = \begin{bmatrix} Q(H_w) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q(H_p) \end{bmatrix};\ \mathbb{R} = \begin{bmatrix} R(0) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R(H_u-1) \end{bmatrix}$$

Combining (16) and (17), we get

$$\begin{bmatrix} \widehat{z}(k+1 \setminus k) \\ \vdots \\ \widehat{z}(k+H_p \setminus k) \end{bmatrix} = \begin{bmatrix} C_z & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_z \end{bmatrix} \begin{bmatrix} A \\ \vdots \\ A^{H_u} \\ A^{H_u+1} \\ \vdots \\ A^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} C_z & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_z \end{bmatrix} \begin{bmatrix} B \\ \vdots \\ \sum_{i=0}^{H_u-1} A^iB \\ \sum_{i=0}^{H_u} A^iB \\ \vdots \\ \sum_{i=0}^{H_p-1} A^iB \end{bmatrix} u(k-1) + \begin{bmatrix} C_z & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_z \end{bmatrix} \begin{bmatrix} B & \cdots & 0 \\ AB+B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_u-1} A^iB & \cdots & B \\ \sum_{i=0}^{H_u} A^iB & \cdots & AB+B \\ \vdots & \cdots & \vdots \\ \sum_{i=0}^{H_p-1} A^iB & \cdots & \sum_{i=0}^{H_p-H_u} A^iB \end{bmatrix} \times \begin{bmatrix} \Delta\widehat{u}(k \setminus k) \\ \vdots \\ \Delta\widehat{u}(k+H_u-1 \setminus k) \end{bmatrix} \tag{20}$$

Equation (20) implies that

$$Z(k) = \Psi x(k) + \Upsilon u(k-1) + \Theta\Delta U(k) \tag{21}$$

defining

$$\varepsilon(k) = T(k) - \Psi x(k) - \Upsilon u(k-1) \tag{22}$$

Making T the subject of the formula in (22) and subtracting it from (21) gives

$$Z(k) - T(k) = \Theta\Delta U(k) - \varepsilon(k)$$

Thus equation (19) is now in the form of

$$V(k) = ||\Theta\Delta U(k) - \varepsilon(k)||^2_{\mathbb{Q}} + ||\Delta U(k)||^2_{\mathbb{R}} \tag{23}$$

$$= [\Delta U(k)^T\Theta^T - \varepsilon(k)^T]\mathbb{Q}[\Delta U(k)\Theta - \varepsilon(k)] + \Delta U(k)^T\mathbb{R}\Delta U(k) \tag{24}$$

$$= \varepsilon(k)^T\mathbb{Q}\varepsilon(k) - 2\Delta U(k)^T\Theta^T\mathbb{Q}\varepsilon(k) + \Delta U(k)^T(\Theta^T\mathbb{Q}\Theta + \mathbb{R})\Delta U(k) \tag{25}$$

Equation (25) implies

$$V(k) = \text{constant} - \Delta U(k)^T\mathcal{G} + \Delta U(k)^T\mathcal{H}\Delta U(k)$$

where $\mathcal{G} = 2\Theta^T\mathbb{Q}\varepsilon(k)$ and $\mathcal{H} = \Theta^T\mathbb{Q}\Theta + \mathbb{R}$ which does not depend on $\Delta U(k)$.

To get the optimal of $\Delta U(k)$, we find the gradient of $V(k)$ and set it to zero, so that the optimal is

$$\Delta U(k)_{\text{opt}} = 0.5\mathcal{H}^{-1}\mathcal{G}$$

If the number of plant inputs is l, then we just use the first l rows of the vector $\Delta U(k)_{\text{opt}}$ which is represented as

$$\Delta U(k)_{\text{opt}} = [I_l \quad 0_l \quad \cdots 0_l]\mathcal{H}^{-1}\Theta^T\varepsilon(k) \tag{26}$$

The only part of the solution which changes from time to time is the 'tracking error' $\varepsilon(k)$.

4.2. Constrained problems

In this section, we look at the formulation of MPC as a QP problem. Now for a system subject to constraints, we recall that these constraints are in the form:

$$E\begin{bmatrix} \Delta U(k) \\ 1 \end{bmatrix} \leq 0 \tag{27}$$

$$F\begin{bmatrix} U(k) \\ 1 \end{bmatrix} \leq 0 \tag{28}$$

$$G\begin{bmatrix} Z(k) \\ 1 \end{bmatrix} \leq 0 \tag{29}$$

where $U(k) = [\widehat{u}(k \setminus k)^T, \ldots, \widehat{u}(k + H_u - 1 \setminus k)^T]^T$ is defined analogously to $\Delta U(k)$. We have to express all of these as constraints on $\Delta U(k)$. Suppose F has the form $F = [F_1,\ F_2,\ \ldots,\ F_{H_u},\ f]$ where each F_i is of size $q \times m$, and f has size $q \times 1$, so that (28) can be written as

$$\sum_{i=1}^{H_u} F_i \widehat{u}(k + i - 1 \setminus k) + f \leq 0$$

Since $\widehat{u}(k+i-1 \setminus k) = u(k-1) + \sum_{j=0}^{i-1} \Delta\widehat{u}(k+j \setminus k)$, we can write (28) as

$$\begin{aligned} \sum_{j=1}^{H_u} F_j \Delta\widehat{u}(k \setminus k) + \sum_{j=2}^{H_u} F_j \Delta\widehat{u}(k+1 \setminus k) + \cdots \\ + F_{H_u} \Delta\widehat{u}(k + H_u - 1 \setminus k) \\ + \sum_{j=1}^{H_u} F_j u(k-1) + f \leq 0 \end{aligned}$$

Now we define $\mathcal{F} = \sum_{j=1}^{H_u} F_j$ and $\mathcal{F} = [\mathcal{F}_1, \cdots, \mathcal{F}_{H_u}]$. Then (28) can be written as

$$\mathcal{F}\Delta U(k) \leq \mathcal{F}_1 u(k-1) - f \tag{30}$$

Equation (28) has been converted into a linear inequality constraint on $\Delta U(k)$. Next, we transform (29), assuming full state measurement, we can use (21) to write (29) as

$$G\begin{bmatrix} \Psi x(k) + \Upsilon u(k-1) + \Theta\Delta U(k) \\ 1 \end{bmatrix} \leq 0$$

Now let $G = [\Gamma,\ g]$, where g is the last column of G, this is the same as

$$\Gamma[\Psi x(k) + \Upsilon u(k-1)] + \Gamma\Theta\Delta U(k) + g \leq 0$$

or

$$\Gamma\Theta\Delta U(k)] \leq -\Gamma[\Psi x(k) + \Upsilon u(k-1)] - g \tag{31}$$

To convert equation (27) to an inequality form, let $E = [W, w]$ which will result in

$$W\Delta U(k) \leq -w \tag{32}$$

Then we assemble the inequalities (30), (31), and (32) into a single inequality

$$\begin{bmatrix} \mathcal{F} \\ \Gamma\Theta \\ W \end{bmatrix} \leq \begin{bmatrix} -\mathcal{F}_1 u(k-1) - f \\ -\Gamma[\Psi x(k) + \Upsilon u(k-1)] - g \\ -w \end{bmatrix} \tag{33}$$

The constrained optimization problem to be solved is

$$\min \Delta U(k)^T \mathcal{H} \Delta U(k) - G^T \Delta U(k) \tag{34}$$

subject to the inequality (33) and this is a standard optimization problem known as *Quadratic Programming*.

The solution of the predictive controller is the same as in the unconstrained case as long as the constraints are inactive. However, if the constraints become active then the controller becomes nonlinear.

4.3. Solving QP problems

Quadratic Programming (QP) methods are widely used in applications of model predictive control. Most applications require a linear model to represent the process of interest over a moving time horizon with a quadratic cost function to drive the controlled variables back to their setpoints. QP problems can be solved using either active set method or interior point method [12]. Active set method is the most common method of solving QP problems. For details see [5]. While the Interior point method is becoming more popular within MPC community, as the convergence rates are far faster. For further details see [7], [13].

4.4. Softening the constraints

Feasibility is usually a term applied to optimization problems and describes whether a solution exists. Sometimes, during the optimization stage, the region defined in the decision variables by the set of constraints is empty. In these conditions, the optimization algorithm cannot find any solution and the optimization problem is said to be *infeasible*. Also, unobtainable control objectives that take the process away from the operating point may cause infeasibility. An optimization problem is *feasible* when the cost function is bounded and there are points in the space of decision variables that satisfy all constraints.

A systematic way of dealing with infeasibility is to *soften* the constraints. That is, rather than regard the constraints as hard boundaries which can never be crossed, to allow them to be crossed occasionally, but only if really necessary.

There is an important distinction between input and output constraints. Usually, input constraints really are 'hard' constraints, and there is no way in which they can be softened: valves, control surfaces, and other actuators have limited ranges of action, and limited slew rates. Once these have been reached there

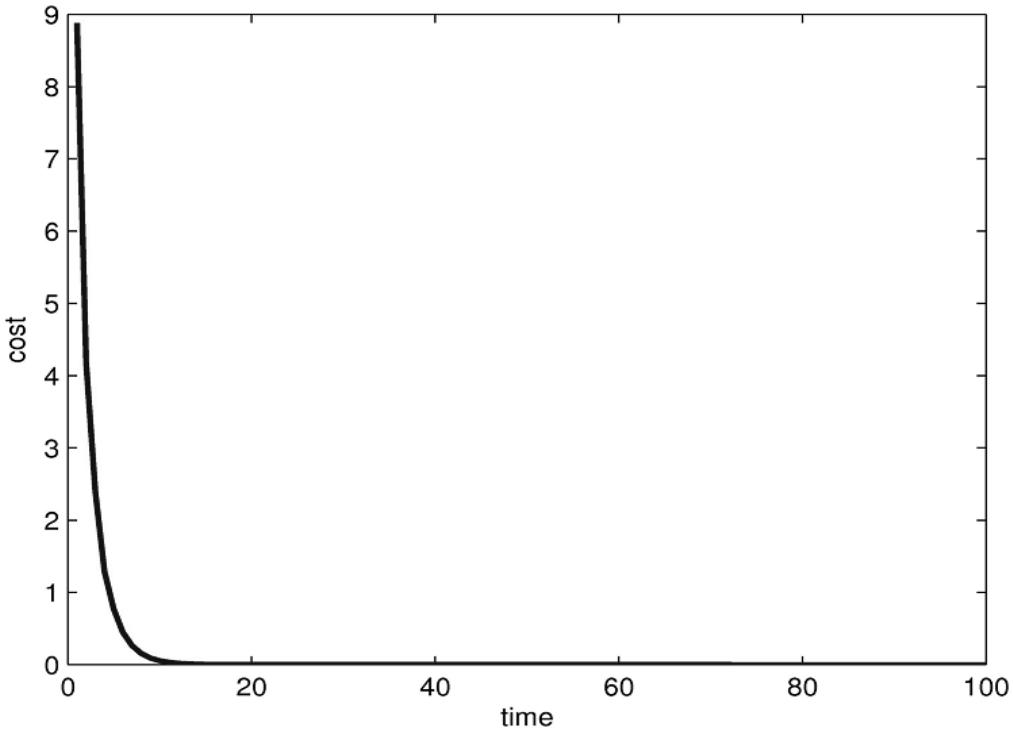

Figure 3: Cost function for unconstrained case.

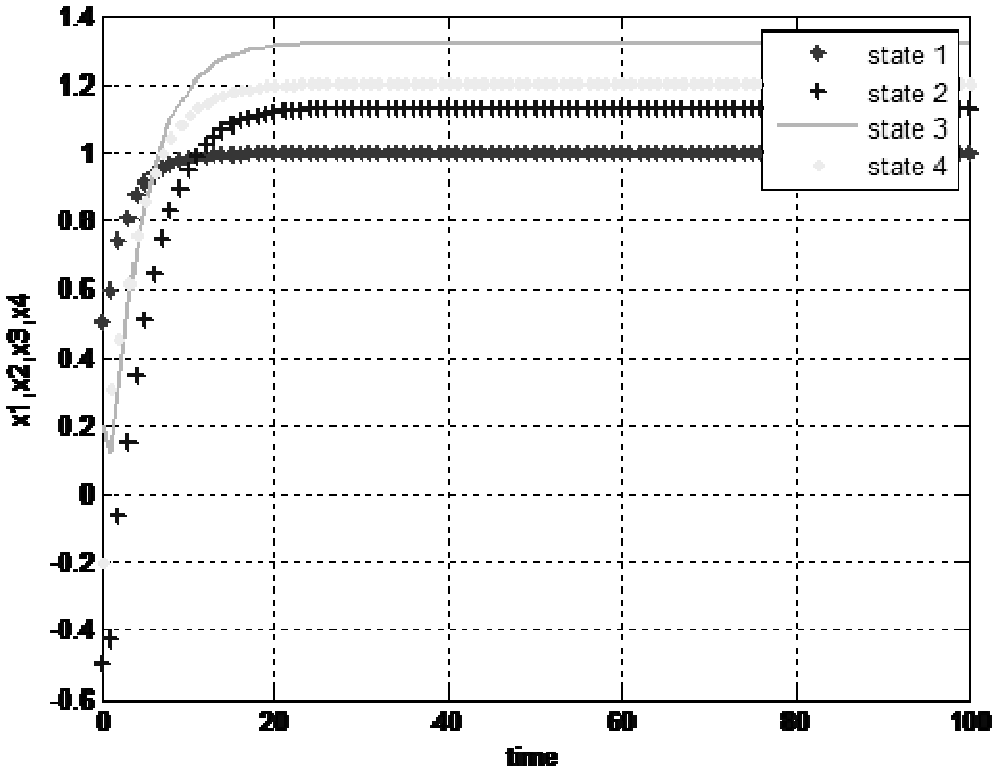

Figure 4: States for unconstrained case.

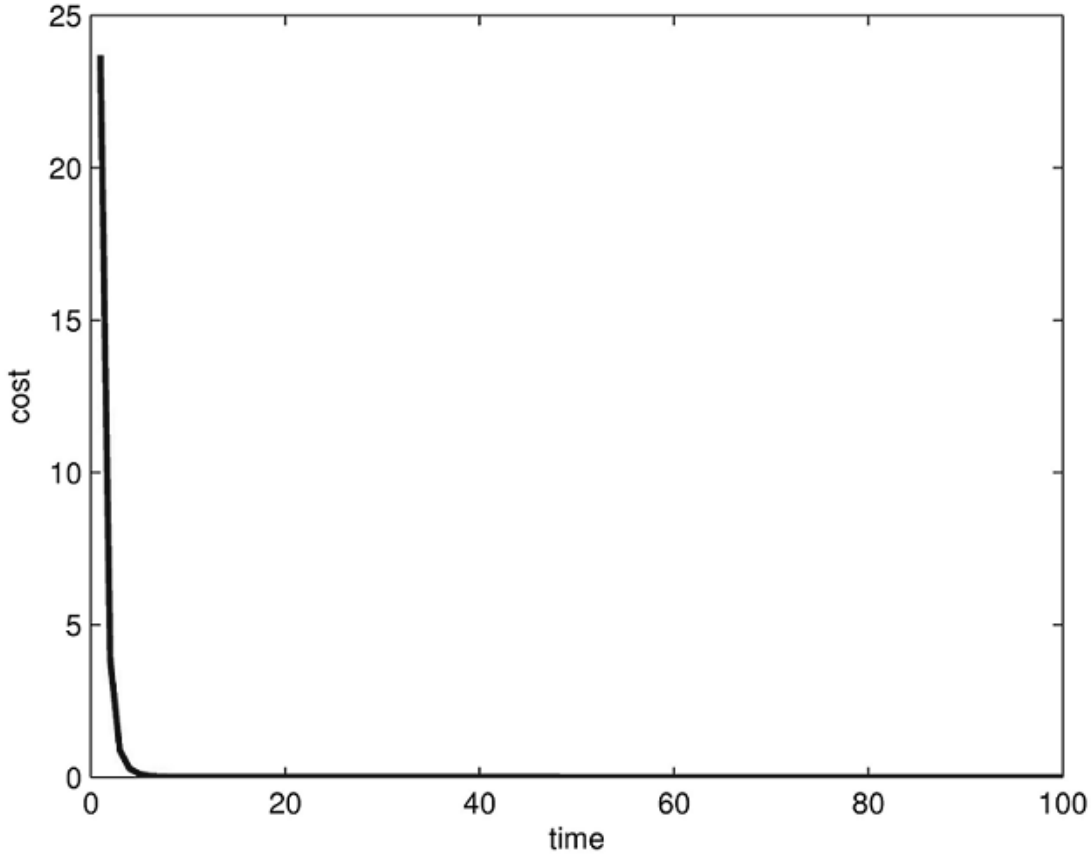

Figure 5: Cost function for the constrained case.

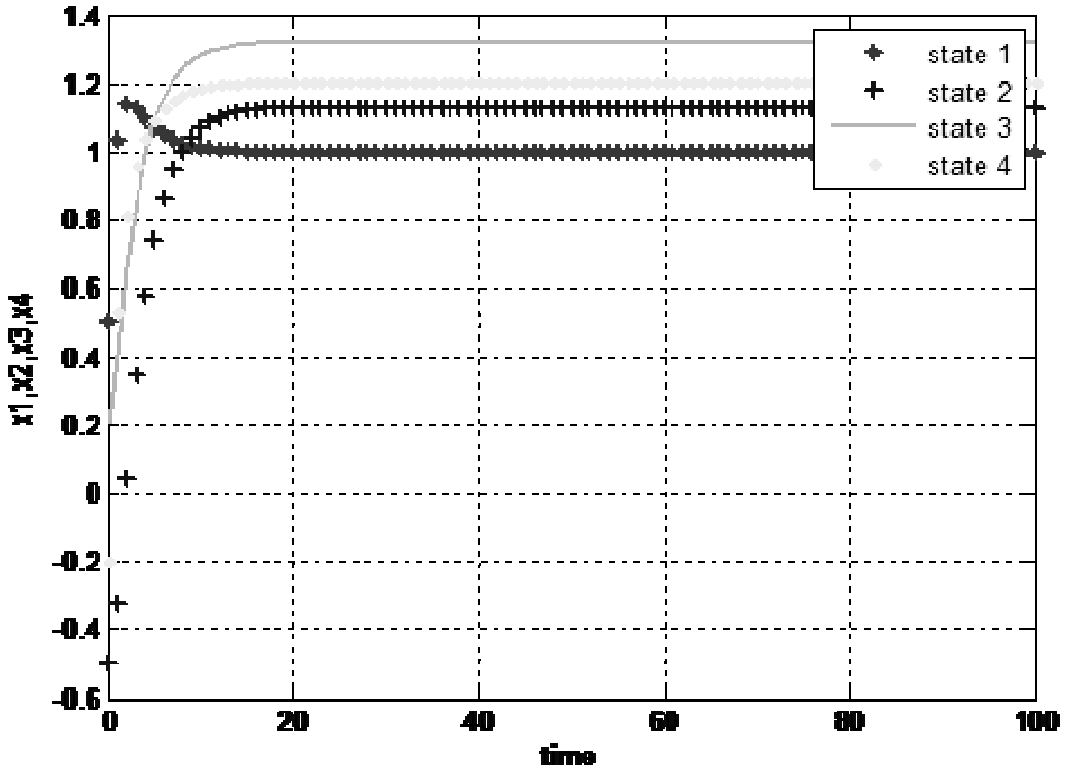

Figure 6: States for the constrained case.

is no way of exceeding them, except for fitting more powerful actuators. Thus input constraints are usually not softened.

A straightforward method of softening output constraints is to add new variables, *slake variables* that are defined in such a way that they are non-zero only if the constraints are violated. Then their non-zero values are heavily penalized in the cost function, so that the optimizer has a strong incentive to keep them at zero if possible. For further details see [13].

5. Simulation Results

To illustrate the algorithm described in previous section, we present an example. A randomly generated system has state space model parameters:

$$A = \begin{bmatrix} -0.1267 & -0.3357 & 0.0958 & -0.1723 \\ -0.4877 & 0.3487 & 0.0511 & 0.6393 \\ 0.0367 & 0.3482 & -0.0547 & -0.0399 \\ 0.0842 & -0.0110 & 0.2125 & 0.0334 \end{bmatrix};$$

$$B = \begin{bmatrix} 0.9501 \\ 0.2311 \\ 0.6068 \\ 0.4860 \end{bmatrix}; \; C = C_z = [1 \quad 0 \quad 0 \quad 0]$$

with $H_p = 5$; $H_u = 3$; $H_w = 1$; $Q > 0$; $R > 0$. We use the following constraints:

$$\begin{aligned} -10 \leq u(k) \leq 10 \\ - \leq \Delta u(k) \leq 2 \\ -3 \leq z(k) \leq 5 \end{aligned}$$

with initial state $x_0 = [0.5 \quad -0.5 \quad 0.2 \quad -0.2]$. The MPC controller was implemented and simulated with or without constraints using the methods outlined in section 3 and 4. Figures 3, 4, 5, and 6 illustrate the results. As can be seen from the figures, stability was achieved using the principle underlying MPC. Furthermore, in the presence of constraints, no violations occurred. Thus the existence of a feasible solution ensures constraint satisfaction. The cost function was decreasing with respect to time and is never less than zero. The minimum performance index, that is $V = 0$, is consistent with offset free tracking, as it implies that both error is zero and control is unchanging.

6. Conclusion

The aim of this paper was to give insight into model predictive control and run Matlab simulations to illustrate some of the theory for linear systems using a randomly generated system. MPC is popular due to its ability to yield high performance control systems capable of operating without expert intervention for long periods of time and also its ability to handle constraints. In this work, we have presented the importance of MPC, components of MPC, and mentioned some practical applications of MPC. MPC gives a controller that stabilizes the system being controlled due to the use of Lyapunov like function that ensures convergence.

References

1. Rismayasari, D., Joelianta, E. and Chaerani, D. The implementation of robust - optimization based model predictive control to waste heat boiler. *International Conference on Instrumentation, Control & Automation*, Bandung, Indonesia, 2009, pg 184 - 189.
2. Orukpe, P. E., Zheng, X., Jaimoukha, I. M., Zolotas, A. C. and Goodall, R. M. Model predictive control based on mixed control approach for active vibration control of railway vehicles. *Vehicle Systems Dynamics*, Vol. 46, Number 1, 2008, pg 151 – 160.
3. Garcia, C. E., Prett, D.M. and Morari, M. Model predictive control: theory and practice – a survey. *Automatica*, Vol. 25, Number 3, 1998, pg 335 – 348.
4. Bemporad, A. and Morari, M. Robust model predictive control: a survey. In Garulli, A., Tesi, A. and Vicino, A., Editors, *Robustness in Identification and Control*, Vol. 245 of Lecture notes in Control and Information Sciences, Springer-Verlag, 1999, pg 207 – 226.
5. Camacho, E. F. and Bordons, C. Model predictive control, Second Edition, Springer, 2004.
6. Mayne, D. Q., Rawlings, J. B., Rao, C.V. and Scokaert, P. O. M. Constrained model predictive control: stability and optimality, Automatica, Vol. 36, 2000, pg 789 814.
7. Maciejowski, J. M. Predictive control with constraints, Prentice Hall, 2002.
8. Rawlings, J. B. and Muske, K. R. The stability of constrained receding horizon control, IEEE Transactions on Automatic Control, Vol. 38, Number. 10, 1993, pg 1152 1156.
9. Qin, S. J. and Badgwell, T. A. A survey of industrial model predictive control technology, Control Engineering Practice, Vol. 11, 2003, pg 733 746.
10. Orukpe, P. E. Basics of model predictive control Imperial College London, Presentation, 2005.
11. Orukpe, P. E. A note on receding horizon control for linear discrete time systems subject to constraints Journal of Electrical and Electronic Engineering, Benin Nigeria, Vol. 10, Number 1, ISSN 1118-5058, 2006, pg 19 26.
12. Bartlett, R. A., Biegler, L. T., Backstorm, J. and Gopal, V. Quadratic programming algorithms for large scale model predictive control, Journal of Process Control, Vol. 12, 2002, pg 775 795.
13. Rao, C. V., Wright, S. J. and Rawlings, J. B. Application of interior point methods to model predictive control, Journal of optimization theory and application, Vol. 99, Number 3, 1998, pg 723 – 757.