Nonlinear MPC

C. Bordons and Eduardo F. Camacho

- Introduction
- Problem Formulation
- Modelling and Identification
- State Estimation
- Techniques for NMP
- Stability
- Open Issues

Introduction

Linear MPC:
- Nowadays: an available option in the market
- Best choice in certain industrial fields (petrochemical, glass, paper, etc.)
- Deep theoretical background

Although many processes are inherently nonlinear, the linear approach is used:
- The identification of a linear model from process data is relatively easy
- Linear models work well near the operating point (regulator problem)
- Linear model + quadratic cost function give rise to a convex QP (reliable market solvers)

However, there exist situations where the nonlinear effects justify other approach:
- Severe nonlinearities (even in the vicinity of the steady state), crucial to the closed-loop stability
  Such as pH control
- Frequent transitions (startups, shutdowns, etc.). Tracking problem usually away from a steady-state regime (polymers)
- Never in steady-state (batch processes)

In these cases, a linear control law will not be very effective. Need for Nonlinear Model Predictive Control, NMPC:
- Great potential. Still a limited number of reported applications
- NMPC has to make headway in those areas where process nonlinearities are strong and market requires regime changes
- It is justified to improve performance or even achieve stable operation

Motivation example

Nonlinear system
\[ y(t+1) = 0.9 y(t) + u(t)^{1/4} \]
with \( 0 < u(t) < 1 \).

Linear model:
\[ y(t+1) = 0.9 y(t) + u(t) \]

Results of a Linear-MPC and a NMPC with \( N=10, \Delta t=0 \)

Nonlinear MPC versus Linear MPC

Theoretically, the extension of MPC fundamentals to the nonlinear case is straightforward, but in practice, several problems appear:
- The availability of nonlinear models from experimental data is an open issue. There is a lack of identification techniques for nonlinear processes.
- Model attainment from first principles is not always feasible
- The optimization problem may be nonconvex. Resolution more difficult than QP. Problems related to local minimum and stability
- Increase in computation time (used in "slow" processes)
- The study of stability and robustness is more complex (open field)
**Problem formulation**

**Problem:** Computation of the control sequence $u$ that drive the process from its current state to a desired state $x_s, y_s, x_s, u_s$ given by the static optimization of an economic criterion.

Process (MIMO)

$$x(t+1) = f(x(t), u(t), d(t), w(t))$$

$$y(t) = g(x(t)) + e(t)$$

**Minimization of a generic objective function:**

$$J = \sum_{j=1}^{N-1} \left[ y_j - y_s \right]^2 + \sum_{j=1}^{N-1} \left[ u_j - u_s \right]^2 + \sum_{j=1}^{N-1} \left[ w_j - w_s \right]^2$$

Subject to model constraint

$$x(t+j) = f(x(t+j-1), u(t+j-1), d(t+j-1), w(t+j-1))$$

$$y(t+j) = g(x(t+j)) + e(t+j)$$

And the rest of constraints (hard, soft and terminal)

$$y_j - y_s \leq \bar{y}_j \leq \bar{y}_j$$

$$u_j - u_s \leq \bar{u}_j \leq \bar{u}_j$$

$$x_{N+1} = 0 \quad \delta \quad x_{N+1} \in W.$$
The greenhouse model considers a single layer cover. Takakura (1993) has discussed the selected greenhouse nonlinear model. The state vector $x(k)$ is formed of:
- $T_c$, cover temperature (measured)
- $T_i$, inside temperature (measured)
- $T_f$, floor temperature (measured)
- $T_p$, seedling temperature (estimated)

The perturbation vector $d(k)$ take into account:
- $R_R$, Direct solar radiation
- $R_d$, Diffuse solar radiation
- $T_o$, Outside temperature
- $T_{lb}$, Temperature under 10 cm-depth layer in the soil

Empirical Models

Fixed structure. Model parameters come from experimental data

1. Input/Output
   - NARMAX
     \[ y(t) = \Phi(y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u), e(t), \ldots, e(t-n_e+1)) \]
     - Volterra (FIR, bilinear)
     - Hammerstein
     - Wiener
     - NN

2. State Space
   \[ x(t+1) = f(x(t), u(t)) \quad y(t) = g(x(t)) \]
**I/O. Volterra models (family)**

\[ y(k + d) = h_0 + \sum_{i=1}^{m} h_i y(k - i) + \sum_{j=1}^{n} \sum_{i=1}^{m} h_{ij} x(k - j) y(k - i) \]

- Diagonal models: \( h_{ij} = 0 \) si \( i \neq j \)
- Auto-regressive

\[ A(q^-)p(k + d) = h_0 + \sum_{i=1}^{n} h_i p(k - i) + \sum_{j=1}^{n} \sum_{i=1}^{n} h_{ij} p(k - j) \]

- Models with static nonlinearities
  - Hammerstein
  - Wiener

**Example. Static nonlinearity**

Nonlinear static model with a linear dynamic part

Secon order dynamic model

Static nonlinearity modelled as a Neural Network

Training with historical data

\[ \delta u(t) = u(t) - u_a \]
\[ \delta y(t) = y(t) - y_a \]

\[ y_a = h_q(u_a) \]

**Application. Identification of a Gypsum Kiln**

- Variable gain
- Implicit dead time

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calcination Temp</td>
<td>Output Temp</td>
</tr>
</tbody>
</table>

**Volterra model for electromechanical systems**

- Underdamped dynamics
- AR model \( N_2 = 2 \)
- Quadratic terms \( N_2 = 4 \)

**Local Model Networks**

A set of models to accommodate local operating regimes

The output of each submodel is passed through a processing function that generates a window of validity.

\[ y(t+1) = F(\Psi(t), \Phi(t)) = \sum_{i=1}^{M} f_i(\Psi(t), \Phi(t), \rho_i(\Phi(t))) \]

Local models are usually linear multiplied by basis functions \( \rho_i(\Phi(t)) \) chosen to have a value close to 1 in regimes where is a good approximation and a value close to 0 in other cases.

Piece Wise Affine (PWA) systems

**Artificial Neural Network (ANN)**

Output: weighted sum of neuron outputs

Activation function
Neural Networks

- Multilayer Perceptron: a nonlinear function with good approximation properties and "backpropagation".
- Input-output with NN: \( y(t) = \text{NN}(y(t-1), ..., y(t-\text{ny}), u(t-1), ..., u(t-\text{nu})) \)
- State space with NN: \( x(t+1) = \text{NN}(x(t), u(t)); \quad y(t) = \text{NN}(x(t)) \)

PWA. (Sontag, 1981)

Nonlinear System:
\[ x(t+1) = f(x(t), u(t)) \]
\[ y(t) = g(x(t)) \]

PWA Approx.
\[
x_{k+1} = Ax_k + Bu_k + f_i
\]
\[
y_k = C^i x_k + g^i
\]
for \( \left[ \begin{array}{c} x_k \\ u_k \end{array} \right] \in X_i \quad X_i \triangleq \left\{ \left[ \begin{array}{c} x_k \\ u_k \end{array} \right] | Q^i \left[ \begin{array}{c} x_k \\ u_k \end{array} \right] \leq q^i \right\} \)

Some hybrid processes can be modeled by PWA system

Function

PWA Approximation

Approximation error

Solution of the NMPC Problem

Non Linear Programming (NLP). There exist software solvers, but with a high computation demand.

Some approaches or simplifications:
- Use of small horizons
- Suboptimal formulations (EPSAC)
- Iterative (Volterra)
- Feedback linearization
- Neural Control
- PWA (MIQP)
NONLINEAR MODEL FOR THE GREENHOUSE

The greenhouse model can be reduced to,
\[ x(k+1) = f(x(k)) + g(x(k))u + p(x(k))d(k) \]  
\[ \gamma = h(x(k)) \]

Where: \( x \) state vector, \( u \) control input, \( d \) disturbances vector, \( \gamma \) controlled output
\( f, g \) and \( p \) are smooth vector fields, and \( h \) is a smooth function

The transformed system can be expressed in normal form as:
\[ \xi(k+1) = A\xi(k) + Bv(k) \]  
\[ \eta(k+1) = T(\xi(k)), \eta(k) + E(\xi(k),\eta(k))d(k) \] 
\[ \gamma(k) = \xi(k) \]  

where \( A = 1, B = 3 \) min. We can note that the \( \eta \) remain Non-Linear.

Therefore, the control law linearizes the map between the transformed input \( v(k) \) and the output \( \gamma(k) \).

FEEDBACK LINEARIZATION OF THE GREENHOUSE MODEL

Control Objectives:

♦ The \( t_i \) should not surpass a 3°C band around the reference during the daytime.
♦ In order to save energy, during the night-time, the \( t_i \) may drop down to 10°C.

Reference and bounds were found using experimental results from the National Institute for Agriculture and Livestock Technology and Research (INTA). San Juan, Argentina.

MPC+FL APPROACH

Then the Objective Function is,
\[ J(k) = \sum_{i=1}^{NC} |r_{i(k+i)} - s_{i(k+i)}|^2 + \sum_{i=1}^{NC} |r_{i(k+i)} - s_{i(k+i)}|^2 \]
subjected to the linear system, \( \xi(k+1) = A\xi(k) + Bv(k) \) and the NL constraints \( v_{\text{min}} \) and \( v_{\text{max}} \).

Where:
\( J_{(i-1)i} \): Output predictive vector.
\( J_{(i+k)i} \): Reference trajectory.
\( v_{cp} \): Sequence of computed control
\( \omega, R \): Weights of states and control variable.
\( N_P, N_C \): Prediction and control horizon (NC>NP).
The desired trajectory was correctly tracked and the imposed constraints were fulfilled. In the three cases, the control objective was achieved.

Figure 3: MPC+FL (Outside Temperature in cyan)

Exp. 1: (red) $Q_d=1, R_d=10, Q_n=R_n=0$

Exp. 2: (blue) $Q_d=100, R_d=0.001, Q_n=R_n=0$

Exp. 3: (black) $Q_d=10, R_d=0, Q_n=R_n=0$

Figure 4: MPC+FL (red) vs. NLMPC (blue).

Both the output and control variables obtained by the NLMPC and MPC+FL techniques show a similar behavior.

The resulting optimization problem

$U = \{u(k), u(k+1), \ldots, u(k+N-1)\}$ real

$I = \{I(k), I(k+1), \ldots, I(k+N-1)\}$ Integer

Mixed Integer-Real Optimization Problem

Application of NMPC to a Labmate mobile robot: Path tracking in an unstructured environment

- Future trajectory known (computed by planner)
- Unexpected obstacles
- Control signals and state are constrained
- System model is highly nonlinear
- The objective function: position error, the acceleration, robot angular velocity and the proximity between the robot and the obstacles (detected with an ultrasound proximity system)
- Unexpected obstacles makes the objective function more complex.

(Gómez & Camacho, 1994)

Labmate Mobile robot model

$\theta(k+1) = \omega_r(k-1) + \omega_l(k-1) / 2$

$x(k+1) = x(k) + V / A \cdot (\sin(\theta) - \sin(\theta))$

$y(k+1) = y(k) - V / A \cdot (\cos(\theta) - \cos(\theta))$

$A = R (\omega_r(k-1) + \omega_l(k-1))$

$V = R (\omega_r(k-1) + \omega_l(k-1)) / 2$

$\theta$ is the steering speed, $\omega_r$ and $\omega_l$ are the right and left wheel angular velocities and the wheel radius.
Parameterization of desired path

- Necessary to calculate the \( N \) future path points.
- The desired point for the current instant is obtained first.
- The next \( N \) points are spaced equally on the path (desired speed).

Objective function

\[
J = \sum_{i=1}^{N} |x(t+i) - x_d(t+i)|^2 + l(x(t+i)) + |u(t+i+1) - u(t+i-1)|^2 + |u(t+i) - u_d(t+i-1)|^2
\]

Potencial function (two obstacles)

\[ J(u) \]

NMPC for mobile robot

NMPC implementation

Controller training

Mobile robot avoiding obstacles
Mobile robot NMPC

NMPC at a Solar plant
Almeria

Solar plant in Almeria
(PSA)

Distributed collectors

Plant diagram

Process model

Metal: \[ \rho_m \cdot C_m \cdot \frac{\partial T_m}{\partial t} = \eta_o \cdot I \cdot G - G \cdot H(T_m, T_J) - L \cdot H(T_m, T_J) \]

Fluid: \[ \rho_f \cdot C_f \cdot \frac{\partial T_f}{\partial t} + \rho_f \cdot C_f \cdot \frac{\partial T_m}{\partial x} = L \cdot H(T_m, T_J) \]

Simulink model can be downloaded from: http://www.esi2.us.es/~eduardo/libro-s/libro.html
**prediction: Solar radiation**

**NMPC computation**

**Plant results: a clear day**

Very fast setpoint tracking, little overshoot

**Plant results: a cloudy day**

**Example: Application to a chemical reactor**

Nonlinear behaviour: Exothermic chemical reaction

\[
\frac{dT}{dT} = \frac{\Delta h}{\Delta T} \left( T_{in} - T_{out} \right) + \left( -\frac{\Delta h}{\Delta C_p} \right) \frac{\Delta C_p}{\Delta T} e^{\frac{E_a}{RT}} C_{in}^{n-1} - \frac{\Delta h}{\Delta C_p} \frac{\Delta C_p}{\Delta T} e^{\frac{E_a}{RT}} C_{in}^{n-1} - \frac{\Delta h}{\Delta C_p} \frac{\Delta C_p}{\Delta T} e^{\frac{E_a}{RT}} C_{in}^{n-1}
\]

**Volterra model identification**

Identification with input/output data

Modelling and identification error
Volterra MPC (Iterative)

Related problems

- Model attainment (already seen)
- NLP solving:
  - Convexity is lost. Global minimum not guaranteed (QP cannot be used)
  - Computation time highly improved:
    - The NLP solver must evaluate the objective function and solve the nonlinear equation (prediction model)
    - The function gradient must be computed, apart from testing constraint violation and final algorithm condition
- Stability: it is not guaranteed by finding the optimum
- Robustness: difficult to be analysed

Good news: Evolution of the computational burden.

1998: 5th order models for distillation columns with a sampling time of 5 minutes [Allgower, Findeisen, 1998]
2001: Distillation column model of 206 order with a sampling time of 20 seconds [D. et al. ‘01]
2006: 5th order engine model with a sampling time of 20 milliseconds [Ferreau et al. ‘06]

5*60*1000 / 20 = 15 000 times faster,
Due to Moore’s law and algorithm improvement

Open Issues

- Modelling: nonlinear models are more complex than linear ones. The identification process is much more difficult. A great battery of tests is needed to capture nonlinearities.
- Problem resolution: the inclusion of the nonlinear model gives rise to a nonconvex optimization. Reliable and fast algorithms must be provided
- Stability and Robustness: further theoretical studies are needed.
- Justification of the required effort: the benefit obtained by using NMPC must be justified, since the effort is clearly bigger
- Other issues as in Linear MPC: tuning, multiobjective functions, fault tolerance, ill conditioning, etc.