CHEMICAL ENGINEERING TRANSACTIONS Volume 17, 2009
Editor Sauro Pierucci
Copyright © 2009, AIDIC Servizi S.r.l., ISBN 978-88-95608-01-3 ISSN 1974-9791
DOI: 10.3303/CET0917202

Generalized Class for Nonlinear Model Predictive Control Based on BzzMath Library

Flavio Manenti [1,*], Guido Buzzi-Ferraris [1], Ivan Dones [2], Heinz A. Preisig [2]

[1] Politecnico di Milano, CMIC Dept. "Giulio Natta"
Piazza Leonardo da Vinci, 32 – 20133 Milano, ITALY

[2] Chemical Engineering Department, NTNU
Sem Sælands vei 4, N-7491 Trondheim, NORWAY

A generalized C++ class to solve nonlinear model predictive control (MPC) and dynamic optimization problems is proposed. Since optimal control problems involve (i) differential equations systems to foresee plants and/or process units dynamics and (ii) constrained optimization issues to meet process specs and requirements, *BzzMath* library is adopted as kernel to numerically face these tasks. The proposed class allows both FORTRAN and C++ users to easily solve MPC and dynamic optimization by defining their differential system and the desired objective function only, without taking care of any numerical problem that may occur in integrating differential systems, in searching for the minimum of a constrained/complex objective function, and in implementing a moving horizon methodology.

1. Introduction

The twofold aim in studying a generalized class for solving optimal control problems is the need of finding an efficient solution for the supply chain management problem as well as to propose and validate a freely downloadable tool to support users in settling nonlinear model predictive control (MPC) and business-wide dynamic optimization.
Actually, it is field-proven that MPC methodology is one of the most promising approaches to ensure flexible and profitable production with an economic optimization of plant operations, that is reduction of downtimes, product waste, and raw materials cost impact, is carried out meeting process constraints and guaranteeing a reliable control system. Notwithstanding, the industrial state of art in process control is still represented by the linear MPC, even though nonlinear applications have significantly increased in number in the ast five years (for more details, see also Bauer and Craig (2008) and Qin and Badgwell (2003)). The delay in nonlinear applications is especially due to the traditional inertia of process industries in acquiring and applying new technologies and even to the lack of free tools to approach multifaceted optimal control problems. *BzzMath* library (Buzzi-Ferraris, 2009a) offers a solid and reliable numerical kernel to develop a generalized class for nonlinear MPC applications. *BzzMath* library is briefly presented in Section 2. Section 3 discusses MPC structure. Section 4 shows the

[*] Corr. author. Ph.: +39 02 2399 3273; Fax: +39 02 7063 8173; Email: flavio.manenti@polimi.it

Please cite this article as: Manenti F., Buzzi-Ferraris G., Dones I. and Preisig H.A., (2009), Generalized class for nonlinear model predictive control based on bzzmath library, Chemical Engineering Transactions, 17, 1209-1214 DOI: 10.3303/CET0917202

implementation of generalized class for MPC. Finally, the proposed tool is validated in Section 5.

2. BzzMath Library as Kernel

This research activity is based on *BzzMath* library, which is freely downloadable at Professor Buzzi-Ferraris's homepage. *BzzMath* is a numerical library entirely written in C++ by adopting object-oriented programming (Buzzi-Ferraris, 1994). It covers several scientific fields such as linear algebra, linear/nonlinear regressions, optimization, differential systems, *etc.*, and some of them are reported hereinafter.

2.1 Linear Algebra

Linear algebra is the essential basis to solve numerical problems. Gill *et al.* (1991) said:

"The importance of numerical linear algebra in modern
scientific computing cannot be overstated"

BzzMath is predisposed not only to solve linear systems, but even to automatically adopt the most performing algorithm in accordance with the type of system to be solved. Being developed in object-oriented way, objects belonging *BzzMath* library can easily identify and, if possible, exploit matrix sparsity and matrix structure of linear systems, by making very performing their solution (Buzzi-Ferraris and Manenti, 2009b). Furthermore a relevant error that still affects other numerical libraries has recently been fixed (Buzzi-Ferraris, 2009b).

2.2 Regression Models

Parameter estimations, outlier detections, model discriminations, and design of experiments are well-known hard problems. *BzzMath* includes specific classes based on very robust algorithms to solve linear and nonlinear regression problems (Buzzi-Ferraris and Manenti, 2009a) and to detect masking effects, heteroscedasticity, parameter correlations, and gross errors (Manenti and Buzzi-Ferraris, 2009). A methodology to discriminate among rival models and, at the same time, to define the optimal design of experiment is even implemented in *BzzMath*; correct meanings of statistical tests and confidence region have recently been redefined (Buzzi-Ferraris and Manenti, 2009c).

2.3 Nonlinear Systems and Optimization

Starting from OPTNOV's variant (Buzzi-Ferraris, 1967) up to the most recent improvements, numerically robust and efficient algorithms are implemented for solving nonlinear systems and optimization problems. On this subject, two examples can be quoted: very large-scale nonlinear systems (sparse blocks matrix with a number of equations in the order of some tens of millions) to characterize a kinetic post-processor (Cuoci *et al.*, 2007), and a constrained multi-scale and multi-objective optimization (diagonal block matrix with unstructured elements) of a polymer plant (Manenti and Rovaglio, 2008) were both solved by means of *BzzMath* library.

2.4 Differential and Differential-Algebraic Systems

At last, *BzzMath* library includes reliable and very performing algorithms for solving ordinary differential equations (ODE) systems and differential and algebraic equations

(DAE) systems (Buzzi-Ferraris and Manca, 1998). Again, its object-oriented structure gives the possibility to significantly reduce the computational time in integrating differential systems, apart from their stiffness. Many applications of *BzzMath* solvers are proposed in the scientific literature. In addition, *ad hoc* solvers to tackle partially structured DAE systems, typical of process control and process systems engineering, have recently been introduced into the numerical library (Manenti *et al.*, 2009).

3. Class Architecture

The spreading of nonlinear model predictive control in these last decades has been mainly dictated by the following reasons: (i) it is intrinsically able to manage nonlinearities in process dynamics and in profits; (ii) it can be based on first-principles mathematical models and on nonlinear semi-empirical models as well; (iii) it allows solving simultaneously the predictive control (quadratic problem) and the dynamic optimization (economic problem).

The basic architecture of a model predictive control application is reported in *Fig. 1*. Assuming an on-line implementation of this technique, the plant provides data to the model predictive control at each sampling time. Specifically, the plant data are sent to the optimizer, which includes an objective function, a dynamic model and, usually, according to the mathematical model type, a numerical integrator to solve specific differential systems, such as ordinary differential, differential-algebraic, partial differential, and partial differential-algebraic equations systems.

If the real time dynamic optimization has to be solved as well, economical data and market scenarios have to be provided to the MPC structure and one or more economical objective functions should be defined.

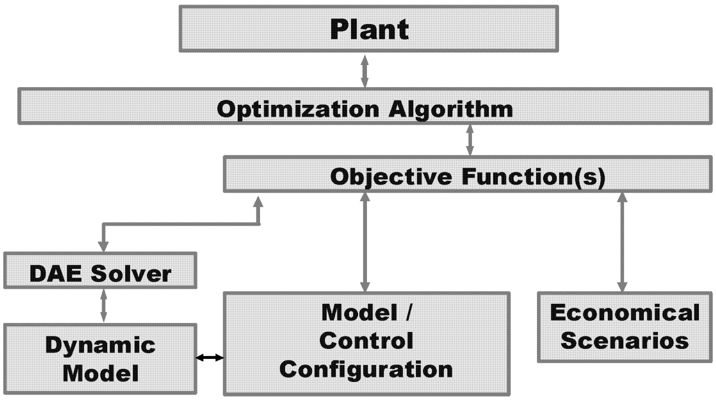

Fig. 1: architecture of nonlinear model predictive control.

3.1 Objective Function

Each optimal control problem can be brought back to the minimization of a weight least squares objective function subject to equality and/or inequality constraints. In the specific case of model predictive control, the generalized formulation is usually the following:

$$\min_{\hat{\mathbf{u}}(k)\ldots\hat{\mathbf{u}}(k+h_c-1)} \left\{ \sum_{j=k+1}^{k+h_p} w_y \left[\hat{\mathbf{e}}_y(j)\right]^2 + \sum_{l=k}^{k+h_p-1} w_{u_1} \left[\Delta\hat{\mathbf{u}}(l)\right]^2 + \sum_{i=k}^{k+h_p-1} w_{u_2} \left[\Delta\hat{\mathbf{u}}(i)\right]^2 \right\}$$

$$\text{subject to:} \begin{cases} u_i^{\min} \le u\left(k+i\,|\,k\right) \le u_i^{\max} \\ \Delta u_i^{\min} \le \Delta u\left(k+i\,|\,k\right) \le \Delta u_i^{\max} \\ -e + y_i^{\min} \le y\left(k+i+1\,|\,k\right) \le y_i^{\max} + e \\ \Delta u\left(k+j\,|\,k = 0\right), \quad j = m,\ldots,p \\ e \ge 0 \\ \textit{differential system} \end{cases} \tag{1}$$

being $\hat{\mathbf{e}}_y(j) = \hat{\mathbf{y}}(j) - \mathbf{y}_{set}(j)$ the gap between controlled variables and their set-points; $\Delta\hat{\mathbf{u}}(l) = \hat{\mathbf{u}}(l) - \hat{\mathbf{u}}_{tar}(l)$ the gap between manipulated variables and their targets; and $\Delta\hat{\mathbf{u}}(i) = \hat{\mathbf{u}}(i) - \hat{\mathbf{u}}(i-1)$ the incremental variations of manipulated variables. w_y, w_{u_1}, w_{u_2} are multiplicative coefficients (weights); $\hat{\mathbf{y}}_{set}(j)$ are set-points of the controlled variables at the $j-th$ time-interval; $\hat{\mathbf{u}}_{tar}(l)$ are targets of the controlled variables at the $j-th$ time-interval; $\hat{\mathbf{y}}(j)$ are controlled variables at the $j-th$ time-interval; $\hat{\mathbf{u}}(l)$ are manipulated variables at the $l-th$ time-interval; h_p is the prediction horizon; h_c is the control horizon; and *differential system* represents constraints dictated by mathematical model of the plant/process unit to be controlled.

3.2 The Algorithm

The aforementioned formulation can be converted into an algorithm based on differential solvers, optimizers, and outliers detection methods belonging to *BzzMath* library. First of all, raw data acquired from the plant should be treated in order to detect any gross error or bad quality measure. An opportune class to reconcile raw data set, which is based on $\mathbf{QR}$ factorization and linear systems solution, can be adopted; for sake of conciseness, this aspect is not described in this paper (for more details, see also Buzzi-Ferraris and Manenti, 2009b). Reconciled data is then used to initialize MPC structure: the optimizer is called the first time to evaluate the best manipulated variables $\mathbf{u}$ by minimizing the objective function (1). To do so, all the equality and inequality constraints (including the differential system) have to be evaluated and an opportune differential solver should be invoked. The differential system is then integrated on a specific prediction horizon h_p in order to predict future system behaviour according to different values of $\mathbf{u}$. After an iterative procedure, the optimal vector $\mathbf{u}$ is implemented in the plant and new data are acquired to restart the cycle.

4. Class Validation

The class was validated on different ODE and DAE systems already proposed in literature. Specifically, by adopting polyethylene terephthalate DAE model proposed by Manenti and Rovaglio (2008), characterized by a diagonal blocks structure, the results obtained by the generalized class corroborate previous trends obtained by more complex procedural structure. *Fig. 2* shows the trends for intrinsic viscosity (IV) and pressure (P) in both the intermediate (IP) and high polymerizer (HP) during a grade change production.

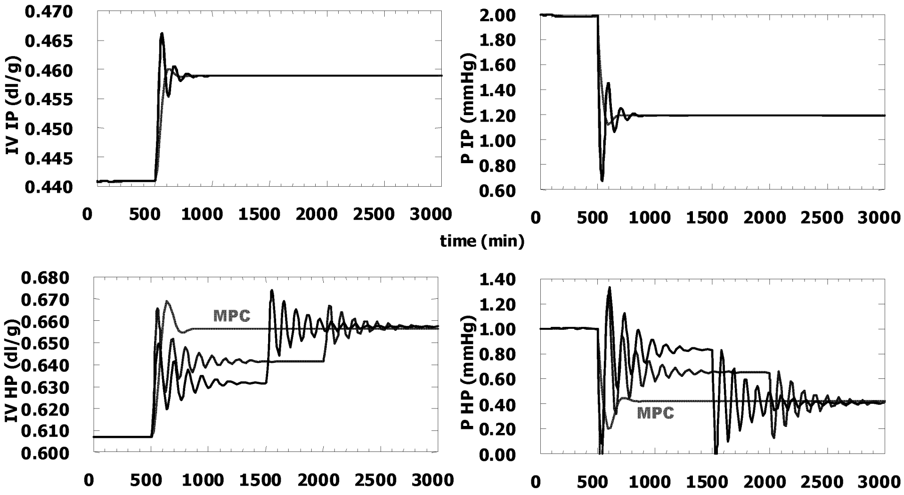

Fig. 2: nonlinear MPC applied to PET plant and compared to conventional control.

In addition, the class was successfully validated on batch models (Abel and Marquardt, 2003), by introducing an additional Boolean logic to manage discontinuous operations.

5. Conclusions

A generalized class for solving nonlinear MPC and dynamic optimization problems based on *BzzMath* library was proposed and validated on different case studies. Encouraging results were obtained not only for continuous applications but even for batch units. Such a class allows the implementation of nonlinear MPC once an adequate objective function and a differential system are defined. It can be easily used in FORTRAN and C++ code without the need to worry about differential solvers and optimization algorithms, since the object-oriented nature of the class and the philosophy adopted in *BzzMath* library synergistically allow an automatic selection of appropriate algorithms to solve these issues.

Appendix – Class Implementation: an Available Constructor

Given an objective function and a differential(or differential-algebraic) system, the MPC can be invoked through the constructor:

```
BzzModelPredictiveControl NMPC(hp,hc,y0,u0,ad,DinSys,FObj,uL,uU);
```

where `hp` is the prediction horizon; `hc` the control horizon; `y0` the reconciled measures acquired by the field for each MPC call; `u0` the initial values of manipulated variables; `ad` is an integer vector for discriminating between algebraic (`ad(i)=0`) and differential (`ad(i)=1`) equations of the system described in the function `DinSys`; `FObj` is the weight least squares objective function; optionally, `uL` and `uU` are minimum and maximum constraints, respectively, of manipulated variables. For more details, see also www.chem.polimi.it/homes/gbuzzi.

FORTRAN users can refer to Buzzi-Ferraris and Manenti (2009a) for a detailed description and examples of C++ class implementation in FORTRAN environment.

7. References

Abel O., & Marquardt W. (2003). Scenario-integrated on-line optimisation of batch reactors, Journal of Process Control, 13(8), 703-715.

Bauer, M., & Craig, I.K. (2008). Economic Assessment of Advanced Process Control - A Survey and Framework. Journal of Process Control, 18, 2-18.

Buzzi-Ferraris, G. (1967). Ottimizzazione di funzioni a più variabili. Nota I. Variabili non vincolate. Ing. Chim. It., 3, 101.

Buzzi-Ferraris, G. (1994). Scientific C++. Addison-Wesley, Cambridge University Press.

Buzzi-Ferraris, G. (2009a). BzzMath: Numerical library in C++. Politecnico di Milano, http://chem.polimi.it/homes/gbuzzi.

Buzzi-Ferraris, G. (2009b). Revise your Linear System Solver. Computers & Chemical Engineering, submitted.

Buzzi-Ferraris, G., & Manca, D., (1998) BzzOde: a new C++ class for the solution of stiff and non-stiff ordinary differential equation systems. Computers & Chemical Engineering, 22(11), 1595-1621.

Buzzi-Ferraris, G., & Manenti, F. (2009a). Interpolation and Regression Models for the Chemical Engineering Solving Numerical Problems. WILEY-VCH, Weinheim, Germany, 2009 (in press).

Buzzi-Ferraris, G., & Manenti, F. (2009b). Fundamentals and Linear Algebra for the Chemical Engineering Solving Numerical Problems. WILEY-VCH, Weinheim, Germany, 2009 (in press).

Buzzi-Ferraris, G., & Manenti, F. (2009c). Kinetic Models Analysis. Chemical Engineering Science, 64(5), 1061-1074.

Cuoci A, Frassoldati A, Ferraris GB, Faravelli T., Ranzi E. (2007). The ignition, combustion and flame structure of carbon monoxide/hydrogen mixtures. Note 2: Fluid dynamics and kinetic aspects of syngas combustion. International Journal Of Hydrogen Energy, 32(15), 3486-3500.

Manenti, F., & Buzzi-Ferraris, G. (2009). Criteria for Outliers Detection in Nonlinear Regression Problems. Proceedings of ESCAPE-19, Cracow, Poland, to appear.

Manenti, F., Dones, I., Buzzi-Ferraris, G., & Preisig, H.A. (2009). An Efficient Numerical Solver for Partially Structured DAE Systems. Industrial & Engineering Chemistry Research, submitted.

Manenti, F., & Rovaglio, M. (2008). Integrated multilevel optimization in large-scale poly(ethylene terephthalate) plants. Industrial & Engineering Chemistry Research, 47(1), 92-104.

Qin, S.J., & Badgwell, T.A. (2003). A survey of industrial model predictive control technology. Control Engineering Practice, 11(7), 733-764.