

OptCon – an Efficient Tool for Rapid Prototyping of Nonlinear Model Predictive Control Applications

Z. K. Nagy

*Chemical Engineering Department, Loughborough University, Loughborough, LE11 3TU,
United Kingdom, email: z.k.nagy@lboro.ac.uk*

Introduction

Nonlinear model predictive control is an optimization-based multivariable constrained control technique that uses a nonlinear dynamic process model for the prediction of the process outputs. At each sampling time the model is updated on the basis of new measurements and state variable estimates. Then the open-loop optimal manipulated variable moves are calculated over a finite prediction horizon with respect to some cost function, and the manipulated variables for the subsequent prediction horizon are implemented. Then the prediction horizon is shifted by usually one sampling time into the future and the previous steps are repeated. NMPC is one of the approaches which inherently can cope with process constraints, nonlinearities and different objectives formulated from economical or environmental considerations. Although the advantages of NMPC have been demonstrated through numerous simulation studies the gap between the number of simulation papers and actual practical implementations is significant. This gap is even more pronounced if consider real industrial applications. The talk will briefly assess the major implementation aspects of NMPC introducing the *integral NMPC design concept*. Figure 1 illustrates the *NMPC pentagon*, which constitutes the basis of the integral NMPC design. All sides of the pentagon have to be considered for successful practical implementation.

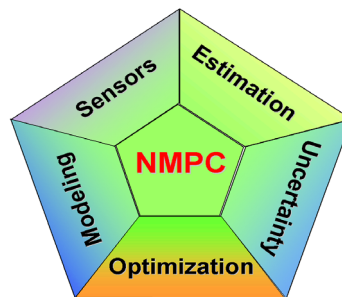


Fig.1. “NMPC Pentagon” that illustrates the main aspects to be considered in an integral design of a practical NMPC.

OPTCON – A Toolbox for rapid NMPC prototyping

A Matlab toolbox is illustrated which has been developed with the aim to provide a rapid prototyping tool for NMPC applications. The NMPC approach presented in this talk is a first principles model based approach that can be used for the solution of moving or shrinking

horizon (batch) control problems. The NMPC tool developed includes a number of desirable features. In particular the NMPC setup can be done in Matlab[®] (The Mathworks Inc.) the most widely used modeling and control environment for control engineers. The model used in the controller has to be developed in the form of Simulink[®] (The Mathworks Inc.) “mex S-function” using C. The S-function interface of the optimization tool provides convenient and fast connectivity with Matlab[®]. The NMPC approach is based on a state-of-the-art large-scale nonlinear optimization solver (HQP) [2], which offers one of the most efficient approaches, based on a multiple shooting algorithm [1], that exploits the special structure of optimization problem that arise in NMPC. In this algorithm the optimization horizon of interest is divided into a number of subintervals (stages) with local control parameterizations. The differential equations and cost on these intervals are integrated independently during each optimization iteration, based on the current guess of the control. The continuity/consistency of the final state trajectory at the end of the optimization is enforced by adding consistency constraints to the nonlinear programming problem. This multiple shooting formulation concept is shown in Figure 2.

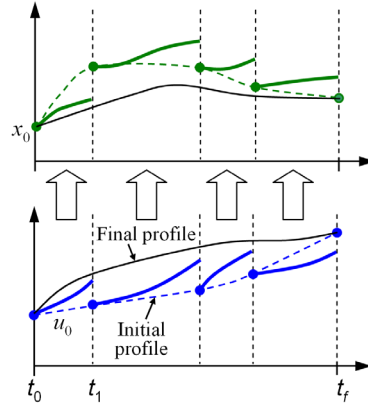


Figure 2. Illustration of the multiple shooting approach.

The optimal control problem to be solved on-line in every sampling time in the NMPC algorithm can be formulated as:

Problem *PI*:

$$\min_{u(t) \in \mathcal{U}} \mathcal{H}(x(t), u(t); \theta), \quad (1)$$

subject to:

$$\dot{x}(t) = f(x(t), u(t); \theta), \quad (2)$$

$$y(t) = g(x(t), u(t); \theta), \quad (3)$$

$$x(t_k) = \hat{x}(t_k), \quad x(t_0) = \hat{x}_0, \quad (4)$$

$$h(x(t), u(t); \theta) \leq 0, \quad t \in [t_k, t_F] \quad (5)$$

where \mathcal{H} is the performance objective, t is the time, t_k is the time at sampling instance k , t_F is the final time at the end of prediction, $x(t) \in \mathbb{R}^{n_x}$ is the n_x vector of states, $u(t) \in \mathcal{U}$ is the n_u set of input vectors, $y(t) \in \mathbb{R}^{n_y}$ is the n_y vector of measured variables used to compute the

estimated states $\hat{x}(t_k)$, $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ is the n_θ vector of possible uncertain parameters, where the set Θ can be either defined by hard bounds or probabilistic, characterized by a multivariate probability density function. The function $f: \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathbb{R}^{n_x}$ is the twice continuously differentiable vector function of the dynamic equations of the system, $g: \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathbb{R}^{n_y}$ is the measurement equations function, and $h: \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathbb{R}^c$ is the vector of functions that describe all linear and nonlinear, time-varying or end-time algebraic constraints for the system, where c denotes the number of these constraints. The objective function can have the following general form:

$$\mathcal{H}(x(t), u(t); \theta) = \mathcal{M}(x(t_f); \theta) + \int_{t_k}^{t_f} \mathcal{L}(x(t), u(t); \theta) dt. \quad (6)$$

We assume that $\mathcal{H}: \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathbb{R}$ is twice continuously differentiable, thus fast optimization algorithms, based on first and second order derivatives may be exploited in the solution of (6). The form of (6) is general enough to express a wide range of objectives encountered in NMPC applications (moving or shrinking horizon approach on regulation and/or setpoint tracking, direct minimization of the operation time, optimal initial conditions, multiple simultaneous objectives, treatment of soft constraints, terminal penalty term for stability, etc.). For batch processes with end-point optimization the objective usually reduces to the Mayer form ($\mathcal{L}(\cdot) = 0$), however the Lagrange term ($\mathcal{L}(\cdot)$) still may be used, e.g. to implement soft constraints on control rate. In NMPC the optimization problem (1)-(5) is solved iteratively on-line, in a moving (receding) horizon ($t_F < t_f$) or shrinking horizon ($t_F = t_f$) approach, where t_f is the batch time.

Considering the discrete nature of the on-line control problem, the continuous time optimization problem involved in the NMPC formulation is solved by formulating a discrete approximation to it, that can be handled by conventional nonlinear programming (NLP) solvers [6]. The time horizon $t \in [t_0, t_f]$ is divided into N equally spaced time intervals Δt (stages), with discrete time steps $t_k = t_0 + k\Delta t$, and $k = 0, 1, \dots, N$. Model equations are discretized,

$$x_{k+1} = f_k(x_k, u_k; \theta) \quad (7)$$

and added to the optimization problem as equality constraints. Problem *P2* gives the general discrete time formulation of Problem *P1*.

Problem *P2*:

$$\min_{u_k, u_{k+1}, \dots, u_{k+N_p}} \{ \mathcal{M}^{k+N_p}(x_{k+N_p}; \theta) + \sum_{j=k}^{k+N_p} \mathcal{L}_j(x_j, u_j; \theta) \}, \quad (8)$$

subject to:

$$G_k(x_k, u_k; \theta) = 0, \quad (9)$$

$$H_k(x_k, u_k; \theta) \leq 0, \quad (10)$$

where N_p is the number of stages in the prediction horizon $[t_k, t_F]$, $G_k: \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathbb{R}^{n_x+n_y}$ corresponds to all equality constraints resulted from (7) and (3), $H_k: \mathbb{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathbb{R}^{c+2n_x}$ is the vector function of all inequality constraints (5), including the constraints on the inputs considering. We consider here that the set of possible inputs is given as hard bounds. It is assumed that the vector functions G and H are twice continuously differentiable.

Real time iteration NMPC algorithm

In NMPC simulation studies usually immediate feedback is considered, i.e. the optimal feedback control corresponding to the information available up to the moment t_k , is computed, $u^*(t_k) = [u_{0|t_k}, u_{1|t_k}, \dots, u_{N_p|t_k}]$, and the first value ($u_{0|t_k}$) is introduced into the process considering no delay. However, the solution of the NLP problem P2 requires a certain, usually not negligible, amount of computation time δ_k , while the system will evolve to a different state, where the solution $u^*(t_k)$ will no longer be optimal. Computational delay δ_k has to be taken into consideration in real-time applications. In the approach used here, in moment t_k , first the control input from the second stage of the previous optimization problem $u_{1|t_{k-1}}$ (which corresponds to the first stage of the current optimization) is injected into the process, and then the solution of the current optimization problem is started, with fixed $u_{0|t_k} = u_{1|t_{k-1}}$. After completion, the optimization idles for the remaining period of $t \in (t_k + \delta_k, t_{k+1})$, and then at the beginning of the next stage, at moment $t_{k+1} = t_k + \Delta t$, $u_{1|t_k}$ is introduced into the process, and the algorithm is repeated. This approach requires real-time feasibility for the solution of each open-loop optimization problems ($\delta_k \leq \Delta t$).

This multistage formulation is advantageous to the efficient application of the SQP-type solver in HQP, and often leads to faster solution of the optimization problem due to the better approximation of the Lagrangian Hessian of the nonlinear subproblems using low rank updates. HQP uses a sparse interior point algorithm for the efficient treatment of the linear-quadratic subproblems in the nonlinear SQP iterations. The main structure of the OptCon software is shown on Figure 3.

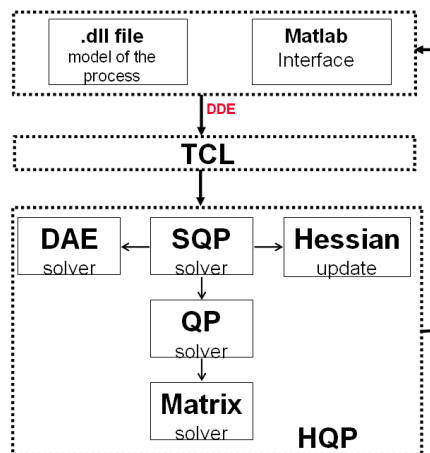


Fig. 3. Structure of the main components of the NMPC toolbox - OptCon

Some of the features and available options of the toolbox are:

- Lagrangian Hessian approximation
 - BFGS – dense BFGS updates for separate diagonal blocks
 - DScale – numerical approximation with a diagonal matrix
 - Gerschgorin – positive definiteness guaranteed by a diagonal offset
- QP solver:
 - Use of barrier method
 - Mehrotra – primal-dual predictor-corrector method
 - Franke – based on interior point algorithm from Wright
- QP matrix solver to solve the extended Karush-Kuhn-Tucker conditions:
 - SpBKP – solution of the system
 - RedSpBKP – solution of the reduced system
 - SpSC – sparse Schur complement method
 - LQDOCP – block – wise solution of the multistage problems.

Available DAEA/ODE solvers: RKsuite – explicit ODE solver; Euler – explicit ODE solver, fixed step; RK4, IMP – implicit ODE solver; DASPK – implicit DAE solver; SDIRK – implicit DAE solver; dt – fixed integration step size.

The optimization problems are configured by setting the required weights in the predetermined objective functions. There are two precompiled objective functions included in the toolbox using the SFunctionOpt and SFunctionEst function from HQP.

The optimization time horizon $[t_0, t_f]$ is split into $k = 0, \dots, KK$ stages $KK = \text{sps} \cdot K$ sample periods, ($\text{sps} = \#$ of samples per stage)

The optimization problem:

$$\begin{aligned}
 J = & \sum_{kk=0}^{KK} \Delta t^{kk} \sum_{i=1}^{n_u} \left\{ u_{weight1} u(t^{kk}) + u_{weight2} \left[u(t^{kk}) - \frac{u_{ref}}{u_{nominal}} \right]^2 \right\}_i + \\
 & + \sum_{kk=0}^{KK-1} \Delta t^{kk} \sum_{i=1}^{n_u} \left\{ der_u_{weight1} \dot{u}(t^{kk}) + der_u_{weight2} \left[\dot{u}(t^{kk}) - \frac{der_u_{ref}}{u_{nominal}} \right]^2 \right\}_i + \\
 & + \sum_{kk=0}^{KK} \Delta t^{kk} \sum_{i=1}^{n_y} \left\{ y_{weight1} y(t^{kk}) + y_{weight2} \left[y(t^{kk}) - \frac{y_{ref}}{y_{nominal}} \right]^2 \right\}_i + \\
 & + \sum_{i=1}^{n_y} \left\{ y_{f_weight1} y(t_f) + y_{f_weight2} \left[y(t_f) - \frac{y_{ref}}{y_{nominal}} \right]^2 \right\}_i \rightarrow \min
 \end{aligned}$$

The estimation problem is formulated as

$$J = \sum_{l=1}^{N_{ex}} \sum_{k=k_{l,v}}^{K_l} \sum_{i=1}^{\dim(l)} \left\{ \left[\frac{y_{s,kl}(I) - y_{s,ref}}{y_{nominal}(I)} \right]^2 \right\}_i \rightarrow \min, I = find(y_{active}),$$

Some of the main functions available in OPTCON are:

- f_optHQP – optimization/optimal control
- f_estHQP – state and parameter estimation
- f_NMPCCHQP – NMPC function
- f_MHEHQP – MHE estimation
- f_rtsim – real-time process simulator

Additional functions which provide implementation templates for NMPC and MHE applications are also available.

The use of the estimation function is illustrated in Figure 4

```

1 EXP.nexp      = 3;
2 EXP.nspexp   = nspex; % No of samples per experiment
3 EXP.times    = times;
4 EXP.y_s_measur = ys_measur;
5 EXP.us       = mdl_us;
6
7 ESTCONFIG.model      = 'sfc_KEplant_withtime_est';
8 ESTCONFIG.p_guess   = [5100 104.9 9758.3*0.8 8560.0*1.2];
9 ESTCONFIG.mdl_p_active = [0 0 1 1];
10 ESTCONFIG.mdl_p_min  = [1000 50 7000 7000];
11 ESTCONFIG.mdl_p_max  = [1e4 500 1e4 1e4];
12
13 ESTCONFIG.mdl_x0s    = [3910.0*0.8, 570.0*0.9, 3.6700*1.1, 74.4400*1.1, 0
14                        3910.0*1.2, 570.0*1.1, 3.6700*0.9, 74.4400*0.9, 0
15                        3910.0*1.1, 570.0*0.15, 3.6700*0.8, 74.4400*0.8, 0];
16
17 ESTCONFIG.mdl_x0_active = [1 1 1 1 0];
18 ESTCONFIG.mdl_x0_min    = [2000 400 1 50 0];
19 ESTCONFIG.mdl_x0_max    = [5000 700 10 100 0];
20
21 ESTCONFIG.mdl_y_active  = [1 0 1 0 0 0 0];
22
23 ESTOPTIONS.plot_res     = 1;
24 ESTOPTIONS.SQP_maxiter = 150;
25 ESTOPTIONS.prg_integrator = 'DASPK';
26
27 EST_RESULTS = f_estHQP(EXP, ESTCONFIG, ESTOPTIONS)
  
```

Define Experiments (lines 1-5)
Configure estimation problem (lines 7-21)
Set Options (lines 23-25)
Call Function (line 27)

Figure 4. Matlab script illustrating the use of the estimation function on OptCon./

OptCon has been successfully applied in a variety of NMPC applications. The application of OptCon to a laboratory four tank system will be illustrated briefly, which is the first practical application of a theoretically well founded NMPC approach (the so called quasi-infinite horizon NMPC), that guarantees stability. Figure 5 and 6 illustrate these results [4].

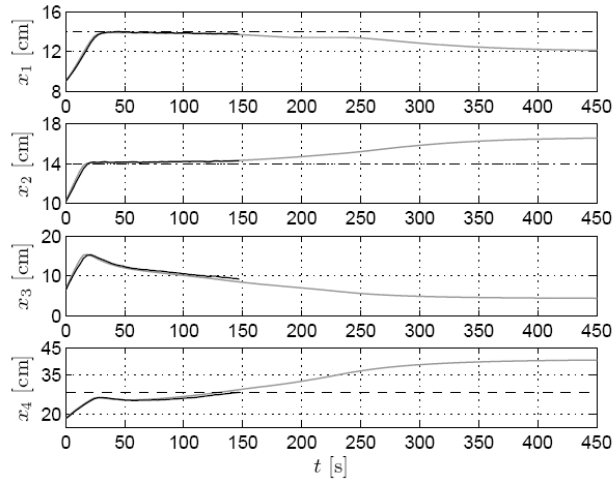


Figure 5. NMPC without stability and state constraints. Setpoint (black dash-dotted), simulation (grey-solid) and experiment (black solid). The figure illustrates that the system cannot be stabilized at the origin without considering stability constraints in the problem formulation.

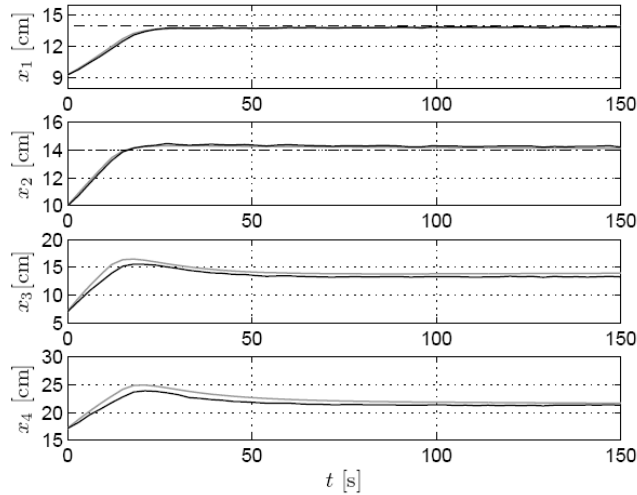


Figure 6. Quasi infinite horizon NMPC with prediction horizon of 60s: set-point (black dash-dotted), simulation (grey-solid) and experiment (black solid). The figure illustrates that the system is stabilized at the origin using the stability constraints.

Next we illustrate how the same tool can be used for prototyping industrial NMPC applications and go from simulation studies to the actual practical tests within minutes. The quality of the product strongly depends on how tightly the temperature profile is tracked during the batch. Disturbances caused by the effects of exothermic reaction cause overshooting when classical cascade-PI control is used, with an increasing degradation in control performance when batch time is reduced. It is shown that NMPC can provide higher control performance for setpoint tracking. Additionally, simulations show that product

quality can significantly decrease in case of certain disturbance scenarios. The NMPC approach proposed here is a hierarchical (two level) NMPC approach, in which a higher level NMPC controller reoptimizes on-line the setpoint, based on the desired end-point properties of the product. The setpoint is sent to a lower level, setpoint-tracking NMPC. The results show that with this approach good temperature tracking is achieved and the product quality at the end of the batch is well controlled. Figure 7 illustrates the temperature tracking results of the NMPC controller implemented to an industrial pilot plant [3].

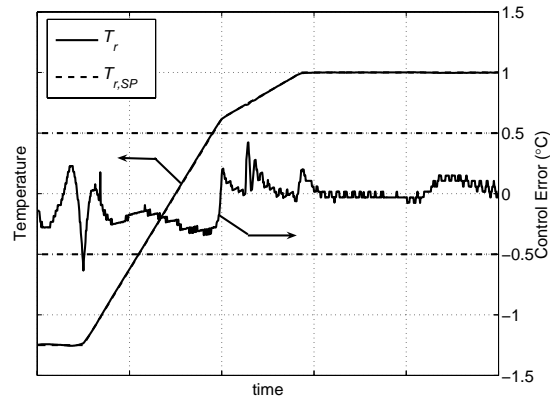


Figure 7. Experimental data with the setpoint tracking performance of the NMPC in the BASF pilot plant. The reactor temperature almost perfectly follows the setpoint (within ± 0.5 °C).

Conclusions

The paper described the main features of a new user friendly tool, OptCon, which can provide an efficient environment for rapid prototyping of industrial nonlinear model predictive control applications. The tool provides a Matlab based connectivity between the efficient optimizations package HQP and Matlab. The package is distributed as a freeware application and can be obtained by contacting the author at (z.k.nagy@lboro.ac.uk).

References

- [1] Diehl, M., H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, F. Allgöwer (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations, *Journal of Process Control*, 12, 577-585.
- [2] Franke, R., E. Arnold and H. Linke. HQP: A solver for nonlinearly constrained large-scale optimization. <http://hqp.sourceforge.net>.
- [3] Nagy, Z. K., B. Mahn, R. Franke, F. Allgöwer, Efficient output feedback nonlinear model predictive control for temperature control of industrial batch reactors, *Control Engineering Practice*, 15, 839-859, 2007.
- [4] Raff, T., S. Huber, Z. K. Nagy, F. Allgöwer, Nonlinear model predictive control of a four tank system: an experimental stability study, in Proc. of IEEE International Conference on Control Applications, Munich, Germany, 237-242, October 4-6, 2006.