SIMULATION-BASED DUAL MODE CONTROLLER FOR NONLINEAR PROCESSES

Jong Min Lee [*] **Jay H. Lee** [*,1]

[*] *311 Ferst Dr., School of Chemical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0100, USA*

Abstract: This paper presents a simulation-based strategy for designing a nonlinear override control scheme to improve the performance of a local linear controller. The higher-level nonlinear controller monitors the dynamic state of the system under the local controller and sends an override control action whenever the system is predicted to move outside an acceptable operating regime under the local controller. For this purpose, a cost-to-go function is defined, an approximation of which is constructed by using simulation or historic operation data. The cost-to-go function delineates the "admissible" region of state space within which the local controller is effective, thereby yielding a switching rule. The same cost-to-go function can also be used to calculate override control actions designed to bring the system state back into the admissible region as quickly as possible. One potential problem of this approach is the lack of robustness when the simulation data sparsely cover the state space and the data-based approximation of the cost-to-go function is extrapolated to a region previously unseen. Hence, successful application of the proposed method requires safeguarding against undue extrapolations. For this reason, a kernel-based local approximation, instead of a global approximator like a neural network, is used to interpolate the cost-to-go values. It is shown that the kernel-based local regression provides convenient means to implement a risk-sensitive control scheme which avoids excessive extrapolation. The proposed scheme is demonstrated and discussed with nonlinear examples.

Keywords: Simulation-Based Approach, Nonlinear Predictive Control, Switching Controller, Kernel-Based Approximator, Cost-to-Go Function

1. INTRODUCTION

Model predictive control (MPC) is being widely used in the process industry because of its ability to control multivariable processes with hard constraints. Most of the current commercial MPC solutions are based on linear dynamic models, which are easier in terms of identification and online computation (Qin and Badgewell, 1997). On the other hand, many chemical processes exhibit strong nonlinearities. This disparity has prompted several studies on MPC formulations with nonlinear system models (Lee, 1997). Since most Nonlinear MPC (NMPC) formulations require online solution of a nonlinear program (NLP), issues related to computational efficiency and stability of a control algorithm have received much attention.

The initial focus was on formulating a computationally tractable NMPC method with guaranteed stability. Mayne and Michalska (1990) showed that stability can be guaranteed by introducing a terminal state equality constraint at the end of

[1] to whom correspondence should be addressed: jay.lee@che.gatech.edu

prediction horizon. In this case, the value function for the NMPC can be shown to be a Lyapunov function under some mild assumptions. Because the equality constraint is difficult to handle numerically, Michalska and Mayne (1993) extended their work to suggest a dual-mode MPC scheme with a local linear state feedback controller inside an elliptical invariant region. This effectively relaxed the terminal equality constraint to an inequality constraint for the NMPC calculation. The dual-mode control scheme was designed to switch between the NMPC and the linear feedback controller depending on the location of the state. Chen and Allgöwer (1998) proposed a quasi-infinite horizon NMPC, which solves a finite horizon problem with a terminal cost and a terminal state inequality constraint. The main difference from the Michalska and Mayne's method is that a fictitious local linear state feedback controller is used only to determine the terminal penalty matrix and the terminal region off-line and switching between controllers is not required.

These NMPC schemes have theoretical rigor but have some practical drawbacks. First, these methods still require solving a multi-stage nonlinear program at each sample time. Assurance of a globally optimal solution or even a feasible solution is difficult to guarantee. Second, the optimization problem for determining the invariant region for a local linear controller and the corresponding terminal weight are both conservative and computationally demanding.

Motivated by the drawbacks and the industry's reluctance to adopt full-blown NMPC, we propose an override (or supervisory) control strategy for monitoring and improving the performance of a local controller. Our method is similar to the dual-mode MPC suggested by Michalska and Mayne in that both switch between two different control policies depending on current location of the state. However, we employ a cost-to-go function based approach instead of NMPC. First a *cost-to-go* function under the local controller is defined, which serves to delineate the admissible region within which the local controller can effectively keep the system inside acceptable operating limits. The same cost-to-go function is also shown to facilitate the calculation of override control actions that will bring the system outside the admissible region back into the region as quickly as possible. We propose to use simulation or historic data to construct an approximation to the cost-to-go function. With the cost-to-go function, an override control action can be calculated by solving a single stage nonlinear optimization problem, which is considerably simpler than the multi-stage nonlinear program solved in the NMPC. One potential problem of using the cost-to-go values approximated using simulation data is that it is only accurate within regions where data existed. Hence, in the on-line calculation, one has to safeguard against unreasonable extrapolation of the cost-to-go function approximator. This leads to a *risk-sensitive* control scheme, where the quality of approximation gets reflected in the cost-to-go value. In this paper, we propose to use a local regression based on Gaussian kernel in order to implement the risk-sensitive control, which avoids unreasonable extrapolations.

2. SIMULATION-BASED CONSTRUCTION OF AN OVERRIDE CONTROLLER

The proposed scheme uses either simulation or actual plant data to identify the region of the state space, in which the local controller can effectively keep the system inside an acceptable operating regime (defined by some inequalities in the state space). We do this by assigning to each state a 'cost-to-go' value, which is defined as

$$J^{\mu}(x_0) = \sum_{i=0}^{\infty} \alpha^i \phi(x_i) \tag{1}$$

where $J^{\mu}(x_0)$ is the cost-to-go for state x_0 under the local control policy $u = \mu(x)$, $0 < \alpha < 1$ is a discount factor, and $\phi(x_i)$ is a stage-wise cost that takes the value of 0 if the state at time i is inside the acceptable operating limit and 1 if outside when x_0 is the state at time 0. This way, if a particular state x_0 under the control policy evolves into a state outside the limit *in some near future* under the policy μ, the cost-to-go value will reflect it. On the other hand, those states that are not a precursor of future violation of the operating limit will have a negligible cost-to-go value. The latter states comprise the "admissible" region.

The cost-to-go function is approximated by first simulating the closed-loop behavior of the nonlinear model under the local linear controller for various possible operating conditions and disturbances. This generates x vs. $J^{\mu}(x)$ data for all the visited states during the simulation. Then the generated data can be interpolated to give an estimate of $J^{\mu}(x)$, $\tilde{J}^{\mu}(x)$, for any given x in the state space.

In the real-time application, whenever the process reaches a state with a significant cost-to-go value, it is considered to be a warning sign that the local controller's action will not be adequate. When this happens, an override control action is calculated and implemented to bring the process back to the "admissible" region where the cost-to-go is insignificant. One can calculate such an action by implementing the override policy of

$$
\begin{aligned}
\text{if} \qquad & \tilde{J}^{\mu}(x_{t+1}(x_t, \mu(x_t))) \geq \eta \qquad , \\
u_t &= \arg\left(\min_{u'_t} \tilde{J}^{\mu}(x_{t+1}(x_t, u'_t))\right)
\end{aligned} \tag{2}
$$

where η is a user-given threshold value for triggering the override control scheme. If no u'_t can be found such that $\tilde{J}^{\mu}(x_{t+1}(x_t, u'_t)) < \tilde{J}^{\mu}(x_{t+1}(x_t, \mu(x_t)))$, then $u_t = \mu(x_t)$ is used for the current sample time.

3. A KERNEL-BASED APPROXIMATOR OF COST-TO-GO FUNCTION

In this paper, we propose to use a local regression instead of the usual choice of a feedforward neural network to approximate the cost-to-go values. Empirical studies show that general approximators (e.g. neural network) are not good choices for the approximation of cost-to-go function due to the high nonlinearity and discontinuity of the cost-to-go function in general (Boyan and Moore, 1995). In addition, Gordon (1995) showed that the local averager with non-expansive property (*e.g.* kernel-based approximation) is compatible with dynamic programming operator and effective for representing local characteristics of state spaces.

Another reason for adopting the local regression approach is our concern for grossly incorrect cost-to-go estimates that can arise from extrapolating to a region not accounted for in the simulation step. In implementing a risk-averse 'cost-to-go' based controller, Kaisare *et al.* (2002) used a feedforward neural network but gridded the state space in order to separate regions visited by simulation from those not. For those cells with little or no data, a high cost-to-go value was assigned to prevent the controller from driving the state trajectory into these uncertain regions. However, this is difficult to implement for cases with high-dimension state spaces.

For a convenient implementation of the risk-averse or rist-sensitive scheme, we propose to use a variation of Gaussian-kernel-based approximators. This structure decides whether a reliable estimate can be given to a query point based on the available data. For a "reliable" query point it gives local weights calculated from a Gaussian kernel to give more influence over the regression to those training points closer to the query point than those farther away. The suggested structure of kernel-base prediction is

$$
\hat{f}(x_0) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)} \tag{3}
$$

where

$$
K_\lambda(x_0, x_i) = \exp\left(-\frac{\|x_0 - x_i\|_2^2}{\lambda^2}\right) \tag{4}
$$

The number of neighbor points N is the number of data points inside a hypersphere, the radius of which is a user-given value r. In addition to r, there are other parameters that user should provide. These are the Gaussian kernel width λ, minimum number of data points inside the hypersphere k_{min}, and the high cost-to-go value J_h to be assigned to an "unreliable" query point. Table 1 describes how the estimate of cost-to-go value for a query point is calculated.

Table 1. "Risk-averse" prediction using Gaussian-kernel-based approximator

Prediction Algorithm
1. Is the query point x_0 in the memory?
a. Yes: Use the value in the memory.
b. No: Go to step 2.
2. Enumerate the data points inside r around the x_0.
Is the number of data points greater than k_{min}?
a. Yes: Average with the kernel.
a. No: $J(x_0)$ cannot be estimated. Assign J_h to x_0.

4. ILLUSTRATIVE EXAMPLES

4.1 Simple Nonlinear Example

4.1.1. Problem Description We consider a system with two states, one output, and one manipulated input described by

$$
\begin{aligned}
x_1(k+1) &= x_1^2(k) - x_2(k) + u(k) \\
x_2(k+1) &= 0.8\exp\{x_1(k)\} - x_2(k)u(k) \\
y(k) &= x_1(k)
\end{aligned} \tag{5}
$$

with an equilibrium point of $x_{eq} = (-0.3898, 0.5418)$, $u_{eq} = 0$.

We also define the *acceptable* operating regime by

$$
\begin{aligned}
W(x) = \left\{(x_1 - x_{1eq}) + \sqrt{3}(x_2 - x_{2eq})\right\}^2 + \\
\left\{\frac{(x_2 - x_{2eq}) - \sqrt{3}(x_1 - x_{1eq})}{0.3}\right\}^2 - 4 \leq 0
\end{aligned} \tag{6}
$$

A linear MPC controller was designed based on a linearized model around the equilibrium point. The control objective is to regulate y to y_{eq}. The linear MPC is used as the local controller with the following design.

$$
\min_{\Delta u} \sum_{i=1}^{p} 5\bar{y}^2(k+i) + \sum_{l=0}^{m-1} \Delta\bar{u}^2(k+l) \tag{7}
$$

with $p = 2$ and $m = 1$.

$$
\begin{aligned}
-3 \leq \bar{u} \leq 3 \\
\Delta\bar{u} \leq 0.2
\end{aligned} \tag{8}
$$

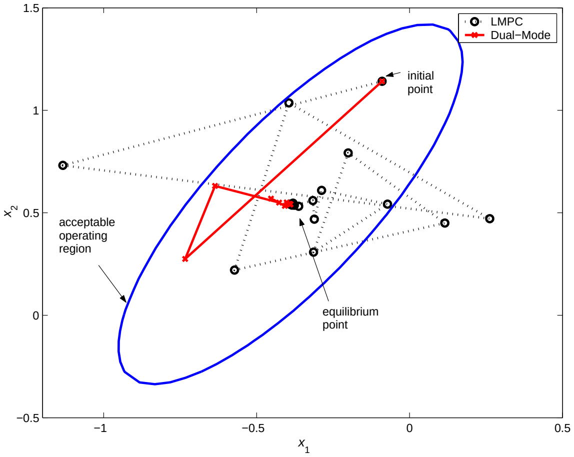

Fig. 1. State trajectories under local MPC and dual-mode controller, $x_0 = [-0.0898\ 1.1418]$

The closed-loop behavior under the local controller starting at $x_0 = x_{eq} + [0.3\ \ 0.6] = [-0.0898\ 1.1418]$ is shown as dotted lines in Fig. 1. Though the initial point is inside the operating limit, the system under the local linear controller violates the limit several times until the system is regulated to the equilibrium point.

4.1.2. Simulation-Based Design To design the proposed override controller, closed-loop simulations under the local controller were performed using 347 initial points inside the operating limit. The simulations generated 17006 data points and cost-to-go values for each state in the trajectory were calculated using Equation (1) with a value of $\alpha = 1$ and

$$\phi(x_t) = \begin{cases} 1 \text{ if } W(x_{1t}, x_{2t}) \leq 0 \\ 0 \text{ if } W(x_{1t}, x_{2t}) > 0 \end{cases} \tag{9}$$

Next step is to design a Gaussian-kernel approximator. Considering the coverage of state space, following parameters were chosen: $r = 0.05$, $k_{min} = 3$, $\lambda = 0.03$, $J_h = 30$.

The actual value of cost-to-go is zero for the states inside the admissible region of a linear controller and outside the region the cost-to-go will be over unity. This makes the structure of cost-to-go function very stiff. However, the approximator will smoothen out the stiff structure a bit by averaging. Therefore small tolerance value ($\eta = 0.02$) was chosen to illustrate a possible shape of the admissible region under the local controller, which is illustrated in Fig. 2.

4.1.3. Real-Time Application To compare online performances of the local controller alone and the dual mode controller (i.e., the local controller combined with the proposed override controller), eight initial points different from the training set were sampled. We also compare the proposed dual-mode controller with the successive linearization based MPC (SLMPC) scheme suggested by Lee and Ricker (1994). Finally, we also simulated the LMPC and the SLMPC with the state constraints of $-0.95 \leq x_1 \leq 0.2$ and $-0.35 \leq x_2 \leq 0.45$ (denoted by scLMPC and scSLMPC). The prediction and control horizons of SLMPC are the same as those of the LMPC.

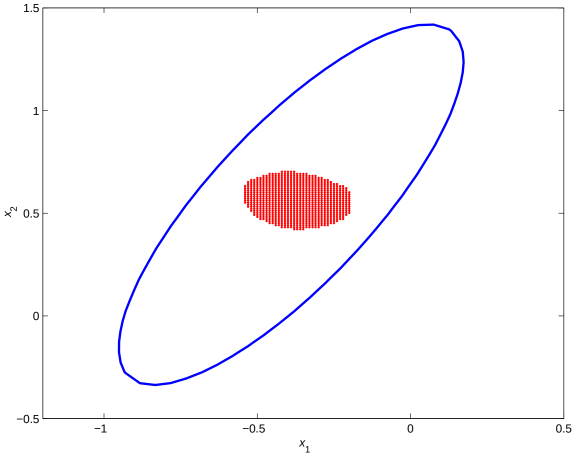

Fig. 2. Regions under local controller with $\tilde{J}(x) < 0.02$

The solid lines in Fig. 1 is the state trajectory with the same initial point under the dual-mode controller. For the first three points, the override control actions were used instead of those of LMPC's. The proposed scheme successfully steers the state back to the region with lower cost-to-go values. Table. 2 shows the sum of stage-wise cost (the total number of violation of operating limit) and the suggested control design outperforms for all the test points. We can also see that imposing state constraints did not work here as many infeasible solutions were returned, eventually causing divergence.

Table 2. Comparison of performances (total # of limit violations)

Test pt	LMPC	SLMPC	scLMPC	scSLMPC	Override
1	div.	5	div.	div.	0
2	3	3	div.	div.	0
3	2	0	0	div.	0
4	2	0	0	div.	0
5	0	0	div.	div.	0
6	0	0	0	div.	0
7	7	15	1	div.	0
8	div.	div.	div.	div.	0

4.2 Bioreactor Example

In this section, we consider a bioreactor example with two states: biomass and substrate (Bequette, 1998). With a substrate inhibition for growth rate expression of biomass, the system shows multiple steady states. To operate at the unstable equilibrium, closed-loop control must be used. The system equation is:

$$\begin{aligned}\frac{dx_1}{dt} &= (\mu - D)x_1 \\ \frac{dx_2}{dt} &= D(x_{2f} - x_2) - \frac{\mu x_1}{Y} \\ \mu &= \frac{\mu_{max} x_2}{k_m + x_2 + k_1 x_2^2}\end{aligned} \tag{10}$$

where x_1 is biomass concentration and x_2 is substrate concentration. Table 3 shows the parameters for the model at the unstable steady state.

Table 3. Model parameters: bioreactor example

μ_{max}	0.53 hr^{-1}	k_m	0.12 g/l
k_1	0.4545 l/g	Y(yield)	0.4
D_s, x_{2fs}	0.3hr^{-1}, 4.0 g/l	x_s	[0.9951 1.5123]

4.2.1. Local Linear Controller A linear MPC was designed based on a linearized model around the unstable equilibrium point with sample time of 0.1h. The control objective is to regulate x to x_s at the equilibrium values and the manipulated variables are the substrate concentration in the feed x_{2f} and the dilution rate D. The LMPC controller parameters we used are $Q = 100I, R = 10I, p = 10$, and $m = 5$, where I is a 2 by 2 identity matrix, Q is a state weighting matrix, and R is an input weighting matrix.

We also define an acceptable operating region as

$$W(x) = \left\{\frac{0.52(x_1 - x_{1eq}) + 0.85(x_2 - x_{2eq})}{7}\right\}^2 + \left\{\frac{-0.85(x_1 - x_{1eq}) + 0.52(x_2 - x_{2eq})}{0.5}\right\}^2 - 1 \leq 0 \tag{11}$$

which is shown in Fig.3. The input constraints for MPC is

$$\begin{array}{ll} 0 \leq D \leq 0.5 & |\Delta D| \leq 0.2 \\ 0 \leq x_{2f} \leq 8 & |\Delta x_{2f}| \leq 2 \end{array} \tag{12}$$

The closed-loop behavior under the LMPC for different initial points are shown in Fig. 3. As in the previous example, the LMPC cannot drive the state back into the equilibrium point without violating the operating limit.

4.2.2. Simulation-Based Dual Mode Controller With the same definition of one-stage cost as in Equation (9), a cost-to-go-based override controller was designed. For the simulation, 109 initial points were sampled inside the operating limit and closed-loop simulations under the LMPC yielded 21909 points. Parameters for a kernel-based approximator were chosen as: $r = 0.1$, $k_{min} = 5$, $\lambda = 0.05$, $J_h = 50$, $\eta = 0.02$.

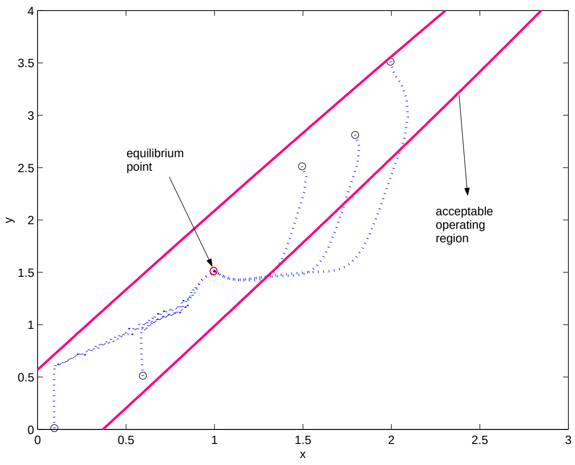

Fig. 3. State trajectories under local MPC

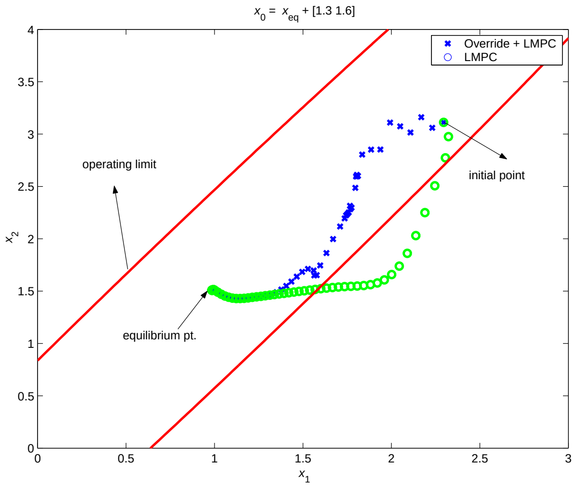

Fig. 4. State trajectory under dual-mode controller

As in the previous example, the dual mode controller successfully navigated the state to the equilibrium point without violating the operating limit by searching for the path with lowest cost-to-go values. One of the sample trajectories tested is shown in Fig. 4.

5. EVOLUTIONARY IMPROVEMENT OF COST-TO-GO

Because the approximator employed in the calculation of override control action is based on the cost-to-go value of the local linear controller, it is not the optimal cost-to-go. The resulting override controller from the suboptimal cost-to-go approximation is also suboptimal. Hence, further improvement of the override control policy to steer the system back into the admissible region of the linear controller is possible by iteratively solving

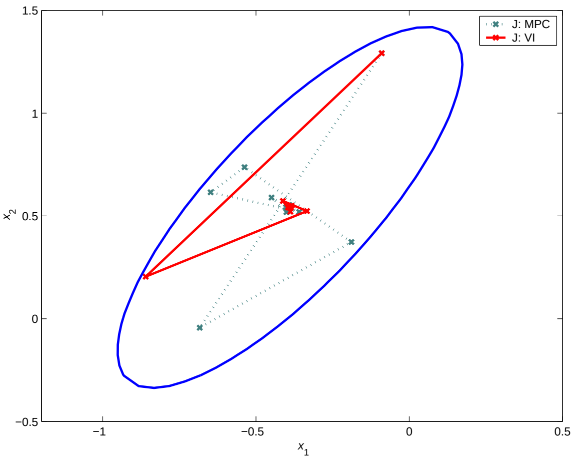

Fig. 5. State trajectory with the dual-mode controller using improved cost-to-go

the following optimality equation (as in *value-iteration*) until $\tilde{J}$ converges.

$$\tilde{J}^{i+1}(x) = \min_{u} \left[\phi^{i}(x) + \tilde{J}^{i+1}(f(x,u)) \right] \quad (13)$$

where f is a state transition equation and i denotes iteration index.

For this purpose, the one-stage cost is re-defined differently as

$$\phi^{i}(x) = \begin{cases} 1 & \tilde{J}^{i}(x) \geq \eta \\ 0 & \tilde{J}^{i}(x) < \eta \end{cases} \quad (14)$$

With this change, the aim of the optimal control is to bring the system state back into the "admissible" region as quickly as possible.

The value iteration was performed for the first illustrative example and the iteration converged after 5 steps with the following convergence criterion.

$$\|\tilde{J}^{i+1}(x) - \tilde{J}^{i}(x)\|_{\infty} < 0.1 \quad (15)$$

Fig. 5 shows one of the state trajectory with the initial point of $x_0 = x_{eq} + [\,0.3 \quad 0.75\,]$ when the improved cost-to-go function is used in the override control calculation. As shown in the figure, the improved override controller bring the state back into the admissible region more efficiently than that based on the cost-to-go approximation under the LMPC.

6. CONCLUSION

A simulation-based override control scheme was shown to improve the performance and stability of a given local controller. The ease of design and implementation makes it a potentially appealing addition to an existing controller in industrial applications. The suggested framework can give operators indications on the future performance of the local controller and also suggest override control actions, if needed. More realistic situations such as the case with plant/model mismatch will be studied next.

Acknowledgement: The authors gratefully acknowledge the financial support from the National Science Foundation (CTS-#0096326).

REFERENCES

Bequette, B. W. (1998). *Process Dynamics: Modeling, Analysis, and Simulation.* Prentice Hall. Upper Saddle River, New Jersey.

Boyan, J. A. and A. W. Moore (1995). Generalization in reinforcement learning: Safely approximating the value function. In: *Advances in Nueral Information Processing Systems: Proceedings of the 1994 Conference.* MIT Press. Cambridge, MA. pp. 369–376.

Chen, H. and F. Allgöwer (1998). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica* **34**(10), 1205–1217.

Gordon, G. J. (1995). Stable function approximation in dynamic programming. In: *Proceedings of the Twelfth International Conference on Machine Learning.* Morgan Kaufmann. San Francisco, CA.

Kaisare, Niket S., Jong Min Lee and Jay H. Lee (2002). Simulation based method for nonlinear optimal control: Application to a microbial cell reactor. *International Journal of Robust and Nonlinear Control.*

Lee, J. H. (1997). Recent advances in model predictive control and other related areas. In: *Chemical Process Control – Assessment and New Directions for Research* (Y. C. Kantor, C. E. Garcia and B. Carnahan, Eds.). Vol. 93. AIChE Symposium series. pp. 201–216.

Lee, J. H. and N. L. Ricker (1994). Extended Kalman filter based nonlinear model predictive control. *Ind. Eng. Chem. Res.* **33**(6), 1530–1541.

Mayne, D. Q. and H. Michalska (1990). Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control* **35**(7), 814–824.

Michalska, H. and D. Q. Mayne (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control* **38**(11), 1623–1633.

Qin, S. J. and T. A. Badgewell (1997). An overview of industrial model predictive control technology. In: *Chemical Process Control - V.* American Insitute of Chemical Engineers. New York. pp. 232–256.