Neurocomputing 129 (2014) 570–584

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Explicit nonlinear predictive control algorithms with neural approximation

Maciej Ławryńczuk *

Institute of Control and Computation Engineering, Faculty of Electronics and Information Technology, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland

ARTICLE INFO

Article history:
Received 15 May 2013
Received in revised form
3 September 2013
Accepted 8 September 2013
Communicated by Prof. R. Tadeusiewicz
Available online 16 October 2013

Keywords:
Process control
Model Predictive Control
Neural networks
Approximation
Optimisation
Soft computing

ABSTRACT

This paper describes two nonlinear Model Predictive Control (MPC) algorithms with neural approximation. The first algorithm mimics the MPC algorithm in which a linear approximation of the model is successively calculated on-line at each sampling instant and used for prediction. The second algorithm mimics the MPC strategy in which a linear approximation of the predicted output trajectory is successively calculated on-line. The presented MPC algorithms with neural approximation are very computationally efficient because the control signal is calculated directly from an explicit control law, without any optimisation. The coefficients of the control law are determined on-line by a neural network (an approximator) which is trained off-line. Thanks to using neural approximation, successive on-line linearisation and calculations typical of the classical MPC algorithms are not necessary. Development of the described MPC algorithms and their advantages (good control accuracy and computational efficiency) are demonstrated in the control system of a high-purity high-pressure ethylene-ethane distillation column. In particular, the algorithms are compared with the classical MPC algorithms with on-line linearisation.

1. Introduction

Model Predictive Control (MPC) refers to a control strategy in which a dynamic model is used on-line to predict future behaviour of the process [9,26,37,39,42]. The current value of the control signal is repeatedly calculated on-line from an optimisation problem in which, typically, the discrepancy between the predicted output trajectory and the desired reference trajectory over some time horizon is minimised. Although the simplest classical MPC algorithms use for prediction only linear models, they have been successfully used for years in numerous advanced industrial applications [36]. It is because they have a few important advantages. Firstly, constraints imposed on process inputs (manipulated variables) and outputs (controlled variables) or state variables can be easily taken into account, directly in the MPC optimisation problem. In real control systems satisfaction of constraints is crucial because constraints usually determine quality, economic efficiency and safety. Secondly, MPC techniques can be efficiently used for multivariable process and for processes with difficult properties (e.g. with significant time-delays, with the inverse step response).

In spite of the fact that the classical linear MPC algorithms quite frequently give good control accuracy, the majority of technological processes are nonlinear by nature. In such cases, when linear MPC algorithms do not give required control performance, nonlinear MPC algorithms based on nonlinear models must be used [15,30,42]. In nonlinear MPC different nonlinear model structures can be used, e.g. Nonlinear Auto Regressive with eXternal input (NARX) polynomial models [41], cascade Wiener [8] and Hammerstein [13] models (a linear dynamic part connected in series with a nonlinear steady-state one), Volterra models [12], Support Vector Machines models [18]. An alternative is to use artificial neural networks [11,25,31,32,40,42], which is very interesting because they offer very good approximation accuracy, have a simple structure and, unlike some of the aforementioned model types, usually have a reasonably low number of parameters.

If for prediction in MPC a nonlinear neural model is used directly, the future predictions of the output variable are nonlinear functions of the calculated control policy. As a result, the MPC optimisation problem is a nonlinear task which must be repeatedly solved on-line in real time. Although very high computational complexity of the MPC approach with nonlinear optimisation is evident, its simulations or applications to different real processes can be found in the literature. Example applications include: a continuous pulp digester [3], a forced-circulation evaporator [4], a polymerisation reactor [17], a fuel-ethanol fermentation process [28] and a fluid catalytic cracking unit [44]. For on-line nonlinear

* Tel.: +48 22 234 76 73; fax: +48 22 825 37 19.
E-mail address: M.Lawrynczuk@ia.pw.edu.pl

0925-2312/$ - see front matter
http://dx.doi.org/10.1016/j.neucom.2013.09.002

optimisation the Sequential Quadratic Programming (SQP) algorithm is usually used, but applications of an evolutionary strategy [34] and a simulated annealing algorithm [1] are also reported.

In order to reduce the computational cost, suboptimal MPC algorithms may be used. In the simplest approach a linear approximation of the neural model is successively calculated on-line at each sampling instant and used for prediction [25,42]. Thanks to linearisation, the MPC optimisation task becomes a quadratic programming problem, which can be solved very efficiently. Example applications include: the autopilot for the F-16 aircraft [2], a solar plant [7], a spark-ignited engine [11,40], a polymerisation reactor and a distillation column [25], chemical extractors [29] and an aircraft gas turbine engine [31]. Alternatively, a linear approximation of the predicted output trajectory may be successively calculated on-line once at each sampling instant [19] or repeated a few times to increase prediction accuracy [20], which also leads to quadratic programming. The MPC approaches with on-line linearisation in practice usually give good control accuracy, very similar, or even the same as the MPC algorithm with full nonlinear optimisation repeated at each sampling instant on-line (comparisons are given, e.g. in [19,20,25,42]). An interesting idea is to use a neural network of a specialised structure for solving the quadratic programming problem in place of a numerical procedure [33].

It is possible to further reduce computational burden associated with the suboptimal MPC algorithms with on-line linearisation. If the constraints are removed from the quadratic programming MPC optimisation problem, it is only necessary to carry out a matrix decomposition task and solve linear equations at each sampling instant, quadratic optimisation is not necessary. Next, the obtained solution is projected onto the admissible set determined by constraints. Implementation details of the explicit MPC algorithm with on-line linearisation and simulation results obtained in the control system of a distillation column are presented in [23]. An alternative is to use approximate MPC algorithms in which the whole control algorithm is replaced by a neural network whose role is to calculate the control signal directly, without any optimisation [5,10,35]. The neural approximator is trained off-line to mimic the MPC algorithm with on-line nonlinear optimisation.

This paper describes two explicit nonlinear MPC algorithms with neural approximation: the MPC algorithm with Nonlinear Prediction and Linearisation and Neural Approximation (MPC-NPL-NA) and the MPC algorithm with Nonlinear Prediction and Linearisation along the Trajectory and Neural Approximation ($\text{MPC}-\text{NPLT}_{\boldsymbol{u}(k|k-1)}-\text{NA}$). The first algorithm mimics the MPC algorithm with successive on-line model linearisation (the MPC algorithm with Nonlinear Prediction and Linearisation (MPC-NPL) [24,25]) whereas the second algorithm mimics the MPC strategy with the predicted output trajectory linearisation (the MPC algorithm with Nonlinear Prediction and Linearisation along the Trajectory (MPC-NPLT) [19]). The considered MPC algorithms are very computationally efficient because the control signal is calculated directly from an explicit control law, without any optimisation. The coefficients of the control law are determined on-line by a neural network (an approximator) which is trained off-line. Thanks to using neural approximation, successive on-line linearisation and calculations typical of the classical explicit MPC algorithms (i.e. a matrix decomposition and linear equations solving) are not necessary. Development of the described explicit MPC algorithms, in particular training and topology selection of the neural approximators, and their advantages (good control accuracy and computational efficiency) are demonstrated in the control system of a high-purity high-pressure ethylene-ethane distillation column. In particular, the algorithms are compared with the classical explicit MPC algorithms with on-line linearisation.

This paper is organised as follows. Firstly, Section 2 reminds the general idea of MPC and Section 3 describes the structure of the neural dynamic model of the process. In Section 4 two classical suboptimal explicit MPC algorithms with successive on-line linearisation which motivates development of the approximate algorithms are shortly discussed. Next, the main part of the paper given in Section 5 details two explicit MPC algorithms with neural approximation. Section 6 presents simulation results. Finally, Section 7 concludes the paper.

2. Model Predictive Control problem formulation

A good control algorithm should lead to accurate and fast control, i.e. the difference between the measured value of the output variable, y, and its desired set-point (the reference value), y^{ref}, should be minimised. One may also expect that changes of the calculated value of the manipulated variable, u, are not very big since excessive increments are likely to badly affect the actuator. A good control algorithm should also have the ability to take into account constraints of process variables, in particular constraints imposed on the value and on the rate of change of the manipulated variable.

In MPC algorithms [9,26,37,39,42] at each consecutive sampling instant k not the only the current value of the control signal (i.e. of the input signal of the process), but a set of future control increments is calculated

$$\triangle\boldsymbol{u}(k)=[\triangle u(k|k)\ \triangle u(k+1|k)\ldots\triangle u(k+N_{\text{u}}-1|k)]^{\text{T}} \tag{1}$$

where N_{u} is the control horizon and increments are defined as

$$\triangle u(k+p|k)=\begin{cases} u(k|k)-u(k-1) & \text{if } p=0\\ u(k+p|k)-u(k+p-1|k) & \text{if } p\geq 1\end{cases}$$

It is assumed that $\triangle u(k+p|k)=0$ for $p\geq N_{\text{u}}$. The objective of the MPC algorithm is to minimise differences between the reference trajectory, $y^{\text{ref}}(k+p|k)$, and predicted output values, $\hat{y}(k+p|k)$, over the prediction horizon N, $N\geq N_{\text{u}}$, i.e. for $p=1,\ldots,N$, and to penalise excessive control increments. Hence, the following quadratic cost function is usually used:

$$J(k)=\sum_{p=1}^{N}(y^{\text{ref}}(k+p|k)-\hat{y}(k+p|k))^2+\sum_{p=0}^{N_{\text{u}}-1}\lambda_p(\triangle u(k+p|k))^2 \tag{2}$$

where $\lambda_p>0$ are weighting coefficients (the bigger λ_p, the slower the algorithm). The problem of tuning MPC algorithms, i.e. adjusting parameters λ_p, N, N_{u}, is discussed elsewhere [26,42]. The cost function (2) is minimised on-line. As a result, future control increments (1) are calculated. Only the first element of the determined sequence is applied to the process, i.e. $u(k)=\triangle u(k|k)+u(k-1)$. At the next sampling instant, $k+1$, the prediction is shifted one step forward and the whole procedure is repeated.

As emphasised in Introduction, the possibility of taking into account constraints is a very important advantage of MPC algorithms. In this paper constraints imposed on input variables are considered. Optimal control increments (1) are repeatedly calculated on-line (at each sampling instant) from the following optimisation problem:

$$\begin{aligned}&\min_{\triangle u(k|k),\ldots,\triangle u(k+N_{\text{u}}-1|k)}\{J(k)\}\\ &\quad\text{subject to}\\ &\quad u^{\min}\leq u(k+p|k)\leq u^{\max},\quad p=0,\ldots,N_{\text{u}}-1\\ &\quad -\triangle u^{\max}\leq\triangle u(k+p|k)\leq\triangle u^{\max},\quad p=0,\ldots,N_{\text{u}}-1\end{aligned} \tag{3}$$

where $u^{\min}$, $u^{\max}$, $\triangle u^{\max}$ define constraints imposed on the magnitude of the input variable and on the increment of the input variable, respectively.

In MPC algorithms an explicit dynamic model is used in order to predict future behaviour of the process, i.e. to calculate predicted values of the output variable, $\hat{y}(k+p|k)$, over the prediction