Available online at www.sciencedirect.com
ScienceDirect
Computers and Chemical Engineering 31 (2007) 142–152
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Predictive control of a gas–liquid separation plant based on a Gaussian process model

Bojan Likar [a], Juš Kocijan [a,b,*]

[a] *Jozef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia*
[b] *University of Nova Gorica, Vipavska 13, SI-5000 Nova Gorica, Slovenia*

Received 26 April 2005; received in revised form 18 May 2006; accepted 21 May 2006
Available online 7 July 2006

Abstract

Gaussian process models provide a probabilistic non-parametric modelling approach for black-box identification of non-linear dynamic systems. The Gaussian processes can highlight areas of the input space where prediction quality is poor, due to the lack of data or to its complexity, by indicating the higher variance around the predicted mean. Gaussian process models contain noticeably less coefficients to be optimised. This paper demonstrates feasibility of application and realisation of a control algorithm based on a Gaussian process model. The extra information provided by the Gaussian process model is used in predictive control, where optimisation of the control signal takes the variance information into account. The feasibility of Gaussian process model usage for predictive control in industrial practice is demonstrated via the control of a gas–liquid separation plant.

Keywords: Model-based predictive control; Gaussian process model; Process control; Dynamic systems identification; Gas–liquid separation plant

1. Introduction

The popularity of MPC can be attributed largely to the ability of MPC algorithms to deal with constraints that are frequently met in control practice and are often not well addressed by other approaches. MPC algorithms can handle hard state and rate constraints on inputs and states that are usually, but not always, incorporated in the algorithms via an optimisation method. Linear model predictive control approaches (Maciejowski, 2002) started appearing in the early 1980s and are well-established in control practice (e.g. Qin & Badgwell, 1997 for an overview). Non-linear model predictive control (NMPC) approaches (Allgöwer, Badgwell, Qin, Rawlings, & Wright, 1999) started to appear about 10 years later and have also found their way into control practice (e.g. Qin & Badgwell, 2000; Young, Bartusiak, & Fontaine, 2001]) though their popularity cannot be compared to linear model predictive control. This is due to the difficulties associated with non-linear model construction and with the lack of the necessary confidence in the model. There have been a number of contributions in the field of non-linear model predictive control dealing with issues such as stability, efficient computation, optimisation, constraints and others. Some recent work in this field can be found in Allgöwer and Zheng (2000) and Kouvaritakis and Cannon (2001). NMPC algorithms are based on various non-linear models. Often these models are developed as first principles models, but other approaches – like black-box identification approaches – are also popular. Various predictive control algorithms are based on neural networks models (e.g. Nørgaard, Ravn, Poulsen, & Hansen, 2000), fuzzy models (e.g. Kavšek-Biasizzo, Škrjanc, & Matko, 1997) or local model networks (e.g. Johansen, Foss, & Sorensen, 1995). Non-linear model-based predictive control, as the name implies, critically depends on the non-linear plant model. The better the model, the better the control. This is where Gaussian process models can offer significant advantages. Gaussian process models provide a measure of confidence, which would be of help in NMPC design as noted in Tsai, Chu, Jang, and Shieh (2002), where a different approach to the same problem was described. The Gaussian process model is an example of a probabilistic non-parametric black-box model that also provides information about prediction uncertainties, which are difficult to evaluate appropriately in non-linear parametric mod-

* Corresponding author. Tel.: +386 1 4773 661; fax: +386 1 4773 994.
E-mail address: jus.kocijan@ijs.si (J. Kocijan).

0098-1354/$ – see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2006.05.011

els. The majority of work on Gaussian processes presented up to now considers the modelling of static non-linearities. The use of Gaussian processes in modelling dynamic systems is a recent development (e.g. Girard & Murray-Smith, 2005; Girard, Rasmussen, Candela, & Murray-Smith, 2003; Kocijan, Girard, Banko, & Murray-Smith, 2003; Kocijan & Likar, et al., 2003; Murray-Smith & Girard, 2001) and some control algorithms based on such an approach are described in Murray-Smith and Sbarbaro (2002) and Gregorčič and Lightbody (2003). This approach to modelling is not considered as a replacement of any existing method, but rather as a complementary approach to modelling. The drawback of Gaussian process models is the considerable computational burden. This burden may be perceived as an obstacle for Gaussian process model usage in industrial control applications. The purpose of this paper is to demonstrate the feasibility of application and realisation of a control algorithm based on a Gaussian process model on a process plant and to highlight some of the potentials. More about the benefits of dynamic systems modelling with Gaussian processes can be found in Girard and Murray-Smith (2005) and Kocijan and Girard, et al. (2003).

The paper is organized as follows. Dynamic Gaussian process models are briefly introduced in the following section. The control algorithm principle is described in Section 3. The example in Section 4 illustrates the operation of NMPC on a gas–liquid separator plant. Conclusions are stated at the end of the paper.

2. Modelling of dynamic systems with Gaussian processes

A Gaussian process is an example of the use of a flexible, probabilistic, non-parametric model with uncertainty predictions. Its use and properties for modelling are reviewed in Williams (1998).

A Gaussian process is a collection of random variables, which have a joint multivariate Gaussian distribution. Assuming a relationship of the form $y = f(\mathbf{x})$ between an input $\mathbf{x}$ and output y, we have $y^1, \ldots, y^n \sim \mathcal{N}(0, \sum)$, where $\Sigma_{pq} = \mathrm{Cov}(y_p, y_q) = C(\mathbf{x}_p, \mathbf{x}_q)$ determines the covariance between output points corresponding to input points $\mathbf{x}_p$ and $\mathbf{x}_q$. Thus, the mean $\mu(\mathbf{x})$ (usually assumed to be zero) and the covariance function $C(\mathbf{x}_p, \mathbf{x}_q)$ fully specify the Gaussian process. Note that the covariance function $C(.,.)$ can be any function having the property of generating a positive definite covariance matrix.

A common choice is:

$$C(\mathbf{x}_p, \mathbf{x}_q) = v_1 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d (x_p^d - x_q^d)\right] + v_0 \delta_{pq}, \tag{1}$$

where $\boldsymbol{\Theta} = [w_1 \ldots w_D v_0 v_1]^T$ are the 'hyperparameters' of the covariance functions, δ_{pq} is the Kronecker operator, and D is the input dimension. Other forms of covariance functions suitable for different applications can be found in Rasmussen (1996). For a given problem, the parameters are learned (identified) using the data at hand. After the learning the w parameters can be used as indicators of 'how important' the corresponding input components (dimensions) are: if w_d is zero or near zero it means that the inputs in dimension d contain little information and could possibly be removed.

Consider a set of N D-dimensional input vectors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ and a vector of output data $\mathbf{y} = [y^1, y^2, \ldots, y^N]^T$. Based on the data $(\mathbf{X}, \mathbf{y})$, and given a new input vector $\mathbf{x}^*$, we wish to find the predictive distribution of the corresponding output y^*. Unlike other models, there is no model parameter determination as such, within a fixed model structure. With this model, most of the effort consists of tuning the parameters of the covariance function. This is done by maximizing the log-likelihood of the parameters, which is computationally relatively demanding since the inverse of the data covariance matrix $(N \times N)$ has to be calculated at every iteration. Nevertheless, the number of parameters to be optimised is small $(D+2$, see Eq. (1)), which means that optimisation convergence might be faster and that the 'curse of dimensionality' so common to black-box identification methods is circumvented or at least decreased.

The described approach can be easily utilized for regression calculation. Based on a training set $\mathbf{X}$ a covariance matrix $\mathbf{K}$ of size $N \times N$ is determined. As already mentioned, the aim is to find the distribution of the corresponding output y^* at some new input vector $\mathbf{x}^* = [x_1(N+1), x_2(N+1), \ldots, x_D(N+1)]^T$.

For a new test input $\mathbf{x}^*$, the predictive distribution of the corresponding output is $y^* | (\mathbf{X}, \mathbf{y}), \mathbf{x}^*$ and is Gaussian, with mean and variance:

$$\mu(\mathbf{x}^*) = \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{y}, \tag{2}$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*) - \mathbf{k}(\mathbf{x}^*)^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}^*), \tag{3}$$

where $\mathbf{k}(\mathbf{x}^*) = [C(\mathbf{x}^1, \mathbf{x}^*), \ldots, C(\mathbf{x}^N, \mathbf{x}^*)]^T$ is the $N \times 1$ vector of covariances between the test and training cases, and $k(\mathbf{x}^*) = C(\mathbf{x}^*, \mathbf{x}^*)$ is the covariance between the test input and itself.

Gaussian processes can, like neural networks, be used to model static non-linearities and can therefore be used for modelling dynamic systems (Girard & Murray-Smith, 2005; Girard et al., 2003; Kocijan & Girard, et al., 2003; Kocijan & Likar, et al., 2003) if delayed input and output signals are fed back and used as regressors. In such cases an autoregressive model is considered, such that the current output depends on previous outputs, as well as on previous control inputs:

$$\begin{aligned} \mathbf{x}(k) &= [\hat{y}(k-1), \hat{y}(k-2), \ldots, \hat{y}(k-L), u(k-1), u(k-2), \ldots, u(k-L)]^T, \\ \hat{y}(k) &= f(\mathbf{x}(k)) + \epsilon, \end{aligned} \tag{4}$$

where k denotes the consecutive number of the data sample. Let $\mathbf{x}$ denote the state vector composed of the previous outputs y and inputs u up to a given lag L, and ϵ is white noise.