



Adaptive support vector regression for UAV flight control

Jongho Shin, H. Jin Kim^{*}, Youdan Kim

School of Mechanical & Aerospace Engineering, Seoul National University, Seoul, Republic of Korea

ARTICLE INFO

Article history:

Received 3 December 2008
Received in revised form 25 June 2010
Accepted 24 September 2010

Keywords:

Support vector regression
Feedback linearization
Unmanned aerial vehicle

ABSTRACT

This paper explores an application of support vector regression for adaptive control of an unmanned aerial vehicle (UAV). Unlike neural networks, support vector regression (SVR) generates global solutions, because SVR basically solves quadratic programming (QP) problems. With this advantage, the input–output feedback-linearized inverse dynamic model and the compensation term for the inversion error are identified off-line, which we call I-SVR (inversion SVR) and C-SVR (compensation SVR), respectively. In order to compensate for the inversion error and the unexpected uncertainty, an online adaptation algorithm for the C-SVR is proposed. Then, the stability of the overall error dynamics is analyzed by the uniformly ultimately bounded property in the nonlinear system theory. In order to validate the effectiveness of the proposed adaptive controller, numerical simulations are performed on the UAV model.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Many dynamic systems to be controlled are affected by various disturbances and uncertain factors. For example, unmanned aerial vehicles (UAVs), which have received a growing interest for various military and civilian applications (Giulietti, Pollini, & Innocenti, 2000; Ryan, Zennaro, Howell, Sengupta, & Hedrick, 2004), are subject to significant wind gusts, vortex effects and time delay from control signal to control servo. Although precise control of UAVs is a basic ingredient for many applications, it is not trivial to obtain an accurate UAV model because of these disturbances and measurement noise.

In order to design a controller against the uncertainties that cannot be predicted a priori, black-box identification using artificial neural networks (ANN) has been extensively studied. Sanner and Slotine (1992), a direct adaptive tracking control architecture using Gaussian radial function networks was designed to adaptively compensate for plant nonlinearities. Talebi, Khorasani, and Patel (1998) developed a position controller using four different neural network based schemes and revealed the validation of the proposed controller in the presence of unmodeled dynamics and nonlinearities. Recently, Giulietti et al. (2000) proposed a controller for a hybrid-electric UAV using neural networks to approximate a result of an energy optimization for a propulsion system and spent less energy than a two-stroke gasoline-powered UAV. The performance of ANN's has been validated in a wide range

of applications (Harmon, Frank, & Joshi, 2005; Kim & Calise, 1997; Polycarpou & Ioannou, 1992; Shin & Kim, 2006; Yabuta & Yamada, 1991) despite the issues of local minima during gradient descent, and selection of the ANN architecture.

On the other hand, kernel-based learning methods such as support vector machine (SVM) or support vector regression (SVR) transform the original problem into a quadratic programming (QP) problem whose global solution can be obtained by QP solvers (Schölkopf, Bartlett, Smola, & Williamson, 1998; Smola & Schölkopf, 1998). Thus, pattern classification (by SVM) and regression problems (by SVR) can be solved without the issues of the local minima. Another advantage of the SVM/SVR is that its structure is fixed, so the selection of the design parameters for the SVR is often straightforward. With such advantages, SVM has found various applications including road profile recognition for autonomous navigation (Holzapfel, Sofsky, & Neuschaefer-Rube, 2003) and target recognition from synthetic aperture radar imaging (Zhao & Principe, 2001).

Compared with the popularity of SVM in many classification and recognition problems, application of SVR in control systems is still in an early stage. In Wang, Pi, and Sun (2007), an online algorithm for an SVR inverse model has been studied. The online algorithm in Wang et al. (2007) is a training method in the context of incremental or decremental learning (Cauwenberghs & Poggio, 2001), rather than an adaptation algorithm performed in the control loop. In Iplicki (2006) and Xi, Poo, and Chou (2007), SVR-based techniques were used for obtaining a plant model for model predictive control (MPC). An off-line trained SVR plant model is fed to the optimization routine and used for the prediction of the future states over the lookahead horizon. And Suykens, Vandewalle, and Moor (2001) proposed a least-squares SVM-based

^{*} Corresponding author. Tel.: +82 2 880 9252; fax: +82 2 887 2662.

E-mail addresses: jh0524@snu.ac.kr (J. Shin), hjinkim@snu.ac.kr (H. Jin Kim), ydkim@snu.ac.kr (Y. Kim).

optimal controller and validated its performance. However, the above studies did not consider the nonlinearity or uncertainties of the plant. Therefore, if the system we want to control changes significantly, overall performance can degrade.

Unlike the previous SVR-based control research, this study uses ideas from the input–output feedback linearization in nonlinear control and then, uses the global property of the solution of the SVR. Two SVR machines are trained offline using input–output data from the input–output feedback-linearized system. The first one called the inversion SVR (I-SVR) is designed for training the feedback-linearized inverse dynamic model and the second one called the compensation SVR (C-SVR) is constructed to estimate the output derivative. However, even though the solution of the SVR has the global property in the sense of being offline, there, in practice, exists uncertainty or unknown disturbances that may be unrepresented in the training data set. In order to handle the unexpected nonlinearities or uncertainties, an online adaptation rule for the C-SVR is designed in an adaptive control framework.

This paper is organized as follows. The SVR algorithm is reviewed briefly in Section 2. In Section 3, an approach for combining input–output feedback linearization and C-SVR, and the online adaptation rule for the SVR are addressed. The overall stability under the adaptation rule is analyzed using the ultimately uniformly bounded property in the nonlinear system theory. Section 4 describes an UAV system whose flight test data is used in this study, and presents the results of the UAV flight control using the proposed approach. A conclusion is given in Section 5.

2. ϵ -support vector regression

This section briefly reviews the ϵ -SVR algorithm (Schölkopf et al., 1998; Schölkopf, Burges, & Smola, 1999; Smola & Schölkopf, 1998). Consider the training dataset $D = \{X_k, Y_k\}_{k=1}^N$ where X_k is the k th input data in the input space $\mathcal{X} \subseteq \mathbb{R}^n$ and Y_k is the corresponding output value in the output space $\mathcal{Y} \subseteq \mathbb{R}$.

ϵ -SVR model is trained by the following relationship between the input and output data points (Vapnik, 1995, 1998):

$$F(X_k) = \langle \mathbf{w}, \Phi(X_k) \rangle + c \quad (1)$$

where \mathbf{w} is a vector in the feature space $\mathcal{F} \subseteq \mathbb{R}^n$, $\Phi(X_k)$ is a mapping from the input space \mathcal{X} to the feature space \mathcal{F} , c is the bias term and $\langle \cdot, \cdot \rangle$ stands for the inner product in \mathcal{F} .

ϵ -SVR model is based on Vapnik's ϵ -insensitive loss function for function approximation (Vapnik, 1995). It is aimed at minimizing the empirical risk

$$\frac{1}{N} \sum_{i=1}^N |Y_i - F(X_i)|_\epsilon$$

with the following ϵ -insensitive model:

$$|Y_i - F(X_i)|_\epsilon = \begin{cases} 0 & \text{if } |Y_i - F(X_i)| \leq \epsilon \\ |Y_i - F(X_i)| - \epsilon & \text{else.} \end{cases} \quad (2)$$

The optimization problem can be formulated in the primal space, as the following (Smola & Schölkopf, 1998; Vapnik, 1995)

$$\min_{\mathbf{w}, c, \xi, \xi^*} P_\epsilon = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (3)$$

subject to the constraints

$$\begin{cases} Y_i - \langle \mathbf{w}, \Phi(X_i) \rangle - c \leq \epsilon + \xi_i \\ \langle \mathbf{w}, \Phi(X_i) \rangle + c - Y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, 2, \dots, N \end{cases} \quad (4)$$

where ϵ is the maximum value of tolerable error, ξ_i 's and ξ_i^* 's are slack variables, $\|\cdot\|$ is the Euclidean norm, and C is a

regularization parameter that represents a trade-off between the model complexity and the tolerance to the error larger than ϵ .

The dual form of (3) becomes a quadratic programming(QP) problem as follows (Smola & Schölkopf, 1998; Vapnik, 1995):

$$\begin{aligned} \min_{\eta, \eta^*} D_\epsilon = & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \kappa(X_i, X_j) (\eta_i - \eta_i^*) (\eta_j - \eta_j^*) \\ & + \epsilon \sum_{i=1}^N (\eta_i + \eta_i^*) - \sum_{i=1}^N Y_i (\eta_i - \eta_i^*) \end{aligned} \quad (5)$$

subject to the constraints

$$0 \leq \eta_i, \eta_i^* \leq C, \quad \sum_{i=1}^N (\eta_i - \eta_i^*) = 0, \quad i = 1, \dots, N \quad (6)$$

where $\kappa(X_i, X_j)$ is a kernel function given by $\kappa(X_i, X_j) = \Phi(X_i)^T \Phi(X_j) = \kappa_{ij}$. Motivated by Mercer's condition, the kernel function handles the inner product in the feature space and hence the explicit form of $\Phi(X_k)$ does not need to be known (Vapnik, 1995). In this study, the following Gaussian radial basis kernel function is used

$$\kappa(X_i, X_j) = \exp\left(-\frac{(X_i - X_j)^T (X_i - X_j)}{\sigma^2}\right). \quad (7)$$

The solution of the QP problem (5) subject to (6) is the optimum values of η_i 's and η_i^* 's. The value of c in the model can be determined by the condition that at the point of the solution the product between dual variables and constraints has to vanish (Smola & Schölkopf, 1998). Then one obtains

$$\mathbf{w} = \sum_{i=1}^N (\eta_i - \eta_i^*) \Phi(X_i),$$

and the data points corresponding to non-zero values of $(\eta_i - \eta_i^*)$ are called support vectors. When only the support vectors are considered, the model becomes

$$F(X_k) = \sum_{i=1, (i \in SV)}^{N_{SV}} \zeta_i \kappa(X_k, X_i) + c \quad (8)$$

where $\zeta_i = \eta_i - \eta_i^*$, N_{SV} denotes the number of support vectors in the model. The obtained SVM model is sparse in the sense that the whole training data are represented by the support vectors only and many of ζ_i are zero.

The design parameters of ϵ -SVR are the maximum tolerable error ϵ at the output, the regularization parameter C , the number of training patterns N and the parameter σ of the kernel function. Although there is no systematic way of determining optimal values of these parameters, some effective guidelines can be found in Cherkassky and Ma (2004).

3. Nonlinear control using feedback linearization and support vector regression

This section reviews the concept of input–output feedback linearization in the nonlinear system theory briefly and explains how to apply support vector regression (SVR) to the feedback-linearized system.

3.1. Input–output feedback linearization

Consider the following nonlinear dynamic system

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات