

Receding horizon iterative dynamic programming with discrete time models[☆]

A. Rusnák^a, M. Fikar^{a,*}, M.A. Latifi^{b,1}, A. Mészáros^a

^a Department of Process Control, CHTF STU, Radlinského 9, SK-812 37 Bratislava, Slovakia

^b Laboratoire des Sciences du Génie Chimique, CNRS-ENSIC, B.P. 451, 1 rue Grandville, 54001 Nancy Cedex, France

Received 5 November 1999; accepted 13 September 2000

Abstract

This contribution proposes a modified version of the Iterative Dynamic Programming (IDP) method. Two main differences to the original method are introduced. The new algorithm deals with discrete-time input–output models compared to continuous-time state–space models described by a set of ODE/DAE used in the original method. The main purpose of these modifications is to reduce computational load of the original method, estimate the process models more easily, and to enable its use on-line in receding horizon predictive control framework. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Iterative dynamic programming; Discrete-time models; Optimal control

1. Introduction

Iterative dynamic programming (IDP) is a method of dynamic optimisation developed by Luus a decade ago (Luus, 1989; Luus & Rosen, 1991). It has attracted the attention of many researches due to its many very favourable properties: is easy to implement, is quite robust, does not involve solution of a non-linear programming (NLP) even in the case of input constraints, is reported to be capable of finding a global optimum, and does not require any differentiation of process equations that is sometimes very difficult.

The main drawback of the method is the problem of dimensionality that results in enormous computational load. This is the reason why it is used mainly for determination of open-loop optimal control policies (Dadebo & McAuley, 1995; Mekarapiruk & Luus, 1997; Fikar, Latifi, Fournier & Creff, 1998) and so it is confined only to theoretical studies.

There are some similar methods that search through the control space in a manner similar to IDP and that are significantly faster (Carrasco & Banga, 1997). This is usually achieved with the compromise of obtaining cost values that are slightly worse than optimal. However, it is well known that the optimum usually lies in a very flat valley and that suboptimal control trajectories may be very different from the optimal one.

As only open-loop trajectories are determined, the method in its original form is unsuitable for on-line implementation. The presence of modeling errors and disturbances can quickly lead to changes in optimal control trajectory resulting in off-spec products. A solution is to apply the principles known from predictive control, namely receding horizon implementation and disturbance estimation.

In this article, we propose a modified IDP method that is suitable for on-line implementation. Unlike the original method, discrete-time input–output models are used. Discrete-time models are in general faster to simulate, which is the main obstacle in IDP. Also as IDP uses principles of control vector parametrisation and usually assumes piece-wise constant control trajectory, discretisation of states/outputs follows quite naturally. Input–output model descriptions are used to avoid problems with state estimation and observer de-

[☆] A preliminary version of this paper was presented at ESCAPE 9, Budapest, 1999.

* Corresponding author. Tel.: +421-7-59325354; fax: +421-7-396469.

E-mail addresses: fikar@cvt.stuba.sk (M. Fikar), latifi@ensic.u-nancy.fr (M.A. Latifi).

¹ Tel.: +33-3-83175234; fax: +33-3-83175326.

sign. Moreover, parameters of input–output models may be more easily identifiable on-line and thus may improve the method in adaptive fashion.

As process predictor, several discrete-time models may be used, ranging from simple linear to complex nonlinear models such as for example artificial neural networks (ANN). The use of ANN in combination with IDP has also been investigated by Tholudur and Ramirez (1995) where the network has been used to identify parameters of known non-linear characteristics of a process and subsequently to estimate continuous states of the process. The disadvantage of such an approach becomes noticeable with not all states measurable. Moreover, the original drawback of IDP, speed, remains.

The article is organised as follows. The next section gives the main results of this article — a modified IDP algorithm together with specification of models needed, and discussion about predictive framework implementation. The simulations in Section 3 deal with the simple multivariable model of a distillation column and with a biochemical reactor. In both cases, a different type of discrete-time model is used. Finally, discussion of the results and conclusions are presented in Section 4.

2. Iterative dynamic programming

The original IDP method has been developed for systems described by a set of differential equations, hence naturally in state–space formulation. Its detailed theoretical analysis can be found for example in Bojkov and Luus (1993), Bojkov and Luus (1995) and Dadebo and Mcauley (1995).

As its name suggests, IDP is based on Bellman's principle of optimality. Its basic principle is to optimise P single control stages in turn starting at the last stage instead of optimising all P stages simultaneously. Thus, whenever the final i stages of the optimal control have been established, the preceding stage may be obtained by simply considering one new stage and then continuing with the already established control policy in the remaining stages.

The modifications proposed in this contribution are due to use of discrete-time input/output models with the advantages reasoned in the introduction.

2.1. Process modeling and prediction

A continuous-time process can generally be described by a set of possibly non-linear ODE/DAE of the form

$$f_1(\dot{x}(t), x(t), y(t), u(t)) = 0 \quad (1)$$

The difficulties of the original IDP stem from the use of these models as their simulation is often time-con-

suming. Let us therefore assume that the following process model is approximated from Eq. 1

$$y(t) = f(y(t-1), y(t-2), \dots, u(t-1), u(t-2), \dots) \quad (2)$$

This model will be used in IDP as an output predictor for some prediction horizon. As this model only approximates the real controlled process, an offset can occur due to disturbances and process mismatch. Therefore, the prediction of the process output j steps forward has to be corrected with regard to information available at time t

$$\hat{y}(t+i) = y_{\text{MOD}}(t+i) + d(t) \quad (3)$$

where d is a disturbance that is assumed to be constant. It is estimated from data available at time t as

$$d(t) = y(t) - y_{\text{MOD}}(t) \quad (4)$$

where $y(t)$ is the measured process output and $y_{\text{MOD}}(t)$ the model output calculated from Eq. 2.

At each sampling period, the following signals are fed into the discrete-time predictor (Eq. 2): past and present plant outputs together with past and proposed future inputs. The predictor calculates predictions of the plant outputs over the relevant horizon, which are corrected with the calculated deviation (Eq. 3) at time t . These corrected model predictions are used in the IDP algorithm.

2.2. Problem formulation

Consider the discrete-time system (Eq. 2) and suppose that the input $u(t+j)$ has to be within limits:

$$u_k^{\min} \leq u_k(t+j) \leq u_k^{\max}, \quad k = 1, 2, \dots, m \quad (5)$$

The associated performance index to be optimised is:

$$J = F(\hat{y}(t+1), \dots, \hat{y}(t+P), u(t), u(t+1) \\ \times \dots, u(t+P-1)) \quad (6)$$

where \hat{y} is a prediction of the process output y . The optimal control problem is to find the piece-wise constant control policy $u(t+j)$, $j=0, \dots, P-1$ such that the performance index given by Eq. 6 is minimal.

2.3. Modified IDP algorithm

At first, the following parameters have to be defined: P , number of stages (or equivalently prediction horizon); M , number of generated control actions; N , number of y -grid points; r^1 , initial size of control region; φ , contraction factor, $\varphi \in [0.7 - 0.9]$; N_i , number of iterations; n , number of steps for which the output trajectories are compared.

Next, an initial control trajectory is chosen and the iteration counter is initialised as $i=1$. The proposed algorithm can be described by the following steps:

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