



Identification and adaptation of linear look-up table parameters using an efficient recursive least-squares technique

James C. Peyton Jones^{*}, Kenneth R. Muske¹

Center for Nonlinear Dynamics and Control, College of Engineering, Villanova University, 800 Lancaster Ave., Villanova, PA 19085, United States

ARTICLE INFO

Article history:

Received 29 January 2009

Received in revised form

9 April 2009

Accepted 13 April 2009

Available online 21 May 2009

Keywords:

Lookup table

Piece-wise linear models

Parameter estimation

Least-squares estimation

ABSTRACT

Linear look-up tables are widely used to approximate and characterize complex nonlinear functional relationships between system input and output. However, both the initial calibration and subsequent real-time adaptation of these tables can be time consuming and prone to error as a result of the large number of table parameters typically required to map the system and the uncertainties and noise in the experimental data on which the calibration is based. In this paper, a new method is presented for identifying or adapting the look-up table parameters using a recursive least-squares based approach. The new method differs from standard recursive least squares algorithms because it exploits the structure of the look-up table equations in order to perform the identification process in a way that is highly computationally and memory efficient. The technique can therefore be implemented within the constraints of typical embedded applications. In the present study, the technique is applied to the identification of the volumetric efficiency look-up table commonly used in gasoline engine fueling strategies. The technique is demonstrated on a Ford 2.0L I4 Duratec engine using time-delayed feedback from a sensor in the exhaust manifold in order to adapt the table parameters online.

© 2009 ISA. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Model-based representations are concise and able to characterize a wide range of operation, including dynamic behavior, with a small set of parameters. Look-up tables or maps, on the other hand, are useful for characterizing systems where the functional relationship is not known or simply too complex to represent analytically. Linear look-up tables are widely used in function approximation applications ranging from machine learning, floating point numerical computation, nonlinear network analysis, and nonlinear adaptive control systems. Details concerning the theoretical development, construction, and implementation of these tables can be found in the text by Leenaerts and van Bokhoven [1]. In this paper, the primary interest is the approximation of nonlinear system response, as in the early work of Bellman and Roth [2], with particular emphasis on real-time control and monitoring applications. In these applications, the use of look-up tables enables the nonlinear response of the system, within some operating region, to be characterized almost arbitrarily by adjusting the relevant table parameter values. Generally only static relationships are represented in this way, although it is possible to represent system dynamics by

mapping the rate of change of a given signal of interest onto one of the axes of the table.

Linear look-up tables are therefore highly flexible, but this flexibility is achieved at the expense of a large number of parameters required to define the grid and the commensurately high effort required to identify their values. For the characterization of multivariable functions, each additional independent variable adds a new axis or dimension to the table and the number of table parameters increases geometrically. For example, a 2-dimensional table with five points on each axis contains only 25 elements but the number of elements increases to 3125 elements for a 5-dimensional table. Because of this geometric increase, linear look-up tables and maps are generally confined to 1 or 2-dimensional problems. Even in these cases, however, the issue of table parameter value identification remains. Not only do appropriate values have to be determined in order to populate the table initially, but in many cases there is also a need to adapt or calibrate the table parameters on-line in order to account for slowly time-varying behaviors of the system.

In this paper, the look-up table input-output relationship is first cast in the form of an over-determined set of linear equations. A new recursive least squares estimation algorithm is then presented in order to identify the table parameters while minimizing the computational and data storage requirements relative to standard recursive least squares methods. These requirements can be particularly critical in real-time embedded applications where the computational resources are limited. The new method takes

^{*} Corresponding author. Tel.: +1 610 519 4216.

E-mail addresses: james.peyton-jones@villanova.edu (J.C. Peyton Jones), kenneth.muske@villanova.edu (K.R. Muske).

¹ Tel.: +1 610 519 6195.

advantage of the particular sparse structure of the recursive least squares matrix equations for 1-dimensional and 2-dimensional look-up tables to obtain a very efficient numeric algorithm for computing the recursive least squares estimate. In order to apply the techniques presented in this work, it is assumed that the look-up table structure (dimension and grid point placement) has been previously established. The determination of the interpolation grid point locations can be carried out using techniques discussed in [1,3] and is not considered within the scope of this work.

This paper is structured as follows. Section 2 presents the mathematical relationships for the linear look-up table parameter identification problem in one and two dimensions respectively. Least-squares identification is discussed in Section 3 along with an analysis of the computational and storage requirements necessary for these algorithms. An alternate recursive least squares approach that takes advantage of the sparse structure of the matrix equations representing the linear look-up table parameter identification problem is presented in Section 4 where a simple 1-dimensional look-up table example is used to illustrate the application of the proposed technique. On-line calibration of the 2-dimensional volumetric efficiency look-up table for an experimental Ford 2.0L I4 Duratec engine is then presented in Section 5 where time-delayed feedback from a wide-ranging exhaust gas oxygen sensor in the exhaust manifold is used to update the table parameters.

2. Linear look-up table mathematical description

Consider the 1-dimensional look-up table illustrated in Fig. 1. The input space of the table is defined by the set of ordered (but not necessarily equi-spaced) points x_i , $i \in [1 \dots n_x]$. The output values at each of these points are defined by the corresponding table parameters, θ_i . Intermediate output values are determined using linear interpolation. The index i of the desired table cell is identified such that the input x falls within the i th table cell

$$i = \max(j) \mid (x_j \leq x) \Rightarrow x_i \leq x < x_{i+1} \quad (1)$$

and the normalized offset of x within the cell is then determined according to,

$$r = \frac{x - x_i}{x_{i+1} - x_i} \quad (2)$$

The interpolated output is given by the relation,

$$y = (1 - r)\theta_i + r\theta_{i+1} \quad (3)$$

which can also be written as the vector–vector dot product shown in Eq. (4).

$$y = ((1 - r) \quad r) \begin{pmatrix} \theta_i \\ \theta_{i+1} \end{pmatrix} \quad (4)$$

If only a single observation of the output is available, then it is not possible to identify the two unknowns (θ_i and θ_{i+1}) using Eq. (4). If, however, there are many observations $y(k)$ of the response to excitation within a single cell i , then Eq. (4) can be written in matrix-vector form

$$\mathbf{y} = \Phi \boldsymbol{\theta} \quad (5)$$

where $\mathbf{y}$ is the vector of output values, Φ is a matrix of input-dependent interpolation factors, and $\boldsymbol{\theta}$ is a 2-element vector of the table parameters associated with the cell of interest as shown in Eq. (6).

$$\mathbf{y} = \begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(n) \end{pmatrix} \quad \Phi = \begin{pmatrix} (1 - r(1)) & r(1) \\ (1 - r(2)) & r(2) \\ \vdots & \vdots \\ (1 - r(n)) & r(n) \end{pmatrix} \quad (6)$$

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_i \\ \theta_{i+1} \end{pmatrix}$$

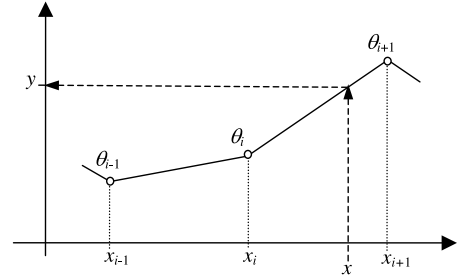


Fig. 1. 1-dimensional linear interpolation.

Eq. (5) is in a form whose solution is readily obtained, but it is unlikely in practice that the data will be constrained to a single table cell. A general analysis to account for data values falling within several cells is presented in the following section.

2.1. General structure of 1-dimensional tables

To generalize the analysis as the input varies over multiple cells, the cell index i is redefined to be function of the time step k

$$i(k) = \max(j) \mid (x_j \leq x(k)) \Rightarrow x_{i(k)} \leq x(k) < x_{i(k)+1} \quad (7)$$

This re-definition propagates to the offset equation

$$r(k) = \frac{x(k) - x_{i(k)}}{x_{i(k)+1} - x_{i(k)}} \quad (8)$$

and to the interpolation equation.

$$y(k) = ((1 - r(k)) \quad r(k)) \begin{pmatrix} \theta_{i(k)} \\ \theta_{i(k)+1} \end{pmatrix} \quad (9)$$

Although Eq. (9) is similar in form to Eq. (4), it is no longer possible to re-write it according to the form presented in Eqs. (5) and (6) because the two parameters $\theta_{i(k)}$, $\theta_{i(k)+1}$ no longer refer exclusively to a single cell and may have different identities at different time steps. The vector $\boldsymbol{\theta}$ is therefore re-defined to contain all n_x table parameter values even though only two of these will be ‘active’ at any given time step.

$$\boldsymbol{\theta} = (\theta_1 \quad \theta_2 \quad \dots \quad \theta_{n_x})^T \quad (10)$$

To select the appropriate pair of parameters, the matrix Φ (which now has n_x columns) is defined with only two non-zero entries in any row $\varphi^T(k)$. These entries define the required interpolation factors at the given time step k as follows.

$$\varphi^T(k)_j = \begin{cases} (1 - r(k)) & j = i(k) \\ r(k) & j = i(k) + 1 \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

For example, if the input $x(k)$ is in cells $i(k) = \{2, 1, 3\}$ for the first three time-steps and the table itself contained $n_x = 5$ unknown parameters, then the matrix Φ becomes

$$\Phi = \begin{pmatrix} 0 & (1 - r(1)) & r(1) & 0 & 0 \\ (1 - r(2)) & r(2) & 0 & 0 & 0 \\ 0 & 0 & (1 - r(3)) & r(3) & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \quad (12)$$

With these new definitions, Eq. (5) can be used to describe the look-up table response to an input whose values $x(k)$ can vary across the entire input space of the table.

2.2. General structure of 2-dimensional tables

The procedure used to compute the output of a 1-dimensional lookup table in the previous section can be extended to compute the output of a 2-dimensional table. In this case, the input space

Download English Version:

<https://daneshyari.com/en/article/5005287>

Download Persian Version:

<https://daneshyari.com/article/5005287>

[Daneshyari.com](https://daneshyari.com)