PERGAMON

Engineering Applications of Artificial Intelligence 13 (2000) 147–157

Engineering Applications of
ARTIFICIAL
INTELLIGENCE

www.elsevier.com/locate/engappai

The exploitation of neural networks in automotive engine management systems

P.J. Shayler[a,*], M. Goodman[a], T. Ma[b]

[a]University of Nottingham, Nottingham, NG7 2RD, UK
[b]Ford Motor Company Research and Engineering Centre, Essex, SS15 6EE, UK

Received 1 October 1998; accepted 1 October 1999

Abstract

The use of electronic engine control systems on spark ignition engines has enabled a high degree of performance optimisation to be achieved. The range of functions performed by these systems, and the level of performance demanded, is rising and thus so are development times and costs. Neural networks have attracted attention as having the potential to simplify software development and improve the performance of this software. The scope and nature of possible applications is described. In particular, the pattern recognition and classification abilities of networks are applied to crankshaft speed fluctuation data for engine-fault diagnosis, and multidimensional mapping capabilities are investigated as an alternative to large 'lookup' tables and calibration functions. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Automotive engines; Calibration; Engine control; Fault diagnosis; Neural networks; Pattern recognition

1. Introduction

Many of the recent advances in spark ignition engine performance stem from, or have been made possible by, the introduction of electronic engine control (EEC) units and associated developments in automotive electronics. Early applications included the control of electronic fuel injection and spark timing. Mechanical controls have been displaced, and indeed, future legislative requirements for on-board-diagnostics and restrictions on the emission of pollutants cannot be met without electronic controls. The range of applications served by microprocessors in EEC units continues to expand, and with this, the complexity of software development and maintenance increases. This is straining conventional methods of development, and there is much interest in alternatives. The use of neural-network computing techniques is attracting interest as one approach that may have advantages in some applications. The aim of the study reported here was to review how neural networks are being exploited, and to investigate three applications that take advantage of a range of attributes associated with these. In each application the neural networks used have a multi-layer perception (MLP) architecture and have been trained using the back-propagation algorithm. The first application utilises the pattern-classification abilities of a neural network in a service bay fault-diagnostic system. The second and third investigate the possibility of replacing specific sections of the engine control strategy software with neural-network-based systems. The perceived advantages include generic applications of these systems to a wide range of engines, requiring only minor calibration changes for each engine modification. Accordingly, the development and maintenance of the calibration and control strategy software for each new engine variant is simpler.

Early examples of neural networks being exploited in automotive applications have been described by

* Corresponding author. Tel.: +115-951-3780; fax: +115-951-3800.
E-mail address: paul.shayler@nottingham.ac.uk (P.J. Shayler).

0952-1976/00/$ - see front matter
PII: S0952-1976(99)00048-2

Marko (1991). He developed neural-network-based systems for the real-time monitoring of vehicle control systems for faults, and the control of a non-linear active suspension system. In the second case, a back-propagation network with only two hidden nodes was trained to provide satisfactory performance using a small amount of data acquired from a vehicle travelling over 'less than 100 feet of bumpy road'.

Lenz and Schroeder (1996) investigated the use of the General Regression Neural Network (GRNN), described by Specht (1991), to learn certain dependencies that characterise the supply of air to an S.I. engine. These networks perform multidimensional regression between the training examples. The GRNN has two main advantages: the output is always bounded by the training examples, and it may be trained using a one-pass algorithm, greatly reducing the training time. Also it does not suffer from the difficulty often associated with the back-propagation algorithm, namely the convergence to a local rather than the global minimum of the error surface. The major disadvantage is that a hidden unit is required for each input pattern (unless the patterns are grouped in clusters), which can lead to prohibitively large networks if large training sets are used.

Multi-layer perceptrons offer a simple structure, and may be trained by a simple algorithm in the form of back-propagation. The range of operation is limited, however, and applications are generally restricted to static mapping tasks. Beaumont and Frith (1994) proposed a solution to this problem for transient air/fuel ratio control of an S.I. engine. By applying delayed and filtered values of relevant measurements to its inputs, their network was provided with some knowledge of the system state or previous time history.

An alternative approach to the control of dynamic plants was presented by Feldkamp et al. (1992). This addressed the problem of requiring knowledge of the recent history of inputs, outputs or internal states by using so-called 'recurrent networks'. These are based on multi-layer perceptrons, but include internal feedback to units of the same or previous layers, usually with a unit time delay in the feedback loop. In this way, the network output is a function of both the current and recent inputs. The node activations provide a recent temporal history of inputs, introducing a degree of context sensitivity to the output. Two alternative schemes are used for network training: dynamic back-propagation and a technique based on the Kalman filter method described by Puskorius and Feldkamp (1991). The latter basically regards network training as a parameter-estimation problem, and offers superior performance and more rapid training than standard back-propagation. The two training methods were specifically adapted to take account of the internal state of the feedback loops. In particular, they note that the training patterns must be presented in order, and not at random as is usually the case for conventional static mapping MLPs.

Feldkamp et al. (1992) also present an optimised scheme for the development of neural controllers comprised of an identification network and a controller network. The identification network is initially treated as a static mapping tool, and is trained using data derived from the plant's response to a random non-repeating input. Its purpose is to predict the state vector of the plant at the next time step, based on the current state and control input. The identification network is used to calculate the gradients required for the controller training, in particular the derivative of the predicted state with respect to the trainable parameters of the controller. The controller, however, computes the control signal on the basis of the actual state, not that predicted by a model. This technique has been used with a high degree of success for many automotive-related control applications, including anti-lock braking systems (Davis et al., 1992), active vehicle suspensions (Puskorius and Feldkamp, 1992) and S.I. engine idle speed control (Puskorius and Feldkamp, 1993).

The first application presented here, in common with most documented MLP approaches, uses binary network outputs, and is concerned with pattern recognition and classification. The latter two make use of analogue outputs from the networks.

2. Engine fault diagnostics

2.1. Aims

This application uses the network as a pattern classifier to assist with service-bay engine-fault diagnosis. The overall aim was to produce a system that could distinguish between intermittent and persistent ignition and fuelling faults, and indeed also identify a 'normal' engine, operating correctly. Ideally, such a system should use the sensors found as standard on a production engine. The signals provided to the engine management system could then be intercepted and used by the diagnostic equipment with the minimum effort.

2.2. Data definition and experimental methodology

The data used for diagnosis was based on the crankshaft speed fluctuations, over 50–100 consecutive combustion events, while the engine was idling. Several authors, including Rizzoni (1987), Mihele and Citron (1984) and Ina et al. (1986) have shown that instantaneous crankshaft speed measurements contain features related to both cylinder-specific and overall