N

Neural Control System for Autonomous Vehicles

Francisco García-Córdova
Polytechnic University of Cartagena (UPCT), Spain

Antonio Guerrero-González
Polytechnic University of Cartagena (UPCT), Spain

Fulgencio Marín-García
Polytechnic University of Cartagena (UPCT), Spain

INTRODUCTION

Neural networks have been used in a number of robotic applications (Das & Kar, 2006; Fierro & Lewis, 1998), including both manipulators and mobile robots. A typical approach is to use neural networks for nonlinear system modelling, including for instance the learning of forward and inverse models of a plant, noise cancellation, and other forms of nonlinear control (Fierro & Lewis, 1998).

An alternative approach is to solve a particular problem by designing a specialized neural network architecture and/or learning rule (Sutton & Barto, 1981). It is clear that biological brains, though exhibiting a certain degree of homogeneity, rely on many specialized circuits designed to solve particular problems.

We are interested in understanding how animals are able to solve complex problems such as learning to navigate in an unknown environment, with the aim of applying what is learned of biology to the control of robots (Chang & Gaudiano, 1998; Martínez-Marín, 2007; Montes-González, Santos-Reyes & Ríos-Figueroa, 2006).

In particular, this article presents a neural architecture that makes possible the integration of a kinematical adaptive neuro-controller for trajectory tracking and an obstacle avoidance adaptive neuro-controller for nonholonomic mobile robots. The kinematical adaptive neuro-controller is a real-time, unsupervised neural network that learns to control a nonholonomic mobile robot in a nonstationary environment, which is termed Self-Organization Direction Mapping Network (SODMN), and combines associative learning and Vector Associative Map (VAM) learning to generate transformations between spatial and velocity coordinates (García-Córdova, Guerrero-González & García-Marín, 2007). The transformations are learned in an unsupervised training phase, during which the robot moves as a result of randomly selected wheel velocities. The obstacle avoidance adaptive neuro-controller is a neural network that learns to control avoidance behaviours in a mobile robot based on a form of animal learning known as operant conditioning. Learning, which requires no supervision, takes place as the robot moves around a cluttered environment with obstacles. The neural network requires no knowledge of the geometry of the robot or of the quality, number, or configuration of the robot's sensors. The efficacy of the proposed neural architecture is tested experimentally by a differentially driven mobile robot.

BACKGROUND

Several heuristic approaches based on neural networks (NNs) have been proposed for identification and adaptive control of nonlinear dynamic systems (Fierro & Lewis, 1998; Pardo-Ayala & Angulo-Bahón, 2007).

In wheeled mobile robots (WMR), the trajectory-tracking problem with exponential convergence has been solved theoretically using time-varying state feedback based on the backstepping technique in (Ping & Nijmeijer, 1997; Das & Kar, 2006). Dynamic feedback linearization has been used for trajectory tracking and posture stabilization of mobile robot systems in chained form (Oriolo, Luca & Vendittelli, 2002).

The study of autonomous behaviour has become an active research area in the field of robotics. Even the simplest organisms are capable of behavioural feats unimaginable for the most sophisticated machines. When an animal has to operate in an unknown environment it must somehow learn to predict the consequences of its own actions. Biological organisms are a clear example that this short of learning is possible in spite of what, from an engineering standpoint, seem to be insurmountable difficulties: noisy sensors, unknown kinematics and dynamics, nostationary statistics, and so on. A related form of learning is known as operant conditioning (Grossberg, 1971). Chang and Gaudiano (1998) introduce a neural network for obstacle avoidance that is based on a model of classical and operant conditioning.

Psychologists have identified classical and operant conditioning as two primary forms of learning that enables animals to acquire the causal structure of their environment. In the classical conditioning paradigm, learning occurs by repeated association of a Conditioned Stimulus (CS), which normally has no particular significance for an animal, with an Unconditioned Stimulus (UCS), which has significance for an animal and always gives rise to an Unconditioned Response (UCR). The response that comes to be elicited by the CS after classical conditioning is known as the Conditioned Response (CR) (Grossberg & Levine, 1987). Hence, classical conditioning is the putative learning process that enables animals to recognize informative stimuli in the environment.

In the case of operant conditioning, an animal learns the consequences of its actions. More specifically, the animal learns to exhibit more frequently a behaviour that has led to reward in the past, and to exhibit less frequently a behaviour that led to punishment.

In the field of neural networks research, it is often suggested that neural networks based on associative learning laws can model the mechanisms of classical conditioning, while neural networks based on reinforcement learning laws can model the mechanisms of operant conditioning (Chang & Gaudiano, 1998).

The reinforcement learning is used to acquire navigation skills for autonomous vehicles, and updates both the vehicle model and optimal behaviour at the same time (Galindo, González & Fernández-Madrigal, 2006; Lamiraux & Laumond, 2001; Galindo, Fernández-Madrigal & González, 2007).

In this article, we propose a neurobiologically inspired neural architecture to show how an organism, in this case a robot, can learn without supervision to recognize simple stimuli in its environment and to associate them with different actions.

ARCHITECTURE OF THE NEURAL CONTROL SYSTEM

Figure 1(a) illustrates our proposed neural architecture. The trajectory tracking control without obstacles is implemented by the SODMN and a neural network of biological behaviour implements the avoidance behaviour of obstacles.

Self-Organization Direction Mapping Network (SODMN)

The transformation of spatial directions to wheels angular velocities is expressed like a linear mapping and is shown in Fig. 1(b). The spatial error is computed to get a spatial direction vector (DVs). The DVs is transformed by the *direction mapping network* elements V_{ik} to corresponding motor direction vector (DVm). On the other hand, a set of tonically active inhibitory cells, which receive broad-based inputs that determine the context of a motor action, was implemented as a context field. The context field selects the V_{ik} elements based on the wheels angular velocities configuration.

A speed-control GO signal acts as a non-specific multiplicative gate and controls the movement's overall speed. The GO signal is an input from a decision centre in the brain, and starts at zero before movement and then grows smoothly to a positive value as the movement develops. During the learning, the GO signal is inactive.

Activities of cells of the DVs and DVm are represented in the neural network by quantities (S_1, S_2, ..., S_m) and (R_1, R_2, ..., R_n), respectively. The direction mapping is formed with a field of cells with activities V_{ik}. Each V_{ik} cell receives the complete set of spatial inputs S_j, $j = 1, ..., m$, but connects to only one R_i cell. The direction mapping cells ($\mathbf{V} \in \mathbb{R}^{n \times k}$) compute a difference of activity between the spatial and motor direction vectors via feedback from DVm. During learning, this difference drives the adjustment of the weights. During performance, the difference drives DVm activity to the value encoded in the learned mapping.

A context field cell pauses when it recognizes a particular velocity state (i.e., a velocity configuration) on its inputs, and thereby disinhibits its target cells. The target cells (direction mapping cells) are completely shut off when their context cells are active (see Fig. 1(b)). Each context field cell projects to a set of