Multimodule Artificial Neural Network Architectures for Autonomous Robot Control Through Behavior Modulation

J. A. Becerra, J. Santos and R. J. Duro

Grupo de Sistemas Autónomos, Universidade da Coruña, Spain
ronin@udc.es, santos@udc.es, richard@udc.es

Abstract. In this paper we consider one of the big challenges when constructing modular behavior architectures for the control of real systems, that is, how to decide which module or combination of modules takes control of the actuators in order to implement the behavior the robot must perform when confronted with a perceptual situation. The problem is addressed from the perspective of combinations of ANNs, each implementing a behavior, that interact through the modulation of their outputs. This approach is demonstrated using a three way predator-prey-food problem where the behavior of the individual should change depending on its energetic situation. The behavior architecture is incrementally evolved.

1. Introduction

According to Harvey [11] when designing a behavior based architecture in a bottom up fashion [4][2] several problems have to be considered. On one hand, the decomposition of a robot control system into subparts is not always evident. Furthermore, the interactions between modules are more complex than direct links. Some of them occur through the environment and, as the complexity of the system grows, the interactions between modules grow exponentially. Thus, as the desired complexity increases, it is more difficult to design the systems.

To address the problem of designing complex behavioral systems different options are possible, from a single module that includes all the necessary mappings between the sensors and actuators, that is, a monolithic approach, to all kinds of multi module architectures. For obvious reasons, a monolithic approach is not practical in systems that must grow in time and where one would like to have reusable components. Consequently, in our work we must look towards multi-module architectures, that is, the global behavior is decomposed into a set of simpler ones, each implemented in its own controller, and a method generated to interconnect them in such a way that a final behavior is presented to the actuators.

There are many approaches to the implementation of multi-module architectures: hierarchical, distributed, hybrid, but the main problem is how to regulate which modules have access to the actuators and in what proportion. Even more important, how can we obtain this global actuation in such a way that the results occurring in situations that have not been previously seen are meaningful?

In hierarchical modular architectures, the global behavior is decomposed, as necessary, into lower level behaviors that will be implemented in particular

J. Mira (Ed.): IWANN 2003, LNCS 2687, pp. 169-176, 2003.

controllers. The higher-level controllers can take information from the sensors or from low-level controllers, and depending on the architecture, act over the actuators or select a lower level controller for activation. The advantage of these methods is that the behaviors can be obtained individually and then the interconnection between them can be established. Also, it is possible to reuse the behaviors obtained when implementing higher-level behaviors. The problem that arises is that the decomposition is not clear in every case, as it implies a specific knowledge of what sub-behaviors must be employed. This, in general, implies a greater participation of the designer in the process of obtaining a global controller. Examples are the subsumption architecture of Brooks [4] and the use of different hierarchical architectures of Colombetti, Dorigo and Borghi [5], where the individual modules are classifier systems.

The second possibility is that of distributed architectures, where there are no hierarchies and all the controllers compete at the same level for control of the actuators each instant of time, leading to less participation of the designer. They also preserve the level of behavior reuse. However, as a drawback, they induce a higher level of difficulty when obtaining complex behaviors. Distributed architectures are exemplified by the motor schema theory of Arbib [1] and Arkin [2], that explain the behavior in terms of the concurrent control of many activities, where there is no arbitration and each behavior contributes in varying degrees to the robot's overall response. As an example of distributed architectures obtained through evolution, in [13] the authors use an incremental distributed architecture, starting from a group of pre-established behavior modules or with learning capacity, to encode, in the genotype of incremental length, the "activation network" between the modules. The architecture is incremental in the sense that, if new modules are added, the current activation network is preserved, so that genetic operations over that network are not allowed. This greatly restricts the solutions that could be obtained.

An alternative classification of control architectures, introduced by Pfeifer and Scheier [12], when obtaining an emergent behavior from a set of modules or basic processes is to differentiate between competitive and cooperative coordination. In the former only one process writes its output to the actuators each moment of time; the others are inhibited or not used. Examples are the subsumption architecture [4], sequencing, in which process outputs are sent to the actuators in a temporal sequence; or the winner-take-all strategy, in which processes compete against others to win the control of the actuators (Urzelai et. al [13]). In cooperative coordination, the outputs of two of more processes that control the same actuators are combined into a single output to be sent to the actuators, usually through summation with different degrees, as the previously commented case of motor schemas [1][2].

One way to bridge the gap between hierarchical architectures, usually competitive due to the enabling-disabling nature of the top level controllers and the ability to have different behaviors contributing to the same output in a given instant of time as presented in cooperative architectures, usually distributed, is to make use of the concept of modulation which would allow behaviors to interact without constraints and thus obtain a greater coherence degree and continuity in the global operation of the multi-behavior architecture. In that line, we are going to consider the application of output modulation as a way to increase the power of hierarchical structures (and generalizing them), to realize a graceful and continuous transition