Applied Soft Computing 22 (2014) 358–371

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A fault tolerant single-chip intelligent agent with feature extraction capability

Koldo Basterretxea[a,*], María Victoria Martínez[b], Inés del Campo[b], Javier Echanobe[b]

[a] Department of Electronics Technology, Industrial Technical Engineering School of Bilbao, University of the Basque Country (UPV/EHU), Paseo Rafael Moreno 3, 48013 Bilbao, Basque Country, Spain

[b] Department of Electricity and Electronics, Faculty of Science and Technology, University of the Basque Country (UPV/EHU), Leioa, Barrio Sarriena s/n, 48940 Basque Country, Spain

ARTICLE INFO

Article history:
Received 22 January 2013
Received in revised form 17 February 2014
Accepted 22 May 2014
Available online 2 June 2014

Keywords:
Autonomous intelligent agent
Neuro-fuzzy systems
System-on-chip
Fault tolerance
Principal component analysis
Ambient intelligence

ABSTRACT

Autonomy and adaptability are key features of intelligent agents. Many applications of intelligent agents, such as the control of ambient intelligence environments and autonomous intelligent robotic systems, require the processing of information coming in from many available sensors to produce adequate output responses in changing scenarios. Autonomy, in these cases, applies not only to the ability of the agent to produce correct outputs without human guidance, but also to its ubiquity and/or portability, low-power consumption and integrability. In this sense, an embedded electronic system implementation paradigm can be applied to the design of autonomous intelligent agents in order to satisfy the above mentioned characteristics. However, processing complex computational intelligence algorithms with tight delay constraints in resource-constrained and low power embedded systems is a challenging engineering problem. In this paper a single-chip intelligent agent based on a computationally efficient neuro-fuzzy information processing core is described. The system has been endowed with an information preprocessing module based on Principal Component Analysis (PCA) that permits a substantial reduction of the input space dimensionality with little loss of modeling capability. Moreover, the PCA module has been tested as a means to achieve deep adaptability in changing environment dynamics and to endow the agent with fault tolerance in the presence of sensor failures. For data driven trials and research, a data set obtained from an experimental intelligent inhabited environment has been used as a benchmark system.

1. Introduction

The term "intelligent agent" applies, in a broad sense, to a computer system that is capable of autonomous action in the environment it is placed in. To accomplish this task, the system must perceive its environment through sensors and must act upon it through effectors. Although there is not a universally accepted definition of what exactly an intelligent agent is, there are some properties that any intelligent agent is supposed to comprise, such as, autonomy, adaptability, reactivity, and proactivity. In particular, autonomy is always stressed as a key feature of intelligent agents, usually meaning the ability to act without the intervention of humans or other systems and to adapt their principles of behavior [1]. Being intelligent agents -at least in their initial phase of development- basically a paradigm for developing software applications, the concept of autonomy has traditionally been focused on algorithmic or functional independence from human guidance [2]. However, when intelligent agents are applied to domains which imply ubiquity (ambient intelligence, internet of things), or mobility (autonomous robots, spacecrafts, etc.) and concepts such as pervasive computing, power autonomy, miniaturization, and mobile sensing are involved, the concept of autonomy must be applied not only to the *agent program* but also to the processing device or *agent architecture*.

The implementation of autonomous intelligent agents to produce invisible (hardware transparency), low-power, embedded systems able to incorporate the information coming in from multiple-sensor platforms and to "intelligently" process that information to produce outputs, involves many design challenges both in the computational paradigm to be applied and in the architectural or hardware structure to be created. Regarding this, the System-on-Chip (SoC) approach in which the processing cores,

* Corresponding author. Tel.: +34 946014302.
E-mail addresses: koldo.basterretxea@ehu.es (K. Basterretxea), victoria.martinez@ehu.es (M.V. Martínez), ines.delcampo@ehu.es (I. del Campo), franciscojavier.echanove@ehu.es (J. Echanobe).

http://dx.doi.org/10.1016/j.asoc.2014.05.022
1568-4946/

the memory and all interface blocks are integrated in a single chip, is a very appropriate solution to design and implement autonomous embedded intelligent agents [3,4]. SoC implementations produce less power-hungry, smaller and more reliable systems compared to multichip implementations. An optimized SoC design, however, implies applying combined hardware and software design techniques, and requires a thorough knowledge of both the target technology and the intended application. These difficulties can be notably eased using FPGA (Field Programmable Gate Array) technology, which at the same time allows for high architectural adaptability and application versatility [5]. At the same time, FPGAs are moving to leading edge process technologies, and are being widely adopted in embedded system solutions because of their comparatively reduced cost in high-end embedded applications [6].

In this paper the design and implementation of a single-chip autonomous intelligent agent embedded in an FPGA is described. The agent intelligence is provided by a neuro-fuzzy computing core based on the PWM-FIS (Piece-Wise Multilinear-Fuzzy Inference System) architecture [7], which is a modification of the well known ANFIS (Adaptive Neuro-Fuzzy Inference System) architecture [8]. The applied modifications produce a more hardware friendly processing scheme, in order to obtain high performance neuro-fuzzy processors able to operate in real time (RT) with very short sampling times. Neural, fuzzy and hybrid neuro-fuzzy paradigms have already been adopted by some researchers to implement intelligent agents [9–20]. The neuro-fuzzy computational scheme embodies both main components of an intelligent agent: the *agent program* and the *agent architecture*, since knowledge is codified in its processing structure and in its parameterization. Self-adaptability and autonomy is achieved by means of learning algorithms that can be applied to optimize system performance in changing environments and to produce a system whose behavior is determined by its own experience (autonomy by learning from the environment). Yet there is an important drawback of neuro-fuzzy systems for their application to highly multidimensional problems when fast operation and/or reduced area and power is required, which is known as the *curse of dimensionality*. This phenomenon affects neuro-fuzzy processing in two ways: On the one hand, it makes learning processes much more ineffective and complex, slowing down system adaptation and sometimes impeding optimization algorithms from converging. On the other hand, if neuro-fuzzy high computing speed wants to be maintained by parallel processing architectures, it produces such an architectural size increase that it makes very costly or even impossible to embed such a system in a single chip.

To avoid the curse of dimensionality, increase learning and adaptation performances, and produce relatively small embeddable neuro-fuzzy intelligent agents, we have added a Principal Component Analysis (PCA) preprocessor to the agent architecture. PCA is a linear technique often used to identify patterns in data and, eventually, reduce data space dimensionality with little loss of information [21]. This is a very relevant technique to be adopted in many application areas of intelligent agents since very often these are multi-sensor scenarios in which thousands of data are collected from many system features or sensed variables. In reality, not always all sensed data are relevant to the agent operation and very often these data are highly correlated, or their relevance may change depending on the evolution of the system/environment. In this sense the PCA processor is a feature extraction system that may also contribute to the interpretability of the neuro-fuzzy system operation. A PCA preprocessor can also be a good means for fault tolerance when communication from one or more sensors is lost. In this case, PCA can reorganize the input space to minimize loss of information and, eventually, improve the robustness of the system for fault tolerant operation.

To analyze how a single-chip neuro-fuzzy intelligent agent holding the properties described above can be designed and implemented, we used data from an ambient intelligence application which was carried out experimentally at the University of Essex by the Intelligent Inhabited Environment Group [11]. Researchers collected thousands of data in a monitored living space with an ubiquitous networked sensor system arrangement. The data were collected from the interaction with the environment of different users of this experimental "intelligent dormitory" during periods of several days in various seasons of the year. This is not a specially demanding application in terms of processing speed (at least in what refers to the input/output signal delay of the agent), but in any other aspect of the requirements of an embedded intelligent agent application (small size, low-power, multiple input/multiple output system, adaptability to a changing environment and autonomy) this is a very suitable real-world data set for the development of this research. In fact, trying to model and predict human activity and preferences over time, even for simple human-environment interactions, is an extremely challenging problem.

The paper is organized as follows: Section 2 is devoted to describing how PCA can be applied to the feature extraction and information rearrangement in the experimental data set. In Section 3 the proposed neuro-fuzzy processing scheme and its modification to operate with a PCA preprocessor that eventually reduces input space dimensionality is explained. The modeling capability of such a system is analyzed by performing some simulation-based experimental tests on the mentioned data-sets. The potential of PCA to obtain a more robust agent in the presence of sensor or communication failures is also investigated in this section. Section 4 is dedicated to a detailed description of the SoC design of the proposed system. Both the software partition and the hardware partition of the system are accurately explained and experimental performance and implementation figures are given. In Section 5 some selected simulation runs already presented in Section 3 are compared with the real operation of the agent implemented on an FPGA to show the actual performance of the real system. Finally some concluding remarks are given in Section 6.

2. Feature extraction by PCA preprocessing

As explained in the previous section, adding a PCA preprocessor to the input stage of the agent's architecture would modify the way acquired data is presented to the neuro-fuzzy information processing system. The object of this preprocessor is twofold. Firstly, to analyze the information provided by the multiple sensors distributed around the experimental environment and, eventually, to reduce the input space dimensionality with little loss of information. Secondly, to provide fault tolerance in case the communication with one or more of the sensors is lost or recovered

Principal component analysis is a linear non-parametric method for extracting relevant information from complex data sets [21,22], and it is a standard technique widely used for dimension reduction in statistical pattern recognition. Many experimental data sets are obtained from multiple sources (sensors), often with little or no previous knowledge of the underlying dynamics of the system being observed and modeled. In consequence, the collected information regarding some of the variables or presumed *features*, which form the basis of the input space of the system, may not be significantly meaningful to describe system dynamics, and some of the selected variables may be highly correlated (information redundancy).

As mentioned in the introduction, the *Intelligent Inhabited Environment Group* from the University of Essex provided us with experimental data from some experiments carried out at their *Intelligent Dormitory* (iDorm). This test bed was designed for the