LOOKING FOR MASCONTROL: A MULTIAGENT SYSTEM FOR IDENTIFICATION AND CONTROL

E. J. González, Alberto Hamilton, L. Moreno, R. L. Marichal, J.A. Méndez, Vanessa Muñoz
Grupo de Computadoras y Control, Dept. Física Fund. y Exp., Elect. y Sistemas,
Av. Astrofísico Fco. Sánchez S/N, CP. 38207, La Laguna, Spain

Keywords: Identification, control, MAS, ontologies.

Abstract: In this paper, MASCONTROL, a multiagent system (MAS) for system identification and process control, is presented. This MAS implements a self-tuning regulator (STR) scheme. In this work, an Ontology Agent (OA) is included, using DAML+OIL as ontology language. From their experience, the authors consider this architecture highly useful for identification and control processes.

1 INTRODUCTION

Multiagent Systems (MAS) (Faratin et al, 2000) have been shown to be an effective tool in a wide range of applications. Nevertheless, researchers involved in process automation have not taken significantly advantage of this tool (Cockburn et al, 1996, Velasco et al, 1996, Seilonen et al, 2002). Some reasons can be found in (Seilonen et al, 2002): automation applications need real-time requirements that are out of currently agent technology reach, difficult agent treatment about interrelations and lack of parallelism.

The aim of this paper consists of presenting the development of a multiagent system oriented to process automation. The development of this standarized and complex multiagent system, called MASCONTROL is detailed in Sections 2-6: a preliminary introduction, a justification of the selected tools, description of the designed framework and the designed ontology (this multiagent system includes an ontology agent for communication with external agents) and results obtained in the control process of a real plant. Finally, conclusions are reported in Section 7.

2 INTRODUCTION TO MASCONTROL

FIPA (Foundation for Intelligent Physical Agents) specifications have become a stronger standard in MAS development and they involve not only agent language specifications but also agent management, conversations, etc. The FIPA agent management reference model provides the normative framework within which FIPA agents exist and operate. The Directory Facilitator (DF) provides yellow pages services to agents that query it to find out services offered by other agents. The Agent Management System (AMS) offers white pages services and maintains a directory, which contains transport addresses for agents registered in the Agent Platform (AP). The Message Transport Service (MTS) is the default communication among agents on different APs.

In this paper, the paper have applied the power of the MAS to control disciplines. In particular, the involved agents are supposed to implement the different roles related to a self-tuning regulator, STR (Aström and Wittenmark, 1995, Söderström and Stoica, 1989, Warwick, 1989). The authors consider this problem as a very interesting one in the application of MAS. The first reason is that a STR presents conceptually different modules, such as sampling, identification or control. Other reason consists of the parallelism in the system, optimizing the calculation capacity this way. Both reasons make MAS appropriate.

3 SELECTION OF THE TOOLS

This section justifies briefly the selection of the used tools for the MAS development. It is pretended to be

J. González E., Hamilton A., Moreno L., L. Marichal R., A. Méndez J. and Muñoz V. (2005).
LOOKING FOR MASCONTROL: A MULTIAGENT SYSTEM FOR IDENTIFICATION AND CONTROL.
In *Proceedings of the Second International Conference on Informatics in Control, Automation and Robotics*, pages 119-124
DOI: 10.5220/0001173401190124

an elementary survey guide for those researchers that would like to use a MAS in their projects.

3.1 Multiagent framework

Once FIPA standard has been chosen for the development of the system, the following step consists of the selection of the multiagent framework. This fact involves being not worried about technical aspects such as a detailed communication implementation. With regard to this, there are many tools for the design and implementation of MAS.
In this sense, the authors have chosen FIPA-OS. This is an Open Source agent framework from research at Nortel Network's Harlow Laboratories that implements the FIPA specifications about agent interoperability. There are other Java-based implementations of FIPA specifications such as JADE and ZEUS. However, FIPA-OS has been chosen in this work due to

> its well-placed use of Java interfaces to separate agent subsystems, translation of incoming messages and system occurrences into events for internal processing, an isolated scheduling policy for task execution, use of a conversation object to enforce protocols and hold messages, and a task generation tool for constructing tasks from protocol definitions (Fonseca, 2002)

3.2 Ontology Language

There are few situations where a standalone agent is able to manage an open and dynamic environment: agents are designed for communicating each other. In this sense, having a shared ontology (Falasconi et al, 1996) makes a multiagent system more powerful, as agents developed by different designers can interact in the same environment. In this case, the problem to solve is too easy (only few terms in the ontology are necessary for the problem, as can be seen below). However, it is interesting to consider its use in more complex situations.
For ontology design, there are many possible ontology languages. The authors have used DAML+OIL, a markup language, as ontology language, because it provides a basic infrastructure that allows a machine to make some sorts of simple inferences that human beings do and because this language has become a standard in ontology representations. At this moment a new, improved and more expressive markup language, known as OWL, is being developed. Nevertheless DAML+OIL is sufficiently expressive for carrying out this work.
This selection involves the use of some related software, in particular an ontology editor (OilEd has been chosen) and a DAML+OIL parser (Jena).

3.3 Evenet2000

The problem of optimization is much related to identification and control. In both disciplines, it is usually necessary to optimize some parameters for a model (identification) or for a controller. In the same way, the training of a neural network consists of finding the best values of the weights of the network. Because of this similarity of methods, the authors have considered the possibility of the application of neural networks training methods to control problems.
For this MAS, the authors have preferred used Evenet2000 (Gonzalez et al, 2001 and 2003), a Java-based neural network toolbox developed at the University of La Laguna. This Open-source tool offers some features such as the possibility of designing an arbitrary neural network to optimize. This tool develops an approach to derive gradient algorithms for time-dependent neural networks, using the *Signal Flow Graph theory.* Moreover, the designed structure makes it not to be limited to algorithms based on the gradient. This tool has been shown as highly useful in a suite of control problems.
In MASCONTROL, Evenet2000 is used in two ways. Firstly, for the optimization of the parameters of a model. Secondly, in the optimization of the parameters of a controller such as the value of K_p in a proportional controller. The different agents import Evenet2000 packages for its use (Gonzalez et al, 2004).

4 MASCONTROL AGENT FRAMEWORK

This framework is composed of 8 different types of agents. Each agent assumes one of the roles that appear in a STR scheme.
ReaderCommandAgent (RCA) This agent samples the output of the system. Another role assumed by this agent consists of calculating and sending the command to the system. Finally, it stores a vector containing input-output data. This vector makes

possible the system identification and determines if the input is rich enough for that identification.

IdentificationAgent (IA) Several identification agents can appear in the system. Each IA tries to identify the system from the input-output vector. For this purpose, it uses Evenet2000 modules. A system user or an IdentificationLoaderAgent (from a record of previous trainings) can select a training method for each IA. The system model is declared in the XML profile and loaded as an Evenet2000 neural network file. In other words, for an IA, the problem is equivalent to the one of weight optimization in a neural network whose training pattern is defined by the input-output vector. The authors have included a one-step ahead predictor in the identification process.

LinearIdentificationAgent (LIA) Similar to the IA, but this agent assumes a model that allows a linear regression, for example an ARX model. In this sense, the model is defined by the orders of the numerator and the denominator of the transfer function. Instead of Evenet2000 modules, an object that implements an identification through the Forgetting Factor technique is used (Aström and Wittenmark, 1995, Söderström and Stoica, 1989, Warwick, 1989).

CentralIdentificationAgent (CIA) This agent manages the Ias, linear or not. Initially it asks the DF for the agents that offer the service of identification in the MAS. Every T seconds, the CIA asks the IAs for the results of the current optimizations (error, parameter set and model file), selecting that optimization offering the best results. Then it informs the rest of the IAs with the same model the set of parameters that minimize the criterion function. This way, the IAs take this set as an initial training one in new optimization processes. With the aim of providing the MAS with some intelligence, the CIA counts how many times each training method obtains the best results optimizing the criterion function. This information is stored in a file (written in DAML+OIL), that could be used in the initiatization other IAs.

OptimizerAgent (OpA) This agent optimizes the controller parameters. For this purpose, it takes the set calculated by the identification agents and includes it as constants in a new system. This system is treated as a neural network whose parameters are the controller ones. In a general way, patterns are chosen as pairs *reference input, reference input* in a serie of different reference inputs. With this implementation, high raising time and valued peaks are penalized. The model of the system can be easily changed due to Evenet2000 modularity.

CentralControlAgent (CCA) This agent is similar to CIA as it plays a manager role in the system, but in this case it is related to the optimization of model parameters. Each T seconds, the CCA asks the CIA for the details (model with the best results, parameter set) of the identification. After analyzing these data, the CCA asks the OpAs for the parameters that minimize the criterion function and for the value of this minimization. This agent stores the results for subsequent sending to the RCA. Finally, the CCA orders the OpAs to stop the current optimizations and to start a new one from the calculated optimal parameter set. As the CIA, the CCA stores the number of times that a given training method has offered the best results for the analyzed control process.

InputOutputAnalyzerAgent (IOAA) This optional agent analyzes process input and output data (calculated by the RCA). This analysis is made in two ways. First, it tests, in an intuitive way, if the system input is rich enough. For this purpose, this agent calculates the maximum and minimum input value in the last N periods, and it tests if the subtraction of these values is less than a defined low enough threshold. If it is the case, it is supposed that the applied input is not rich enough and the IOAA suggests the RCA that the reference input should be changed. This change is supposed to improve the identification process. In a similar way, output data are analyzed too. In this case, the IOAA could suggest a reference input change or a study about the type of the system. The option of changing the reference input can be inhibited on-line through a user interface.

Ontology Agent (OA) This is one of the key considerations in this phase of the work. This way, it differs from other MAS-based control systems. The definition of an external ontology provides numerous advantages: it allows consultation with respect to concepts, the updating and use of ontologies and the elimination of the need of programming the entire ontology in every agent, hence reducing required resources. Currently, in MASCONTROL, this agent only takes part in the study of the type of the system.

5 MASCONTROL ONTOLOGY

This section will describe the main features defined in the implemented ontology. As stated above, most of the concepts have not been used in MASCONTROL. However, they are defined looking for an open system where a new agent,

implemented by other researchers, could interact with MASCONTROL agents. Defined classes are mainly related to control concepts, such as System, Input, Output, ReferenceInput, Error…
A second group of classes are referred to the implemented optimization algorithms in Evenet2000: OptimizationMethod, TrainingMethod. As an example, the following DAML+OIL code defines NonLinearSystem as a subclass of System.

```
<daml:Class rdf:about="#NonLinearSystem">
<rdfs:subClassOf>
  <daml:Class rdf:about="#System"/>
 </rdfs:subClassOf>
</daml:Class>
```

With respect to the defined properties, these are mainly referred to the values of defined classes: valueOfPole, valueOfZero. On the other hand, other properties are related to control concepts such as order and type of a system. These definitions allow the system to make some interesting inferences from some axioms defined in the ontology. For example, if a given system does not reach a desired reference input with a proportional control action, the MAS could deduce that the type of the system is 0. A suitable control action for a system like that is a proportional-integral (PI) one. Therefore, an agent can deduce that a PI control action is a suitable control action for a system that does not reach a desired reference input with a proportional control action and when this axiom does not appear explicitly in the ontology.

6 RESULTS

With the proposed agent architecture, several experiments related to the control of a plant have been carried out.

6.1 Description of the plant

The plant consists of an educational module for the control of processes. This module, shown in the Figure 1, is from the PROCON (Process Control Trainer) series of the Feedback Company. for the level and flow control.
The plant consists of a closed-circuit of water that allows the study of the principles of control considering the water level and flow as variables to control. The plant includes a dual tank in the top where water level is measured. This tank is connected to a store-tank through five manual and three solenoid valves. Water is pumped from the store-tank to the dual tank through a flowmeter and a servovalve. The plant is controlled through a standard 4-20 mA action, although for implementation purposes the reference values (and the outputs) are set/measured in volts. This limitation makes more difficult the optimization of the controller parameters, introducing a non-derivability in the Criterion Function. This fact is perfectly managed by the modules designed in Evenet2000 although it can be defined a Criterion Function that would punish those values out of the allowed ones.

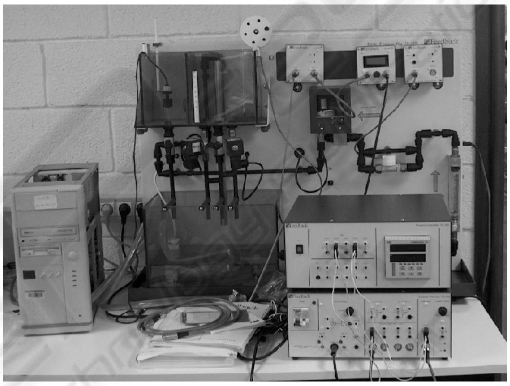

Figure 1: Plant controlled by MASCONTROL

6.2 Experiments

The experiments were carried out using three different control actions: proportional, proportional-integral and pole placement. The identification process has been carried out through several IAs acting simultaneously in the multiagent system. These agents have used several models for the plant, both linear and not linear. Moreover, the identification and the optimization of the parameters of the controller have been carried out using simultaneously different optimization methods. With an initial value k_p=1, a learning phase is carried. In this phase reference input value is continually modified, looking for a better identification. When it is considered that identification is good enough (after a long time, when parameters values are stabilized or manually through a user-interface), reference input is set to the desired value.

In the following sub-subsections the authors will analyze the results related to the PI and pole placement control actions.

6.2.1 PI Control Action

Figure 2 shows the optimization carried out by the OpAs. with a variable reference (3, 1, 2, 0.8 and 1.5 V.). In this sense, with this type of sequence (increasing/decreasing), the model behaviour is supposed to be more independent from a given reference input.
In this case, it can be seen that the system output reaches the reference input in each section, without overshoot. This optimization is reflected in the system output: it reaches the desired reference inputs (2.5, 3.7, 4 and 3 V) and with a low overshoot. The two mentioned phases (training and desired reference input) can be distinguished (see Figure 3). Moreover, the measured constant time (150 seconds aprox.) is of the same order as the measured one. This fact is considered to be a validation of the MAS.

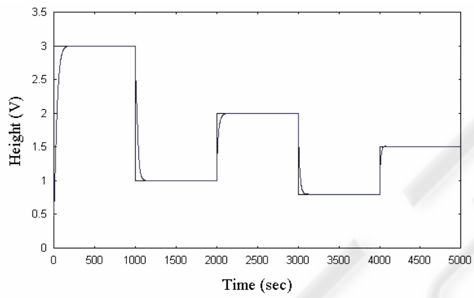

Figure 2: Optimization result obtained by the OpAs

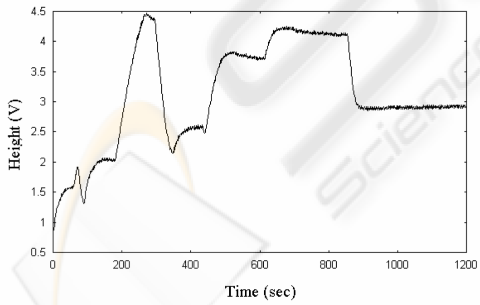

Figure 3: Results obtained using PI action control

The following experiment illustrates the use of an OA, and ontologies in general, in a control system. Figure 3 shows the output evolution initially controlled by a proportional control action. When the IOAA realizes that the system output is stabilized and that it has not reached the reference input, the RCA asks the OA for the suitable control action when a system does not reach a reference input using a proportional action. At this point, the OA looks for this fact in the system ontology and informs the RCA that the answer is a proportional-integral control action. Then, MASCONTROL makes the necessary modifications (control action and model). As it was expected, the process output reaches the desired reference input after these modifications. This fact is only an example of the power that ontologies are able to supply to a process control system.

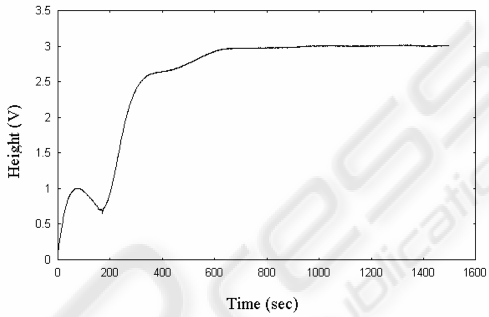

Figure 4: Ontology Effect: P to PI control action

6.2.2 Pole replacement control action

The following work focuses on studying that goodness for a totally different control action, in particular, pole replacement. This control action depends critically on system identification.

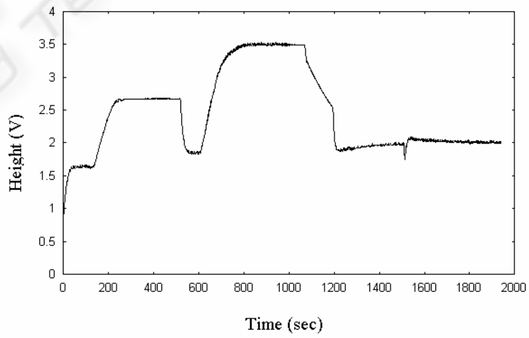

Figure 5: Pole replacement control action

Due to MASCONTROL modularity, some minor modifications are necessary. Firstly, the RCA should implement this new control action. Secondly, controller parameter set optimization has no sense for this control action. Therefore, the CCA is only limited to propagate CIA results. Figure 5 shows the obtained results. In this case, as it can be seen, the system identification (with proportional-integral control action) phase has been longer, looking for a better identification.

7 CONCLUSIONS

The development of a MAS architecture for process control is presented in this paper. In particular this MAS, called MASCONTROL implements a self-tuning regulator (STR) scheme. It is another alternative to solve problems of process control. This MAS has adopted FIPA specifications because they have become a stronger standard in MAS development and it involves not only agent language specifications but also agent management and conversations.

In this work, an Ontology Agent (OA) is included. An OA is an agent that provides access to one or more ontology servers and which provide ontology services to an agent community. For ontology design, DAML+OIL is a sufficiently expressive language for carrying out this work. Apart from the standard tools, the authors have included Evenet2000. This toolkit has been shown to be highly useful in control problems.

The authors strongly recommend the use of MAS and ontologies in order to develop applications such as those analyzed in this paper. The results that are reported in this paper indicete good future possibilities. They expect to have shown that its use offers more than they cost.

REFERENCES

Aström K.J. and Wittenmark B., 1995, Adaptive Control - Second Edition. Addison-Wesley (ISBN 0-201-55866-1).

Cockburn D. and Jennings. N.R., 1996, *ARCHON: A Distributed Artificial Intelligence System for Industrial Applications*. In Foundations of Distributed Artificial Intelligence, G. M. P. O'Hare and N. R. Jennings, editors.

Falasconi S., Lanzola G., Stefanelli M., 1996, *Using ontologies in Multi-Agent Systems*. Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop, 1996

Faratin P.; Jennings N.R.; Buckle P. and Sierra C., 2000. *Automated Negotiation for Provisioning Virtual Private Networks using FIPA-Compliant Agents*. Proc. 5th Int. Conf. on the Pratcical Application of Intelligent Agents and Multi- Agent Systems (PAAM-2000), Manchester, UK, 185-202

Fonseca S.P., Griss M.L., Letsinger R. *Agent Behavior Architectures. A MAS Framework Comparison*. Proceedings of the first international joint conference on Autonomous agents and multiagent systems, Bologna, Italy, 2002.

González E.J. , Hamilton A., Moreno L., Marichal R.L., Muñoz V., 2004, *MASCONTROL: A MAS for System Identification and Process Control,* 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, August 9-13.

Gonzalez E.J., Hamilton A., Moreno L., Aguilar R.M., Marichal R.L., 2003, *Neural networks teaching using Evenet-2000.* Computer Applications in Engineering Education. Volume 11, Issue 1, 1-5.

Gonzalez E.J., Hamilton A., Moreno L., Sigut J., Marichal R.L., 2001, *Evenet-2000: Designing and Training Arbitrary Neural Networks in Java*. Bio-Inspired Applications of Conectionism. Springer-Verlag (2085) 104-110. 2001.

González E.J., Marichal R.L., Hamilton A.F., Moreno L., 2003, *Evenet2000 as Software Application for System Identification and Control*. The 11th Mediterranean Conference on Control and Automation. Rhodes. Greece.

Seilonen I., Appelqvist P., Vainio M., Halme A.and Koskinen K., 2002, *A Concept of an Agent-Augmented Process Automation System*. 15th IFAC World Congress on Automatic Control, Barcelona, Spain.

Söderström T., Stoica P., 1989, System Identification. Prentice Hall International Series in Systems and Control Engineering.

Velasco J., González J.C., Magdalena L. and Iglesias C., 1996, *Multiagent-based Control Systems: a Hybrid Approach to Distributed Process.* Control Engineering Practice. Vol. 4 839-846.

Warwick K., 1989, *Control Systems: An Introduction*. Prentice Hall International Series in Systems and Control Engineering.