Distributed Intelligent Control for a Mine Refrigeration System

I.M. MacLeod and A. Stothert

One way to construct intelligent controllers is to use an agent-centered approach in which the agents themselves determine the global structure of the controller and the inter-agent cooperation methods. To assess this approach, the design and testing of a distributed intelligent controller for a laboratory-scale mine refrigeration plant is discussed. Underlying theoretical concepts are briefly reviewed, and experimental results are presented. Important findings are that the approach supports the use of multiple knowledge representations, the use of partial results, and dynamic structuring of the distributed controller. The results presented indicate that development for practical, supervisory-level control applications is worthy of further investigation.

Introduction

The community metaphor, as applied in distributed artificial intelligence systems, is well known [1, 2]. The suitability of intelligent control techniques for large-scale industrial processes has also been well highlighted [3, 4, 5]. The result, when the community metaphor approach is combined with intelligent control techniques, is a distributed intelligent control system. This is a specialized application of multi-agent systems [6] and has distinct advantages over traditional control approaches in terms of distribution (both physical and logical), diversity (heterogeneity of modeling techniques), flexibility, and reliability.

This article examines the application of distributed intelligent control techniques to a specific problem, namely, an underground refrigerated water reticulation system. This kind of system has, to date, had to be controlled manually. The article aims to illustrate the design and explain the operation of the resulting self-structuring distributed intelligent control system.

The article is arranged as follows. The second section (next) briefly introduces the distributed/multi-agent system framework. The third section describes the refrigeration system, and the fourth section discusses the proposed distributed intelligent controller. Some experimental results are then discussed, and the final section concludes the article by examining the characteristics of the controller and indicating how these could be generalized in future distributed intelligent control systems.

Distributed Intelligent Control Framework

Distributed artificial intelligence [7, 8, 9, 6, 10, 11] focuses on the use of multiple autonomous programs (agents) that communicate with one another to facilitate cooperation. This is known as the community metaphor. Through cooperation, the agents interact constructively to achieve some "social" goal(s) [12]. A common approach has been to apply social interaction and community theories to agent design, with the result that agent interaction takes the form of various cooperation mechanisms, e.g., tacit agreement, dictatorial, democratic, fully cooperative, or antagonistic. At the implementation level, cooperation is generally achieved via communication between agents.

In message-passing systems, where agents communicate only by exchanging information, an agent's "view" of the global distributed system is dependent on what other agents tell it. At its most basic level, message passing facilitates cooperation through information exchange. More complex cooperation schemes can also be implemented using message passing, e.g., blackboard architectures [13, 14] and negotiation [15], both of which provide mechanisms for opportunistic cooperation between agents.

Our approach to the application of distributed artificial intelligence techniques to control problems is to focus on the implemented agents [16]. This results in an agent-centered approach, similar to that advocated by Werner [12], which allows the agents themselves to determine [17] the global structure of the distributed system, i.e., the number and type of agents in the system, who the agents communicate with, and the type of cooperation the agents employ. For example, each agent template could include a list of the cooperation mechanisms that the agent is capable of using.

The idea is to have an off-line "pool" of pre-defined agent types (templates are used to define agents). Individual agents are created by existing agents from the pool during operation. Note that a human operator can be treated as an agent. Cooperation, in the sense of the community metaphor, is achieved in this framework by the dynamic creation of groups of agents in conjunction with traditional methods such as communication between agents.

Central to the approach is the use of individual agent behavior and role, and the use of rules within each agent to realize that behavior and role. The rules used to capture an agent's role and behavior are collectively called interaction rules. It is these interaction rules that enable self-organization and dynamic structuring of the global distributed system. Practically, this means that distributed intelligent controllers are capable of adapting to suit the plant being controlled. This kind of adaptation is within a pre-specified "solution space," not adaption in the most general sense.

Within our framework, design of distributed intelligent controllers thus focuses on the choice of what agents to implement. Practically, the agents are determined by functional decomposition of the control problem, e.g., using techniques similar to [18].
Once the agents to be implemented have been defined, both the behavior and role of each agent can be specified. The behavior and role specification is used to develop an agent's interaction rules, and these are coded into the agent template [19]. Identification of the agent's role also determines the types of communication necessary, e.g., plain message passing or negotiation, i.e., how the agent cooperates with other agents is also coded into the agent template.

A software framework within which such a distributed system can be designed and implemented has been developed. For more details see [20, 17]. This framework is used in the refrigeration plant study described below.

Refrigeration System

In deep-level mines (>3000m) where the rock temperature typically reaches 50 degrees C, refrigerated water is used to reduce the ambient temperature [21, 22]. The scaled water reticulation system shown in Fig. 1 has been constructed to serve as a laboratory test bed for different control strategies. This scaled system loosely models part of the reticulation system of an operational mine [23].

Instead of cooling the water, the laboratory model heats water. It includes a 15-liter surface dam preceded by two parallel 3kW heaters (surface “heater” plant in Fig. 1). The underground cooling system is modeled by two parallel heating legs, each with two series-connected 2kW heaters (underground “heater” plant in Fig. 1). A 100-liter drum models an underground chiller dam (demand grouping in Fig. 1). The demand pump has a maximum flow rate of 6.5 l/min, while the flow rate through each underground heater is limited to 4 l/min; the surface heaters have no flow control. At maximum flow and maximum heater power, an individual heater can raise the water temperature by approximately 10 degrees C. Knowledge of such limits must be built into the controller for this plant—it is used in the construction of a plan, described in the “Distributed Controller” section below. As indicated in Fig. 1, the model includes sensors to measure temperature and flow as well as dam levels. Control valve openings, solenoid valve status, heater powers, and pump status are the available plant control inputs. The control objective is to maintain a heated water supply in the face of process structural and load changes. The control objective is specified by an operator as a constant temperature and flow rate over a given time interval.

The sensors and actuators of the laboratory model are connected to a SCADA system [24]. In effect, the SCADA system provides a real-time database of process variables that records sensor values and allows plant actuator values to be set. It can be accessed and manipulated either via a graphical man-machine interface or via dynamic data exchange (DDE) [25]. It permits basic logic control to be specified directly in terms of database variables. For example, basic control interlocks are implemented within the SCADA system. Apart from interlock control, the SCADA system performs no control—it simply routes process variables to and from the distributed controller. When included

![Diagram of the refrigeration system](https://example.com/diagram.png)

**Fig. 1.** Scaled model of an underground refrigeration system.
Fig. 2. Agents for refrigeration plant controller.

If the inlet-outlet temperature difference is > 10, contact a supply agent for assistance.
If the inlet-outlet temperature difference is <= 10, contact the configuration agent for a suitable heater configuration.

Fig. 3. Example of rules for the underground agent.

in the distributed controller, the SCADA system can be considered as simply another agent.

Distributed Controller

The refrigeration system is intended to be operated as follows. A sequence of demand intervals for heated (cooled) water is specified by the system operator. Each demand interval comprises a start and stop time and the required temperature and flow for that time interval. It is assumed that the demand is specified well in advance of when it is actually required in order to avoid difficulties due to unreasonable deadlines. For example, at 08h00 a demand of 40 degrees C at 2 l/min for 12h00-16h00, and then 35 degrees C at 4 l/min for 20h00-0h00 with no demand from 16h00-20h00. Details of a distributed intelligent controller that is intended to cater for this kind of intended operation are now discussed.

The agents implemented for the refrigeration system are shown in Fig. 2. Three supervisory-level agents are defined, one each for the physical divisions: surface, underground, and demand (outlined in Fig. 1). Note that all the agents in Fig. 2 do not always exist in the distributed system. They are added and removed dynamically, as necessary.

Planning Agents

The three supervisory agents are responsible for system planning. Each agent implements a rule set that enables planning. Two typical rules coded in the underground agent template are shown in Fig. 3.

Because of the distributed nature of the agents and the use of rules for planning, the planning mechanism resembles a distributed expert system. Once a demand has been specified by an operator (time interval, flow, and temperature), the supervisory agents cooperate to construct a plan for meeting the demand. This plan is influenced by dam levels, ambient temperature, heater availability, heater water temperature raising capability, etc. In some cases, the plan may simply call for the demand agent to start the outlet pump at a given time. In more complex cases, the surface agent may be required to heat the inlet water.

The planning mechanism employed is simple. Note that the refrigeration system is a unidirectional stream, with water flowing in at the surface "heater" plant and leaving from the plant outlet. The supervisory agents are arranged logically in a similar stream: A demand agent is downstream from an underground agent, which is downstream from a surface agent, etc. The general sequence of planning is thus as follows.

First, the downstream agent generates a plan proposal and passes it to the upstream agent. The plan proposal passed to the upstream agent is modified to indicate the capabilities of the local agent (see example below). This proposed plan is based on the refrigeration system conditions and state known to the agent, e.g., the underground agent checks heater availability, underground dam level, heater inlet temperature, etc. If the upstream agent cannot satisfy the plan locally¹, it returns an intermediate plan to the downstream agent and it forwards a secondary plan to its own upstream agent. This upstream agent tries to satisfy the proposed secondary plan.

An example illustrating the typical planning process is provided in Fig. 4. Obviously, the details of the plans passed between the agents depend on actual demand values and plant conditions. For example, if due to maintenance heaters are unavailable, the intermediate plans are changed to reflect this, and different reasoning is applied to develop the final plan. As is evident, the plan is modified at each stage of the planning process, e.g., if an agent receives a plan request that it can partially meet, the secondary plan request passed to its upstream agent is modified to reflect this. Within this planning framework, the operator is the agent furthest downstream and the surface agent is the agent furthest upstream. If a plan cannot be fully met, e.g., because of insufficient supply, a “next best” plan is proposed. This plan could be rejected by the operator.

Additional Agents

The planning or supervisory agents have subsidiary agents, which they use to assist in plan formulation or to fulfill the plan. The demand agent is the simplest. It processes operator inputs, i.e., flow and temperature requirements, and starts the demand pump and control valve when the demand is active. Because of its simplicity, the demand agent is actually incorporated into the underground agent. The underground agent has

¹For the sake of planning, the surface dam is considered part of the underground system and agent. This is because the underground agent can satisfy a plan locally if sufficient heated water is available in the surface dam.
two subsidiary agents, one to calculate the optimal heater configuration to use and another to control the heater bank.

The heater configuration is calculated by a configuration agent that uses fuzzy logic to implement a decision matrix [23]. The fuzzy logic has two inputs (flow and temperature) and one output (configuration). The input domains have five sets and the output domain four sets. The logic employed is centered around the two conditions: all heaters for high flow, high temperature and one heater for low flow, low temperature. Given a required flow and temperature, the fuzzy logic returns a required heater configuration, e.g., two heaters in series.

The first in the sequence of demand intervals is announced. The underground dam is not capable of satisfying the demand locally due to insufficient volume; the demand agent modifies the plan to indicate the required volume (flow, temperature, interval) and passes the modified plan (a proposal) to the underground agent.

The underground agent formulates a plan but discovers that the heater inlet temperature is too low. The underground agent does two things:

First it passes a plan proposal on to the surface agent requesting the heater inlet temperature to be raised.

Second, it returns a plan proposal to the demand agent indicating that it can supply the demand requirement, but only at a reduced temperature.

In the mean time, the surface agent works on the plan proposal received from the underground agent and finds that it can satisfy the proposed plan requirements. The surface agent indicates to the underground agent that it can satisfy the proposed plan that it received.

The underground agent now indicates to the demand agent that the required demand can be met fully. Planning for the demand interval is now complete.

**Fig. 4. Planning example.**

**Experiment 1:**

**Requirements**

- Flow = 2 l/min.
- Temperature = 45°C.
- Only heater 1000 available (constraint)

**Results**

Planned heater configuration = 1000 with surface "heater" plant outlet temperature set to 31.8°C and underground outlet temperature set to 41.8°C.

**Plan Comments**

- Inlet Temp low -- attempting to heat.
- Inlet being heated.
- Only single series heater available -- limited maximum Temp.
- Plan finalised.

Prior to heating inlet water temp and adjusting for availability constraints, an underground outlet temperature of 39.2°C was proposed by the underground agent. This would have been modified to around 38°C if the surface "heater" plant was unable to heat water.

**Fig. 5. Refrigeration experiment: Plan 1 results.**

The PID agents implement a standard backward difference equation to perform PID control [26]. The PID constants and error signal and control signal sources are stored by the heater agent and communicated to newly launched PID agents. Step tests on each heater were used to determine suitable PID constants.

The refrigeration plant thus requires six agent templates to be defined: demand, underground, surface, heater, config, and PID. Each agent template implements the functionality described above as well as interaction rules that, by capturing the role and behavior of agents, define when "neighboring" agents are required.

Consider the underground agent. If it cannot satisfy the plan locally, it will launch a surface agent. The underground agent must also launch a configuration agent, and if a valid plan is available it will launch a heater agent. The heater agent only launches the required PID agents when the underground agent's plan becomes active. It terminates the agents when the plan is completed. Similarly, the surface agent launches and terminates agents as required. The result is that the number of agents in the distributed system changes dynamically to suit operating requirements and conditions.

The distributed controller described above was implemented on a two-node IEEE Control Systems
Windows NT 3.51 Ethernet-based network. Each node consisted of a single x486DX2 processor with 32MB RAM. The nodes were physically separated by 150 m and were connected over a conventional office network operating at typical traffic loads. Under worst-case conditions, the developed distributed system consists of 12 agents (distributed between the two processing nodes) and 30 communication variables (transmitted between agents).

Experimental Results

Results of experiments with the refrigeration controller planning mechanism are discussed. Ambient water temperature during the experiments was recorded as 24 degrees C. Note that the heater configuration specifies the operation of the heaters in the underground section of Fig. 1 as a four-bit word. The first two bits represent the “top” leg of the heater bank and the last two the “bottom” leg. Each two-bit nibble represents the heaters from right to left. Once a configuration has been determined, a heater agent is launched to control and configure the underground heater bank.

Fig. 5 indicates the operator demand, final plan, and some intermediate plan comments for the first experiment. Without regard to heater availability, the underground agent recognizes that the plan cannot be satisfied locally (the required output temperature is too high) and so launches a surface agent. The surface agent plans to raise the surface “heater” plant outlet temperature, and as a result the underground agent launches an agent to determine the best configuration. The returned configuration (two underground heaters in series) cannot be implemented physically, e.g., due to heater maintenance. As a result, the underground agent proposes a plan where the final outlet temperature is 41.8 degrees C, not the required 45 degrees C. As indicated in Fig. 5, prior to the underground agent cooperating with the surface agent an initial outlet temperature of 39.2 degrees C (full heater availability) and finally 35 degrees C (restricted heater availability) was proposed.

Experiment 2 (Fig. 6) outlines a situation where the underground agent believes it can satisfy the plan locally, but actually cannot because of heater availability restrictions. After suggesting a local solution, the underground agent launches a configuration agent to determine a suitable heater configuration. Due to heater maintenance the returned configuration cannot be physically realized. As a result, the underground agent launches a surface agent to see whether raising the underground inlet temperature will assist in satisfying the plan—which it does.

Experiments to test the dynamic ability of the planning mechanism were also carried out. For example, an initial plan similar to experiment 1 was developed. Heater availability was then changed. It was observed that if replacement heaters were available, they were used. If not, the controller modified the plan to reflect the unavailability of heaters, e.g., by attempting to make use of the surface heaters to raise the water temperature.

The planning results given above are for single demand intervals. Multiple intervals can be strung together to form a demand profile. The distributed system uses the planning mechanism described above to formulate a plan for each demand interval. While simple mechanisms for resolving demand conflicts between intervals have been implemented (e.g., insufficient warmup/cooldown time), mechanisms to optimize transitions between demand intervals have not yet been implemented.

Figs. 8 and 9 present results obtained in a third experiment with a user demand profile of
- 32.5 degrees C at 2 l/min for 15 minutes, followed 1 minute later by
- 49 degrees C at 4 l/min for 15 minutes
Due to heater maintenance, only heaters 0101 were available. The first demand interval (32.5 degrees C at 2 l/min for 15 minutes) was met without using the surface heaters to raise the inlet water temperature. Heater 0001 was used to raise the water temperature to 34.12 degrees C—the increased temperature setpoint compensates for ambient temperature losses as the water is transported to the underground dam. The second demand interval could not be met by the underground heaters alone (the temperature difference was too high), and as a result, a surface agent was

Experiment 2:
Requirements
- Flow = 5 l/min.
- Temperature = 40°C.
- Only heaters 0110 available (constraint)

Results
Planned Heater configuration = 0110 with the surface "heater" plant outlet being heated to 29.3°C and underground outlet set to 40°C.

Plan Comments
- No series heater branches available -- limited maximum Temp.
- Inlet being heated.
- Plan finalised.

Prior to heating the surface "heater" plant outlet an underground outlet temperature of 34.2°C was proposed by the underground agent.

Fig. 6. Refrigeration experiment: Plan 2 results.

- Inlet Temp low -- attempting to heat.
- Inlet being heated.
- No series heater branch available -- limited maximum Temp.

Fig. 7. Plan comments.
launched to heat the inlet water. In developing the final plan, the comments of Fig. 7 were generated. The final plan called for a surface agent to heat the water to 33 degrees C and the two parallel heaters 0101 to raise the temperature to 43 degrees C. The heater outlet temperature setpoints were adjusted to 45.23 degrees C to account for ambient losses. Note that planning and plan execution (starting pumps, heaters, opening valves, etc.) were completely automated by the distributed system—only the intervals and required temperatures were input by the operator. The configuration, heater, surface, and underground agents were launched on one processing node, while the PID agents were launched on both processing nodes. Note that the PID agent configuration (i.e., the number of PID agents) changed to suit each demand interval.

Fig. 8 shows how, at a physical level, the temperature and flow requirements were satisfied by the distributed control system. For real refrigeration systems, manual control by a skilled operator is necessary in order to achieve similar results. Note that heater 0100 is only used to satisfy the second demand interval (higher flow and temperature setpoints). The temperature profiles indicate that the initial demand interval temperature setpoint was met approximately 100 seconds after the interval start; the second interval temperature setpoint was met approximately 150 seconds after the interval start. A small flow disturbance was introduced into the 0001 heater 900 seconds into the experiment. The result is a drop (subsequently recovered) in water temperature and a rise in heater power. The water temperature drop soon after the flow disturbance marks the end of the first demand interval. The temperature drop is a result of the flow going to zero, heater 0001 being turned off, and conduction losses. A corresponding temperature rise is evident in Fig. 9 (underground inlet temperature) and is due to conduction from the heated water present in heater 0001. Fig. 9 also indicates how the surface agent heats the water temperature in order to meet the second demand interval. Note that conduction is possible during the time between the demand intervals because no water flows during this period.

Conclusions
A distributed intelligent controller for a laboratory-scale model of a mine refrigeration system has been discussed. Important characteristics of this approach to control include:

- the combining of different knowledge representation and manipulation techniques into one system
- the use of multiple communication paradigms
- planning and plan modification to suit operating conditions
- the use of partial results in planning, and
- support for dynamic structuring.

Three different forms of knowledge manipulation [17] were successfully combined in the prototype controller: numeric (PID agents), strict symbolic (planning rules) and fuzzy symbolic (configuration agent). Additional forms of knowledge manipulation are also present in the form of interaction rules, dynamic connections between agents, and "evolutionary" adaptation.
Communication between agents was not limited to one paradigm. Communication between the demand, underground, and surface agents is similar to a negotiation protocol [15] in which plans are proposed and modified. Other communication between agents was limited to simple poke and request messages, e.g., between the PID agents and the SCADA system. In addition to differences in communication paradigms, communication content varied. At the planning level, communication consisted of complete data structures, i.e., the entire plan is communicated (start time, stop time, temperature, flow), while at control-loop level, communication consisted of single data values. This variety of communication content indicates the power of a distributed system to facilitate both simple and complex patterns of communication. In future developments, a more generalized method of transmitting complete objects [27] between agents could be implemented.

The application of dynamic structuring in the prototype controller proved a successful and useful characteristic, particularly with respect to the ability of the overall system to adapt to the availability of underground heaters. The authors contend that similar levels of flexibility would be difficult to achieve or impractical in control systems based on ad hoc combinations of conventional controllers. The dynamic structuring comes from the ability to model the refrigeration system as a cooperating multi-agent system and results in a planning mechanism that launches specialized agents to suit operating conditions and operator demand. The main effect of dynamic structuring in the refrigeration example is to allocate resources as needed. In a more complex controller, where the possibility of mutually exclusive agents exists, this can be used to good effect. For example, a controller and fault diagnosis agent would not necessarily need to exist at the same time. The same computational resource can thus be used for both functions. Dynamic structuring has an additional important characteristic. It allows the planning mechanisms to provide partial or "best-guess" solutions. For example, the underground agent might require the cooperation of a surface agent to satisfy a plan. The underground agent provides an intermediate solution while the surface agent is launched, and waits until the surface agent is able to return a solution of its own. The key point is that workable sub-optimal solutions exist until a final solution is agreed to.

In summary, the design and testing of a prototype distributed intelligent controller for a laboratory-scale mine refrigeration system has indicated that the multi-agent approach to control has several interesting characteristics and is worthy of further development for practical, supervisory-level control of large-scale, complex systems. The technique also has the potential to support implementation of advanced features such as conflict resolution, learning, and diagnostic agents.

Acknowledgements
The support of the South African Foundation for Research Development and the University of the Witwatersrand is gratefully acknowledged. The authors would also like to thank the reviewers for their insightful and constructive comments.

References


---

**H∞** Control for Suppressing Stick-Slip in Oil Well Drillstrings

*Continued from page 30*

**Jan J. Kok** received the M.E. and doctor of engineering degrees in mechanical engineering from the Delft University of Technology, the Netherlands, in 1968 and 1978, respectively. From 1968 until 1985 he worked in the Man-Machine Systems Group of the Department of Mechanical Engineering at that university. Since 1985 he has been professor in systems and control in the Department of Mechanical Engineering at the Eindhoven University of Technology, and since 1991 also in the Department of Technical Physics. His interests include (nonlinear) modeling, identification and control of electromechanical systems and its applications.

**Jan van den Steen** studied electrical engineering at the University of Twente, both in the Netherlands. After working in various areas of research at Philips Research and TNO Medical Biological Laboratory, he joined Shell Research, currently called Shell International Exploration and Production, in 1981 to perform research on subsea control systems for oil and gas production. Since 1990 he has been involved in drillstring dynamics research and, in particular, in the active damping of torsional drillstring oscillations.

---

Ian MacLeod received the B.Sc.(Eng) (cum laude) in electrical engineering and the Ph.D. from the University of the Witwatersrand in 1973 and 1983. After several years in the British and South African chemical industries as a control and instrumentation engineer, he joined the University of the Witwatersrand in 1981. Since 1984 he has been Fuchs Professor of Control Engineering in the Department of Electrical Engineering at the University of the Witwatersrand, Johannesburg, South Africa, where he teaches courses on control engineering, industrial process control, and real-time computer control systems and directs research in the fields of distributed computer control systems and engineering applications of artificial intelligence. He is a member of the South African Institute of Electrical Engineers, the South African Institute of Measurement and Control, and the Computer Society of South Africa. He is a registered professional engineer. He has served on the program committees of many international conferences and is an associate editor of the *Transaction of the South African Institute of Electrical Engineers* and an advisory editor for *Engineering Applications of Artificial Intelligence and Control Engineering Practice*.

Alec Stothert received the B.Sc.(Eng) (cum laude) and M.Sc.(Eng) in electrical engineering and the Ph.D. from the University of the Witwatersrand in 1991, 1993, and 1996 respectively. He is presently a research officer in the Department of Electrical Engineering at the University of the Witwatersrand. His interests include distributed computer control, artificial intelligence, and temporal logic.