15th Triennial World Congress, Barcelona, Spain

ON COORDINATING MULTI-AGENT SYSTEMS USING A TIME CONSTRAINED CONTRACT NETWORK PROTOCOL

Carlos Mitidieri [*]
Carlos Eduardo Pereira [**]
Leandro Lisboa Penz [**]

[*] *Informatics Institute/PPGC/UFRGS - Porto Alegre - Brazil*
miti@inf.ufrgs.br
[**] *DELET/UFRGS Porto Alegre - Brazil*
{cpereira,lpenz}@delet.ufrgs.br

Abstract
For many reasons, predictable coordination of autonomous computing entities under time constraints is a demanding research subject. Devising a general framework that is able to cope with the somewhat contradictory requirements such as timeliness and adaptability is a challenge. This paper presents the preliminary results of an ongoing research project dealing with the application of multi-agent systems in real-time applications. The paper focuses on a well known coordination protocol, the contract network protocol, and proposes some extensions to it in order to tackle real-time systems. The developed concepts are illustrated using a case study of a robotic system. *Copyright 2002 IFAC.*

Keywords:
Multi-agent systems, Real-time, Robotics

1. INTRODUCTION

The growth of the so-called information society, materialized on the massive interconnection of heterogeneous networks of computing devices, has become to a point where there is an opportunity to develop the foundations of an entire new class of distributed applications devoted to the safe operation of all sort of apparatuses without ostensive human supervision. For the purpose of building such applications, required attributes are autonomy, pro-activity and spontaneous interaction, as well as the maintenance of strong references to the physical world. Stated purely in these terms, this set of characteristics has been formerly collected in the field of artificial intelligence, through the concepts of agency and multi-agent systems (Sycara, 1998). In spite of being overloaded with numerous definitions, which can be found in the literature (Wooldridge, 1999), the general notion behind the agent metaphor is well understood, therefore it will be used in order to describe the ideas forward in the present text. Likewise, it is apparent that in such computing environments dedicated to control, timeliness and reliability are additional non-functional requirements that must be filled.

The interaction model, which determines how agents exert mutual influence, is a plain concern in multi-agent systems (MAS). Interaction comprises the communication patterns and the problem of coordinating individual activities, in order to achieve a consistent, goal-oriented collective behavior. A discussion on the requirements for autonomy and spontaneity, as well as a comparative analysis of several high level communication models usually applied in the context of object-based systems, can be found in a previous work from the authors (Pereira *et al.*, 2001). In the present paper, the focus is on cooperative coordination issues, having time as one of the prime analysis dimensions. More specifically, the contract net protocol (Smith, 1979) is examined in the context of a case study derived from robotics.

As usual, it is wished to put the joint discussion on coordination and timeliness in terms of some general principles. In (Jennings, 1993), for instance, it is claimed that commitments (promises to undertake a specific course of action) and conventions (means of managing commitments in a dynamic environment) are the foundations of coordination in multi-agent systems, which means that these abstractions would be discernible in any possibly designed coordination scheme. Although these underlying abstractions may hold, it is argued in (Durfee, 2001) that there is no unique coordination strategy able to work well under all circumstances, i.e., any specific strategy can be stressed over a given non-functional dimension to the failure point. For example, timeliness (derived from system dynamics) is a highly stressing dimension for any coordination strategy designed to be applied in control scenarios. Indeed, in face of time constraints, it can be hard to achieve optimal coordination, even when limited solution spaces are considered. Therefore satisficing methods, like flexible computation (Horvitz, 1988), anytime algorithms (Dean and Boddy, 1988) and design-to-time scheduling (Garvey and Lesser, 1993), prevail in computing environments where uncertainties must coexist with time constraints. In section 5.1, it will be argued that a satisficing solution for time constrained execution of the contract net protocol can be constructed by the combination of two independent and concurrent criterions: optimization and fairness.

A bottom-up methodology was adopted in the development of the present work. Accordingly, the provided analysis was derived from the empirical deployment of a multi-agent based heuristic developed earlier to solve the inverse kinematics problem in the context of robotic manipulators with redundant degrees of freedom. Redundant manipulators which have to find out a way through partially unknown environments must have on-line support of sensory data about obstacles (e.g., tactile, ultra-sound, etc.). These data should be integrated in real-time to the path planning process and to the inverse kinematics problem solving. Recalculations of paths and inverse kinematics have a high cost with traditional mathematical methods and would result in significative delays in the operation, if they had to be done on-line. Some alternative solutions (based on the MAS paradigm) to the inverse kinematics problem have been proposed in the past (Beyer and Smieja, 1997; Bohner and Lüppen, 1997). The contribution of the present effort was to add a coordinating protocol for the enforcement of a specified path and analyze the resultant scheme from a real-time point of view. The target platform consisted in a robotic system with anthropomorphic inspiration, the JANUS system, which is described ahead in the text. Considering that the focus of this work is not the inverse kinematics problem itself, but on how time constraints are reflected on the coordination among autonomous objects under CNP, there is no review on the previous research on classical mathematical methods for solving inverse kinematics. However, such a review can be found in a previous article from the authors (Mitidieri *et al.*, 2001).

This paper is organized as follows: the applied heuristic is described in section 2 with references to related work. In section 3 the case study used for evaluating the proposed approach is presented. Sections 4 and 5 describe into more detail the multi-agent approach and the coordination scheme being proposed. Section 6 discusses some preliminary results from simulation. Finally, in section 7 conclusions are drawn and directions of future work are signaled.

2. MAS APPROACHES FOR SOLVING THE INVERSE KINEMATICS PROBLEM

Its is assumed that the system architecture follows the two levels architecture proposed by (Bohner and Lüppen, 1997): the lower level being highly reactive and the other designed for complex planning. The reactive level is modeled as a multi-agent system, with an agent mapped to each joint and its respective sensors, motor and link. The reactive agents locally integrate sensory data and collectively act to reach objectives (e.g., Cartesian coordinates) under constraints (e.g., a certain orientation for a particular link) defined and informed by the planning level. At the reactive level, two criteria are taken into account by the agents for reactively determining the path to be followed by end-effector: the goal (described in sec. 4) and the obstacle avoiding criteria. The interaction among joint agents is cyclic and follows a token passing scheme: a cycle starts with the end-effector agent performing a move in the physical end-effector based on the goal criterion and then informing its new state to all other joints, before passing the token to the next joint in the chain. This then repeats the procedure, until the base joint is reached, which closes the cycle passing the token back to the end-effector agent.

Also in (Beyer and Smieja, 1997) a heuristic multi-agent approach is applied for solving the inverse kinematics problem. In this work, a team of reflexive autonomous agents is deployed for each joint, each agent producing a suggestion for a joint move based in some criterion: (*i*) a *goto* agent, performing the same angle minimization algorithm described in the last paragraph as the goal criterion; (*ii*) a *pain* agent, trying to keep the local control variable within its specified range; (*iii*) a *relax* agent, forcing the local control variable to stay in the vicinity of its equilibrium value; *(iv)* an *orientation* agent, constraining the direction of its respective link. The communication and interaction among agents is accomplished through the Flip-Tick

Architecture (Richter, 1999), which hierarchically organizes agent in teams. The referenced work pointed out that these heuristics were not able to produce a smooth path movement, but since the production of a smooth path (using such heuristics) was not the main goal, the achieved results were considered satisfactory.

These previous works did not addressed the problem of planned path enforcement which are an issue in many applications: assembly and disassembly, arc welding, etc. Therefore, the present work makes a contribution designing and investigating some extensions to the multi-agent heuristics earlier described in order to assess the feasibility of smooth and controllable paths with such approaches, while satisfying real-time constraints.

3. THE JANUS PLATFORM

Multi-agent concepts developed in the present work were experimentally tested using the JANUS robotic system, which is a stationary system with anthropomorphic inspiration. It consists of a vision sub-system and two complex manipulators. The vision sub-system is mounted on a "neck" with two degrees of freedom. To grab and manipulate objects, the system is equipped with two manipulators with eight degrees of freedom each. All joints are monitored by optical encoders which, together with the vision sub-system, form a complex sensor network. Figure 1 shows the control architecture which rises the multi-agent abstraction used in the case study.

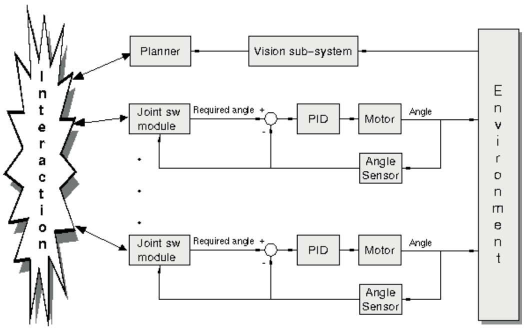


Figure 1. Control scheme.

4. HEURISTIC SOLVING OF INVERSE KINEMATICS PROBLEM

The distributed algorithm for reactive path determination presented in (Bohner and Lüppen, 1997) as the goal criterion and implemented in (Beyer and Smieja, 1997) as the goto agents is further detailed in this section. For a simplified overview, we focus in the two dimensional case. Figure 2 schematically represents a robotic manipulator contained in a plane, with three links and three rotational joints. The algorithm can be generalized to the three dimensional case, as is fully demonstrated in (Bohner and Lüppen, 1997).

On its turn (e.g., when it possess the token, as described in sec. 2), each joint-agent minimizes the angle defined by: (*i*) the vector connecting its associated joint and the end-effector; and (*ii*) the vector connecting its associated joint to the Cartesian coordinates of the goal. Once the minimization is actually accomplished, the new joint angle is communicated to all other joint agents, as long as it is necessary to maintain the arm state updated in order to calculate the angles describing the vectors with respect to a convenient Cartesian global reference.

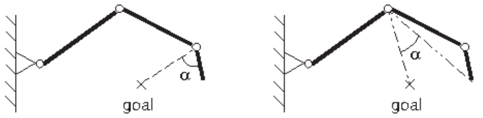


Figure 2. Schematic illustration of the distributed angle minimization algorithm.

One additional parameter to be set is the magnitude of the angle minimization performed at each step of the algorithm at a particular joint agent. One criterion for this setting would be making the step proportional to the size of the α angle (fig. 2). However, when smooth paths are required, it is interesting to keep this steps as small as possible in practice, in order to enforce small displacements of the end-effector.

5. PROPOSED COORDINATION SCHEME

The proposed approach refines the high level interaction among joint-agents by introducing the contract net protocol (CNP) as the coordinating mechanism for the distributed execution of the minimization algorithm described in section 4. Functionally, the main goal is to enforce the covering of a pre-planned path. CNP is a well known and widely applied protocol aimed for dynamic task allocation, i.e., for the on-line identification of the agent which can better service a given task at a given moment. Following conventions are assumed: an agent with a task to be served is called a *manager*, while an agent that might be able to serve a task is a *contractor*. From manager's perspective the protocol has the following steps:

- Task announcement;
- Bids reception from potential contractors;
- Evaluation and awarding of a contract to a suitable contractor;
- Reception and synthesis of results

Complementary, from the contractor perspective, the protocol steps are:

- Reception of tasks announcements;
- Evaluation of self capacity to serve the task;

- Respond: decline or bid;
- Perform the task if the bid is accepted;
- Report results.

In general, subcontracting is allowed, i.e., a contractor can decompose an awarded task and sub-allocate it to other agents. The CNP offers the advantage of graceful degradation, since the population of potential contractors can be made high by design. On the other hand, the manager represents a single point of failure.

The structure of a task announcement includes the fields for address, eligibility, task specification, and expiration time (deadline for receiving bids). Once an announcement is received on the contractor side, it must meet the eligibility criterion to return a bid. The eligibility criterion is an attempt to fairly distribute the effort to reach a given goal among all joints. It consists in constraining the maximum angular deviation from the mean position (i.e., the equidistant position from the end-of-course positions) that a given joint may present to have its bids accepted by the manager. As a consequence of the eligibility criterion, it may happens that during some instances of the algorithm execution, a configuration occurs in which all joints of a manipulator reach the maximum angular displacement, i.e., all joints become ineligible. In this case, the manager does not receive any bid and its procedure then is to send a new announcement with a relaxed value in the eligibility field.

5.1 *Real-time requirements*

The deployment of the CNP as a path constraining complementary technique for the angle minimization heuristic should be followed by the enforcement of the time constraints imposed by this problem. Although the joint-agents are conceptually distributed over a network of processing units (such as microcontroller-based boards) and their computation progress in parallel, the communication on the second phase of the CNP (i.e., bid returns) is serialized on the CNP level, simply because there is only one manager receiving bids.

Therefore, we put the real-time problem related to the communication in the CNP layer in the following terms: if the manager waits for the bids from all possible correct contractors before awarding a contract then, in principle, the highest quality solution can be attained, i.e., the path actually followed by the end-effector of the manipulator will be the closest to the required, with minimum oscillation. Nevertheless, if for some reason, a deadline has to be imposed for bid acceptance on the manager side, then it is possible that some of the contractors in a set would not have enough time to bid in a particular round of the CNP execution, due to the serialization discussed above. In such case, it is clear that a tradeoff must be eventually made between the quality of the solution and the time for the CNP round completion.

In conventional CNP, contractors return their bids in a unknown order. In this a case, if there is a deadline for bids reception, there is no control over which contractors will have success on biding. On the other hand, if an order is statically imposed for returning bids, e.g., by means of a *token* passing scheme or static priority assignment, then a situation could arise on which some joint agents would be potentially avoided to ever have a chance to bid. To avoid this situation, some scheme for dynamic ordering the bids is required. It is clear that such a dynamic scheme can not produce deterministic orders because the ordering algorithm is distributed over the contractors and the consensus would causes an undesirable overhead, which would diminish the efficiency of each protocol round. Moreover, the ordering scheme has to match the following requirements:

(1) *Optimization*: assign higher priorities to contractors with higher probability of producing higher quality bids;
(2) *Fairness*: guarantees that all contractors will eventually have the chance to bid. Even those with little chance of producing a good quality bid (thanks to recent history) should eventually have the chance to bid;
(3) *Timeliness*: priorities have to reflect time constraints expressed as deadlines.

To evaluate their probability of producing good quality bids, the contractors must receive a feedback from the manager, which is the only agent that has information about the optimum action (i.e., the path to be followed). Therefore, a new field should be added to the task announcement structure, in which the manager could annotate the bids ranking of the last protocol round. This information could then be used to feed an utility function that dynamically define an order, as will be discussed in section 5.3.

5.2 *Dynamic bidding ordering*

To fill the requirements stated at the end of last subsection, it is adopted a message scheduling model that is equivalent to the $(m,k)-firm$ task scheduling model (Ramanathan and Hamdaoui, 1995). Following the $(m,k)-firm$ scheme, a periodic message is allowed to miss $k-m$ deadlines in a sliding window of k consecutive transmissions. In other words, a message designed with a $(m,k)-firm$ deadline attains its specified quality of service (QoS) level whenever, in a sliding window of k consecutive transmissions, it satisfies the deadline in m occasions. The system experiments a dynamic failure whenever this condition is not satisfied.

Priorities are dynamically assigned to messages instances as a function of their proximity to a dynamic failure: the closer to the dynamic failure, the higher the priority. The highest priority is mapped to numeral "1", the second highest to numeral "2", and so on. For example, a message instance (bidding) specified as $(2,$ 4)-firm with a current transmission window described by the sequence "BBXX" (where B denotes successful bid transmission and X a deadline transgression, the most recent event in the right side) has a priority of "1" (see table 1), meaning that if it misses the next deadline it will experiment a dynamic failure (Montez *et al.*, 1999).

Table 1. Mapping of priorities.

priority	interpretation
0	failure
1	if misses next deadline, then fails
2	if misses next 2 deadlines, then fails
n	if misses next *n* dealines, then fails

5.3 *Balance of criterions*

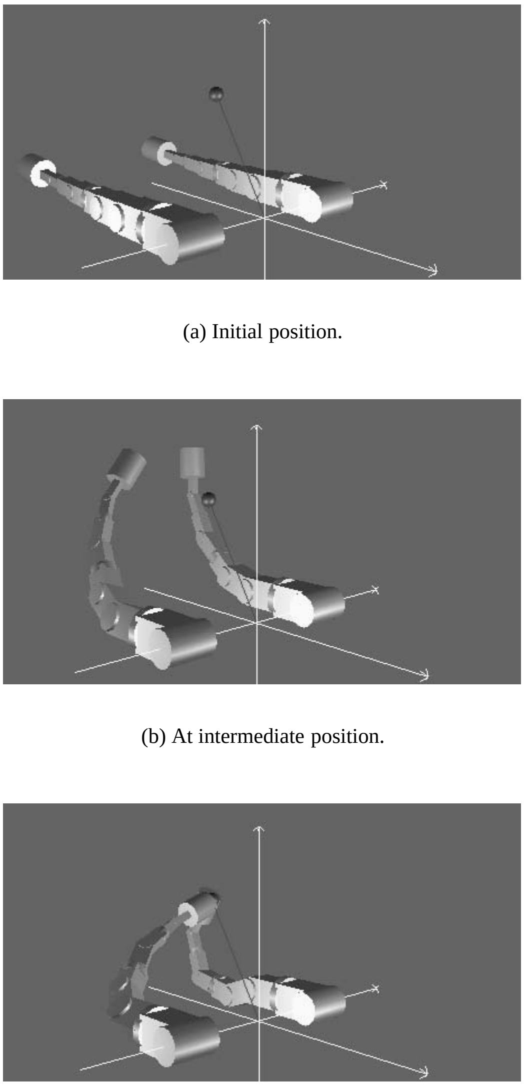
The obvious solution for the scheduling of bids is to centralize this function on the manager side, which has complete information on the bidding from contractors, both in terms of quality and timeliness. In this way, instead of feeding-back just the quality ranking of the last round, the manager would just inform the ordering for returning bids in the next protocol round. It is possible to implement this scheme in the specific viewed case because there is a large number of related protocol rounds for each manipulator movement. Nevertheless, the interest is on distributed solutions that could be generalized to other cases.

The dynamic bidding priority (*Pr*) of joint-agents should reflect the probability of producing a good quality bidding, besides giving to any agent the chance to eventually submit a movement suggestion (sec. 5.1). More specifically, the ordering should be a function of up-to-date information on the quality ranking (*Ra*) provided by the manager and the priority derived from the (m,k)-firm scheduling policy (*Au*). These two quantities are combined through the following heuristic:

$$Pr = \max(Ra, Au)$$

The dynamic priority for bidding is the maximum (priority) between the priorities assigned separately by the criterion of ranking and by the (m,k)-firm policy. In order to facilitate direct comparison, ranking is expressed as natural numbers, with 0 being mapped to the maximum relative priority and so on.

It is clear that nothing prevents more than one contractor from having the same priority for message transmission on a given round of the protocol, since the consensus would impose more difficulties on the scaling of the CNP, with direct impact on the timeliness. Therefore ties must be eventually broken at lower layers of the communication stack. This is ongoing work. Finally, observe that the bidding messages transmitted by each joint-agent should be specified with a deadline of the type (m,k) at the design phase.



(a) Initial position.

(b) At intermediate position.

(c) Reaching Cartesian goal.

Figure 3. 3D simulation of the general scheme.

6. SIMULATION RESULTS

We have simulated the proposed scheme with a 3D visualization model, which was constructed in rigorous conformance to the geometry of the real robot. Figure 3 shows 3 snapshots from the visualization tool: (a) at an arbitrary initial position, (b) at an intermediate position, (c) reaching the Cartesian goal. The Cartesian goal is marked as a sphere linked by a straight line to the origin of the reference system. It is worthwhile to note that the effort to reach the Cartesian goal was fairly distributed among all joints, while the timeliness was guaranteed at the manager side.

7. CONCLUSION AND FUTURE WORK

This paper has focused on the timing analysis of cooperative coordination with the contract network protocol, which was applied in the context of a multi-agent approach to the problem of inverse kinematics in redundant manipulators. The CNP was selected as a starting point for the investigation of time constrained coordination in the context of control environments due to its simplicity. The goal was to devise principles and properties that could be mapped to more complex problems. Accordingly, it was inferred that a satisficing solution for time constrained coordination with CNP could be obtained by simultaneously fulfilling the requirements for optimization and fairness. Future work will concentrate on further development of an event based interaction model, having timeliness and autonomy as the basic requirements, as well as on the integration of present and future results on such model.

Acknowledgments.

This work has been supported by CNPq, a research agency from Brazil. Our thanks also to GMD - National Research Center for Information Technology - Germany, for donation of JANUS.

8. REFERENCES

Beyer, U. and F. Smieja (1997). A heuristic approach to the inverse diffrential kinematics problem. *Journal of Intelligent Robotic Systems* **18**(4), 309–327.

Bohner, P. and R. Lüppen (1997). Redundant manipulator control based on multi-agent systems. In: *Proceedings of the 3th IFAC Symposium on Intelligent Components and Instruments for Control Applications (SICICA'97)*. pp. 357–362.

Dean, T. and M. Boddy (1988). An analysis of time-dependent planning. In: *Seventh Conference on Artificial Intelligence*. St. Paul, Minessota, USA. pp. 49–54.

Durfee, E.H. (2001). Scaling up agent coordination strategies. *IEEE Computer* pp. 39–46.

Garvey, A.J. and V.R. Lesser (1993). Design-to-time real time scheduling. *IEEE Transactions on Systems, Man and Cybernetics* **23**(6), 1491–1502.

Horvitz, E.J. (1988). Reasoning under varying and uncertain resource constraints. In: *Proc. of the Seventh National Conference on Artificial Intelligence*. pp. 111–116.

Jennings, N. (1993). Commitments and conventions: The foundation of coordination in multiagent systems. *The Knowledge Engineering Review* **8**(3), 223–250.

Mitidieri, Carlos, Carlos Eduardo Pereira, Renato Ventura Bayan Henriques and Leandro Lisboa Penz (2001). Coordinating heuristic for path enforcement of robotic manipulators in assembly and disassembly lines. To be published in the IFAC Workshop on Intelligent Assembly and Disassembly 2001.

Montez, C., J. Fraga, R. Oliveira and J-M. Farines (1999). An adaptive scheduling approach in real-time corba. In: *ISORC 99*. IEEE Computer Society Press. Los Alamitos, California.

Pereira, C. E., J. Kaiser, C. Mitidieri, C. Villela and L. B. Becker (2001). On evaluating interaction and communication schemes for automation applications based on real-time distributed objects. In: *Proceedings of the 4th Int. Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'01)*. Magdeburg, Germany.

Ramanathan, P. and M. Hamdaoui (1995). A dynamic priority assignenment technique for streams with (m,k)-firm deadlines. *IEEE Transactions on Computers* **44**(12), 1443–1451.

Richter, Gernot (1999). Flip-tick architecture: A design paradigm for cycle-oriented distributed systems. Technical Report 84. GMD National Research Center for Information Technology.

Smith, R. G. (1979). The contract net protocol: High-level communication and control in a distributed problem solver. In: *Proceedings of the 1st International Conference on Distributed Computing Systems*. IEEE Computer Society. Washington, DC. pp. 186–192.

Sycara, Katia P. (1998). Multiagent systems. *AI Magazine* pp. 79–92.

Wooldridge, Michael (1999). Intelligent agents. In: *Multiagent Systems: A modern approach to distributed artificial intelligence* (Gerhard Weiss, Ed.). Chap. 1, pp. 28–77. The MIT Press.