COOPERATION AMONG DISTRIBUTED CONTROLLED ROBOTS BY LOCAL INTERACTION PROTOCOLS

TIM C. LUETH
FZI Research Center, Haid-und-Neu-Str. 10-14, D-76131 Karlsruhe, Germany, email: t.lueth@ieee.org

RONALD GRASMAN, THOMAS LAENGLE
IPR Institute for Real-Time Computer Systems and Robotics, U. of Karlsruhe, D-76128 Karlsruhe, Germany, email: laengle@ira.uka.de

JING WANG
College of Engineering, U. of California, Riverside, CA 92521, U.S.A

ABSTRACT

Inter-robot cooperation is useful on the symbolic and physical levels. It requires the dynamic coupling of planning systems but also the coupling of real-time control systems. This paper describes the basic cooperation mechanisms. Furthermore, the use of a cooperative robot operation system CAIC and the FZI's local infrared communication system for the Khepera robots during inter-robot cooperation is described.

KEYWORDS: Inter-robot cooperation, coupled cooperative robot control, local communication

COOPERATION DURING INDEPENDENT TASK EXECUTION

Typically, robots are designed in a first step without considering multi robot environments. These robots receive their tasks from one or more *clients* and operate as a *server*. During task execution the robots elaborate the *symbolic task description* depending on the current environment, *map* the elaborated task to a *network of control loops* [1], and *physically change* the environment. For instance, they transport or manipulate assembly parts. This means, besides the symbolic task description there is an actual physical robot-environment-interaction.

If several mobile robots are active as servers in the same environment and receive independent tasks from the same clients, they physically have to share resources, i.e., at least their work space. To master arising conflicts, passive or active interaction levels can be chosen. For each interaction level, a different execution-control architecture is required. The levels can roughly be classified in:

1 The individual robot classifies an other robot as an *uncooperative* robot. It delays, accelerates, or adapts its own task execution (i.e., individual conflict compensation).
2 The individual robot classifies an other robot as an equal environment *participant* that obeys a common set of passive physical interaction rules such as using traffic lanes or access rules (CSMA/CA/CD). Both robots minimize the conflict potential together [2].
3 The individual robot classifies an other robot as an equal environment *participant*. It expects *broadcasts* of the participant's resource-requirement schedule automatically or

Lueth, T.; R. Grasman, Th. Laengle, J. Wang (1996): Cooperation Among Distributed Controlled Robots by Local Interaction Protocols. Invited paper at ISRAM Int'l. Symp. on Robotics and Manufacturing, Montpellier, France, May, Robotics and Manufacturing, Vol. 6, ASME Press, 1996, pp. 405-410

on request. The exchange of symbolic description avoids misinterpretation by individual estimation of other participants' future resource-requirements. The information exchange requires communication capability and a protocol [3].

4 The individual robot classifies an other robot as an environment *partner*. Both robots exchange their resource schedules and optimize the joint schedules based on individual (negotiation) or common needs [4].

In all four *interaction levels*, the individual tasks are performed individually and independently even if communication, coordination, and symbolic cooperation are used for optimization. Each robot has to classify an unknown robot as *uncooperative*, *participant*, *communicating participant*, or *partner*. It has to select the required interaction protocol for optimal individual behavior. Optimal means overall-cost efficiency for the individual robot. Overall-costs depend on delays in physically task execution but also on delays by communication, negotiation and planning and in addition on the availability of communication and information processing capacity. Therefor, an individual robot can decide not to behave like a partner even if it could so, since from its point of view it's inefficient. Further considerations belong to *cooperative game theory*.

The number of possible conflict relations among two individual robots is proportional to the square of the number of robots. If the number of robots is increasing, the time between two conflict situations will decrease correspondingly. In a similar manner, the available time to master a conflict situation is decreasing and the need for efficient conflict avoidance becomes evident. The need of an individual robot to behave like a partner will automatically arise.

Cooperation during independent task execution is caused by resource conflicts and requires coordination (delay, acceleration) of alternating execution. Therefore, the following steps must be performed in each robot:

1 Decision on the maximal offered individual interaction level.
2 Negotiation (Offering, accepting, refusing, ignoring) on the common interaction level.
3 Using the common understood interaction level.

In contrast to that, cooperation by coupled distributed controlled system is caused by the need for joint physical environment changes [5]. It requires simultaneously and timely executions.

COOPERATION BY COUPLED DISTRIBUTED CONTROL SYSTEMS

If a task cannot be physically performed by only one robot it must be given to a team of robots that are able to perform the task by physical cooperation. Consequently, the symbolic task description must be elaborated in terms of the team's configuration and mapped to a network of control loops. Then, the network is separated into smaller control sub-networks in terms of the available capacity (robots). Afterwards, the *subnets* are distributed to the individual robot-control systems [6].

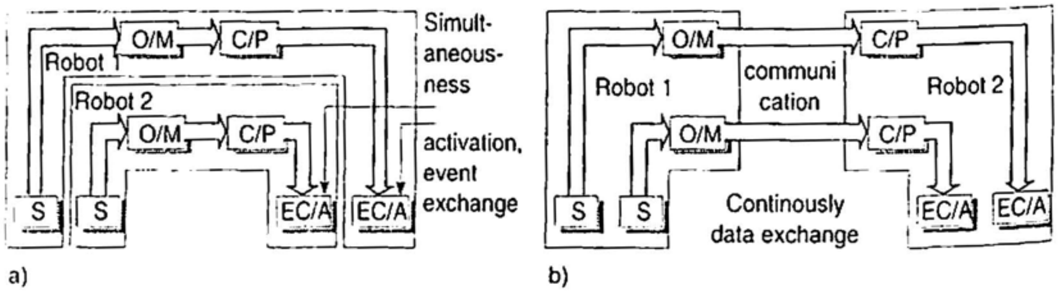

Figure 1. a) Coupled activation of control networks. b) Coupling within control networks

Often, closed subnets of the network are distributable completely to one robot (Fig. 1a). Then, there is only a need for *event-based synchronizing the activation* of local subnets with subnets running on other robots.

Sometimes, it is necessary to distribute even single closed-control loops to different robots by *separating* the control loops into linked loop modules for *sensing* (S), *observation/modeling* (O/M), *control/planning* (C/P), and *execution control* (EC/A) (Fig. 1b). Therefore, *continuous information flow* (*time-based* or *on request*) must be established between two robots and also the module activation must be synchronized [7].

LOCAL COMMUNICATION FOR COOPERATIVE ROBOTS

Dynamic distribution of control-network modules and dynamic distribution of events and control information between the modules require a *robot operating systems*. For this purpose, the *CAIC* system (*Cooperative Architecture for Intelligent Control*) has been developed in Karlsruhe [7,8]. CAIC is an extension for existing real-time operating systems. It allows the flexible linkage of encapsulated control modules by supporting different module activation (time-based, event-based) and information exchange (continuously, on request, on event). Furthermore, it allows to initiate the execution of control modules on other robots.

To support robot operating systems such as CAIC and to allow the dynamic coupling of real-time control loops operating on different robots, a local communication system is required. It should have the following features:

- No central communication backbone even if wireless (Fig. 2a),
- real-time data exchange, i.e., little and known transfer delay,
- selection of communication partners based on physical conditions (Fig. 2b), and
- limited channels width to avoid disturbance by other teams (Fig. 2c).

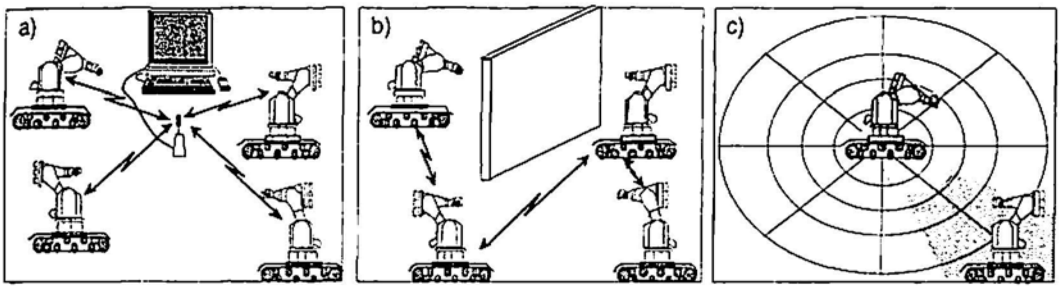

Figure 2. a) Central backbone. b) Visible communication partner selection. c) Limited consumption of the communication medium.

THE ROBOTS AND THEIR COMMUNICATION SYSTEM

For our experiments, small mobile manipulators (100 g, Ø 55 mm) of the Khepera family [9] have been used. For mobility, two servo motors equipped with encoders and gearboxes (25:1) are used. The incremental encoder resolution is 12 tics/mm. For each motor a PID control loop is used for controlling the speed. The PID-coefficients are programmable and the internal counters (I-term) are readable.

Eight infrared proximity sensors (sender and receiver) are able to measure back light with a resolution of 10 Bit. The sender emits light with a frequency of 2 or 10 Hz depending on the BIOS version. Therefore, the sensors are able to measure distances of the robot to other objects in the range between 10 and 60 mm depending on the objects' reflection.

The robot is controlled by a 32 Bit micro controller (MC68331) with 256 K ROM and 256 K RAM running with 16 MHz. The BIOS supports a simple multi-tasking operating system for 15 concurrent tasks. Task switching is performed by a 5 ms round-robin mechanism. After a reset, the BIOS boots the system through an optional serial line with 9.6 or 38.4 KBit/s. A serial line can be used for power supply and communication with a work station too.

The gripper turret on top of the robot's basis turret is equipped with a 8 Bit micro controller (68HC11) and an EPROM for the BIOS extensions. The turrets are connected physically and electrically via a bus connector system. The gripper has two DoF: jaw opening width and gripper angle. Both are controlled by stepper motors. A photosensitive sensor can detect objects between the gripper jaws.

To support local robot interaction without a central communication workstation/server an infrared communication (IRC) turret (Fig. 3) has been developed at FZI/IPR [10]. The IRC-turret is mounted on top of the robot's I/O-turret.

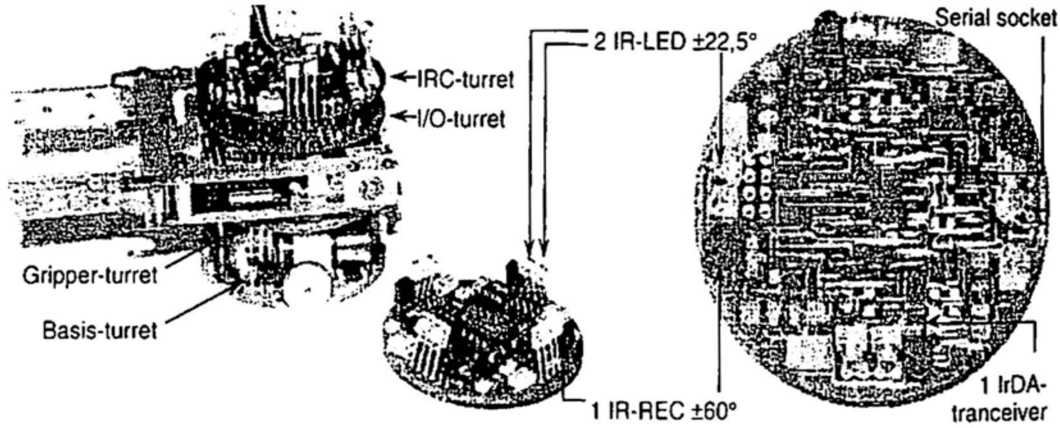

Figure 3. The FZI's inter-robot infrared communication turret for the Khepera robots

The IRC-turret for inter-robot communication switches the serial port (UART) of the MC68331 either to the serial cable connector of the turret or to one of four independent infrared communication transceiver modules of the turret. Each module is based on the IrDA standard [11, 12] (half-duplex) and uses two Ir-senders for emitting the information with 9.6 KBit/s within an approximately 90° degree area. The Ir-receiver of each module is able to receive of an approximately 120° degree area (Fig. 4a).

One feature of the FZI's IRC-turret is the programmable sender resistor and receiver sensitivity. By changing the sender's resistor, the maximal communication range (100% reliable communication) can be increased up to 100 cm. By changing the receiver's sensitivity (5-Bit), the communication range can be modified as shown in Fig. 4b.

To broadcast a message into all directions, it is necessary to sequentially select the four modules and send the message. Since the communication rate is only 9.6 KBit/s, the micro controller is able to send a Byte in all directions in less than 1 ms. On the other hand, it is not possible to receive concurrently data from all four module by switching. Nevertheless, by a special circuitry it is possible to detect, which modules are receiving data. Then, the program can select which module will be used for listening.

To support the use of the IRC turret, a set of real-time communication procedures and processes (tasks for the multi-tasking BIOS) are available. The communication procedures are used for establishing [13] basic exchange of events (signal/wait), for synchronization, and for continuously data (processed sensor information, execution commands) exchange. For each 90°-sender module, a cycle send-buffer exists that is sent and emptied with a defined frequency or on demand. For the selected 120°-receiver module, a cycle receive-buffer is filled with a frequency of 1 ms. All control modules/tasks can write messages into the send-buffer. The content of the receive-buffer is interpreted independently by several control modules/tasks that wait for external information.

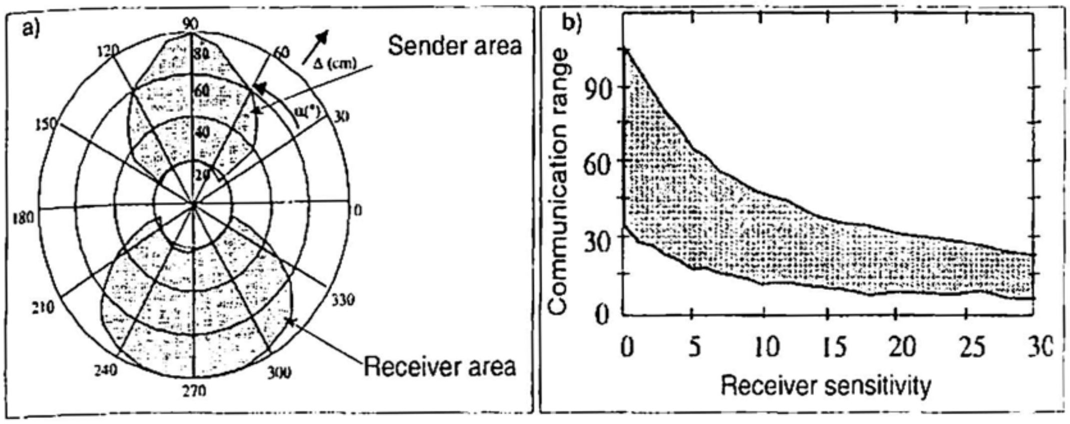

Figure 4. a) Sender and receiver range field for 100% reliable communication. b) Reliable communication range in terms of 5 Bit receiver sensitivity [10]

INTERACTION PROTOCOLS AND EXPERIMENTS

Using the IRC-turret and the CAIC operating system, inter-robot cooperation is achieved by the following strategy. In CAIC, the *control-state* of a control-subnet describes which control loops are active simultaneously. If two robots want to synchronize their states, they exchange their readiness for changing to a new state: If a robot itself is ready for a change and interprets an external request for a change, it broadcasts that it will now change its state. Afterwards, both robots switch to the new state in which, typically, a different interpreter and communication protocol are used for processing the received information. In Fig. 5. an example is shown for using this method in transporting an object in a closed kinematic chain. Doing this in a master-slave configuration requires the continuous transfer of data from the master to the slave.

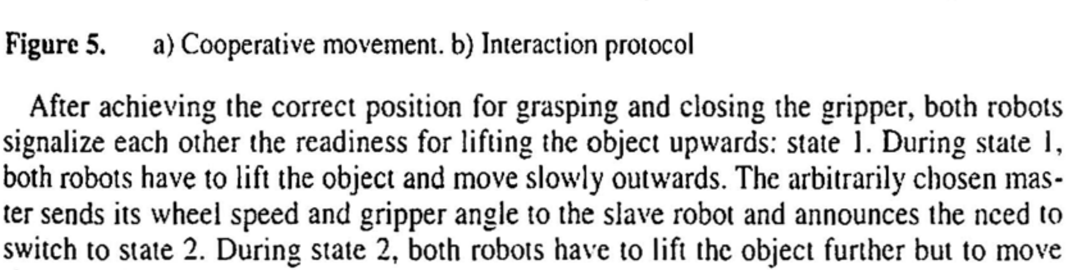

Figure 5. a) Cooperative movement. b) Interaction protocol

After achieving the correct position for grasping and closing the gripper, both robots signalize each other the readiness for lifting the object upwards: state 1. During state 1, both robots have to lift the object and move slowly outwards. The arbitrarily chosen master sends its wheel speed and gripper angle to the slave robot and announces the need to switch to state 2. During state 2, both robots have to lift the object further but to move slowly inside. Again, an arbitrarily chosen master sends its wheel speed and gripper angle to the slave robot and announces the need to switch to state 3. During state 3, both robots move in the same direction to transport the object.

Previous experiments [6] have shown that an individual robot can be used even for assembly operations (grasp, transport, insert) of the *Cranfield-Assembly-Benchmark.* By using the CAIC system, the IRC-turret, and the interaction protocols, two robots are able to grasp and transport cooperatively big and heavy parts. Ongoing research deals with cooperative execution of assembly tasks such as insert by two robots.

CONCLUSION

Cooperation among distributed controlled robots is more than plan synchronization, coordination, and cooperative planning. For physical cooperation, a common control network must be distributed to several robots that communicate to synchronize the activation of control modules and to transfer control loop information between the robots' control systems. To achieve a dynamic coupling of robot control systems, a cooperative robot control system (for instance CAIC [7,8]) must be supported by an adequate local communication system. The FZI's IRC-turret for the Khepera robot is such a system. Interaction protocols that are grounded inside encapsulated control modules are the basis for dynamically coupled control systems.

ACKNOWLEDGMENT

This research is being performed at FZI • Forschungszentrum Informatik, Division TE&R • Robotics, and at IPR • Institute for Real-Time Computer Systems and Robotics (Prof. Dr.-Ing. U. Rembold, Prof. Dr.-Ing. R. Dillmann), University of Karlsruhe. The project is being partially funded by the NATO Project CRG 950340 of Prof. J. Wang and Dr. T. Lueth. Thanks to Prof. Alex Zelinsky (U. of Wollongong, Australia) for his support and Johan Hellqvist (Univ. of Lund LTH, Sweden) for his work.

REFERENCES

1. Lueth. T.C., U. Rembold, T. Ogasawara: Task Specification, Scheduling, Execution, and Monitoring for Centralized and Decentralized Intelligent Robots. R&A IEEE Int. Conf. on Robotics and Automation. Nagoya, Japan, May, 1995, WP2 Task Driven Intelligent Robotics.
2. Parker, L.: Heterogeneous Multi-Robot Cooperation, Doctoral dissertation, MIT AI Lab technical report 1465, MIT Press, 1994.
3. Noreils, F.R.: Coordinated Protocols: An Appraoch to Formalize Coodination Between Mobile Robots. IROS Int'l. Conf. Intelligent Robots and Systems, Raleigh, NC, July, 1992, pp. 717-724.
4. Le Pape, C.: A Combination of Centralized and Distributed Methods for Multi-Agent Planning and Scheduling. IEEE Int'l. Conf. Robotics and Automation, Cincinatti, U.S.A, May, 1990, pp. 488-493.
5. Zimmermann, M.: Concurrent Behavior Control - A System's Thinking Approach to Intelligent Behavior. Swiss Federal Institute of Technology, Dissertation, 1993.
6. Lueth, T.; Th. Laengle, J. Heinzman: Dynamic Task Mapping for Real-Time Controller of Distributed Cooperative Robot Systems. IFAC WS Distributed Computer Control Systems, Toulouse-Blagnac, France, September, 1995, pp. 37-42.
7. Lüth, T.; Th. Längle, J. Hellqvist (1995): Distributed Robot Control by Structural Adaptive Control (in German). In Dillmann, Rembold, Lüth (Ed.), Autonome Mobile Systeme , Springer, pp. 250-258.
8. Hellqvist, J.: Design and Implementation of Real-Time Robot Operation System (in German), University of Karlsruhe, Master Thesis, Faculty of Informatics, February, 1996.
9. K-Team: Khepera User Manual, Ver. 3.0, LAMI-EPFL, Lausanne, Swiss, 1994.
10. Grasman, R.: Local Communication and Coupled Control for Small Mobile Manipulators (in German), University of Karlsruhe, Master Thesis, Faculty for Electrical Engineering, February, 1996.
11. IrDA: Press Release, IrDA Walnut Creek, CA, USA, 1995.
12. TEMIC: Design Guide: IrDA-Compatible Transmission, Telefunken Semiconductors Issue 05.95
13. Premvuti, S.; J. Wang: A Medium Access Protocol (CSMA/CD-W) Supporting Wireless Inter-Robot Communication. In Distributed Autonomous Robot Systems, Springer-Verlag, 1994.