In Proceedings of 4th International Symposium on Distributed Autonomous Robotic Systems, 1998, Karlsruhe

Development of a Robot Agent for Interactive Assembly

Jainwei Zhang, Yorck von Collani and Alois Knoll[*]

[*]*Technical Computer Science, Faculty of Technology, University of Bielefeld, 33501 Bielefeld, Germany*

Abstract. The development of a robotic system interacting with a human instructor requires not only highly-skilled sensorimotor coordination and action planning but also the capability of communicating with a human being in a natural way. A typical application of such a system is interactive assembly. A human communicator sharing a view of the assembly scenario with the robot instructs the latter by speaking to it in the same way that he would communicate with a human partner. His instructions can be under-specified, incomplete and/or context-dependent. After introducing the general purpose of our project, we present the hardware and software components of a robot agent necessary for interactive assembly tasks. The architecture of the robot agent is discussed. We then describe the functionalities of the cognition, scheduling and execution levels. The implementation of a learning methodology for a general sensor/actor system is also introduced.

Key Words. human-robot interface, cognition architecture, sensor-based control, skill learning, multiple sensor/actor systems

1 Introduction

In the Technical Computer Science research group of the University of Bielefeld, a two-arm robotic system is being developed. The general goal of this system is to model and realise human sensorimotor skills for performing manipulation and assembly tasks. This requires a comprehensive set of actuators and sensors, which perform functions similar to those of the human arms, hands and perception channels (i.e. vision, touch, acoustics). To organise the interaction of the complex sensor and control subsystems, sensor data cannot be acquired and processed independently of the movements of the actuators. It is mandatory that both be performed simultaneously and in view of the task the actuators will work on. This is the reason, therefore, that the three major components of an intelligent robot system, i.e. perception, planning and control, are considered synergetically instead of separately. In this way complex, cooperative behaviours involving sensors and actuators can be realised.

The development of our robotic system is closely linked to the on-going interdisciplinary research program of the project SFB[1] 360 "Situated Artificial Communicators" at the University of Bielefeld. The SFB 360 is aimed at the discovery of linguistic and cognitive characteristics of human intelligence for communication purposes. The results of the project are to be transferred to several application domains, one of which is the emulation of human cognitive principles for information processing systems, (SFB360, 1993). The primary example for demonstrating the usefulness of these newly developed techniques is the robot system mentioned above, whose numerous sensor and actuator modules can be used as an ideal testbed for investigating the interaction between human "natural" communicators and machine systems in the real-world. Furthermore it will be used for validating the complete concept by integrating different linguistic and cognitive components. As a basic scenario, the assembly procedure of a toy aircraft (constructed with "Baufix" parts, see Fig. 1) was selected. A number of separate parts must be recognised, manipulated and built together to construct the model aircraft. Within the framework of the SFB, in each of these steps, a human communicator instructs the robot, which implies that the interaction between them plays an important role in the whole process.

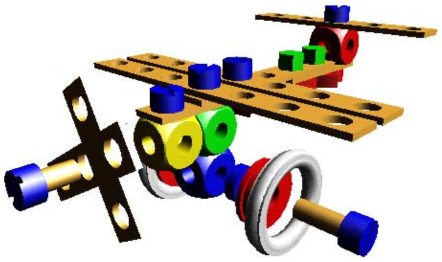


Figure 1 The assembly of a toy aircraft.

The aim of our work is to automate the process of multi-sensor supported assembly by gradually enabling the robot/sensor system to carry out the individual steps in a more and more autonomous fashion. A fully automatic assembly, however, presupposes a precise task description; unfortunately, not much work has been done in this potentially very fruitful area of robotics research. While simulated robot agents are becoming a popular research theme, e.g. (Thorissen, 1997), few work on communicative agents realised with real robots has been reported. One of the recent project was performed with the KAMRO system (Laengle *et al.*, 1996), in which natural language interface was used as a "front-end" of an autonomous robot.

---

[1] Collaborative research effort funded by the Deutsche Forschungsgemeinschaft.

2 System Overview

Our system has been designed to fulfill the demands of flexible “fixture-less” assembly. Its hardware configuration enables a high-speed of the (partly massive) data flows inside the system and the possibility for adding further actuator and sensor components. The structure of the software was so chosen as to ensure the compatibility of different program modules to make the whole system work without collisions and deadlocks. Robot control architecture combines the functional modules of perception, planning, control and the human interface and cooperate both hardware and software components on each abstraction level.

2.1 Hardware Configuration

The physical set-up of our system consists of the following components:

Main actuators: Two 6 d.o.f. PUMA-260 manipulators are installed overhead on a stationary assembly cell, possessing common work space which is similar to the two-arm configuration used in the KAMRO system. On each wrist of the manipulator, a pneumatic jaw-gripper with integrated force/torque sensor and “self-viewing” hand-eye system is mounted.

Computer system: A multi-computer system consists of robot controllers, UNIX-workstations and PCs for data acquisition, data processing and actuator control.

Sensors:

- Two 6 d.o.f. commercial force-torque sensors are installed on the robots’ wrists. They are used for detecting contact, force control as well as two-arm coordination.
- Two miniature colour cameras, each of which is mounted next to the robot gripper. Their function is to perceive local information for fine-manipulation, like grasping, searching a hole, etc.
- Laser sensors are also mounted on the robot gripper. They will be applied to explore the three-dimensional information of overlapping assembly parts as well as assembled aggregates.
- Multiple cameras, some of them articulated, are installed around the assembly table. Their tasks are to build 2D/3D world models, to supervise gross motion of the robot as well as tract the face and hand of the human instructor.
- A microphone-matrix will be integrated to help to determine the speaker location.

Other actuators:

- A small mobile gripper system “Khepera” serves as a transportation agent in the assembly area in cooperation with two robot manipulators.

- In the future, additional robot arms will be installed around the assembly area, carrying the compact cameras to supervise the assembly process actively or be used for extending the working space of the robot system.

2.2 Software Organisation

On the lowest level, the main actuators of the our system are controlled by Multi-RCCL/RCI (*Robot Control C Library/Real-time Control Interface*), see (Lloyd and Hayward, 1989). With this library, the two robots can be synchronised and can run in interpolation cycles as short as 10 ms. The high-speed communication between the sensor systems and the robot task-level control is realised using parallel buses.

Motions of two robot manipulators are controlled by the main control program, which runs on one UNIX-workstation. Sensor processing, control of other actuators and peripherals, and simulation programs communicate with the main control program through sockets or serial interface connection. The generated motion steps of the two manipulators are sent to the “trajectory generator”, which computes the exact joint values for each control cycle. Through the bus adaptor, joint data are further transferred to the joint controllers of the two PUMA-robots.

The control of the robot is divided in two parts. The first part is the real-time control of the robot, which is distributed on several computers. The other part is the programming environment, which includes the communication with the robots and the user interface. Additionally, it also contains the basic motion skills of the robots. We implement a framework called OPERA (*Open Environment for Robot Applications*) to create software for the robot agent.

To program an assembly sequence multiple modules are needed. These modules are loaded dynamically from the environment and include different complex parameterised movements. In our case these are the operations and basic primitives for the assembly scenario.

3 Control Architecture

In order to achieve the main objective described in section 1, the designed system adopts the interactive hierarchical architecture according to Fig. 2. A *Human Communicator* (HC) is closely involved in the whole assembly process.

3.1 Cognition Level

Our robot agent integrates the research results of linguistic description, intention detection, speech recognition, etc. from different partners within the SFB. The robot system should be able to understand not only the simple verbal instructions, but also to detect the context related ambiguity with profound linguistic background.

The system and the HC interact with natural speech (and in the near future with hand-gestures). Human instructions that are in nature mostly situated, ambiguous, sometimes incomplete will be recognised. An example is the command: *Grasp the left screw*. The system has to identify the operation and the object for this operation.

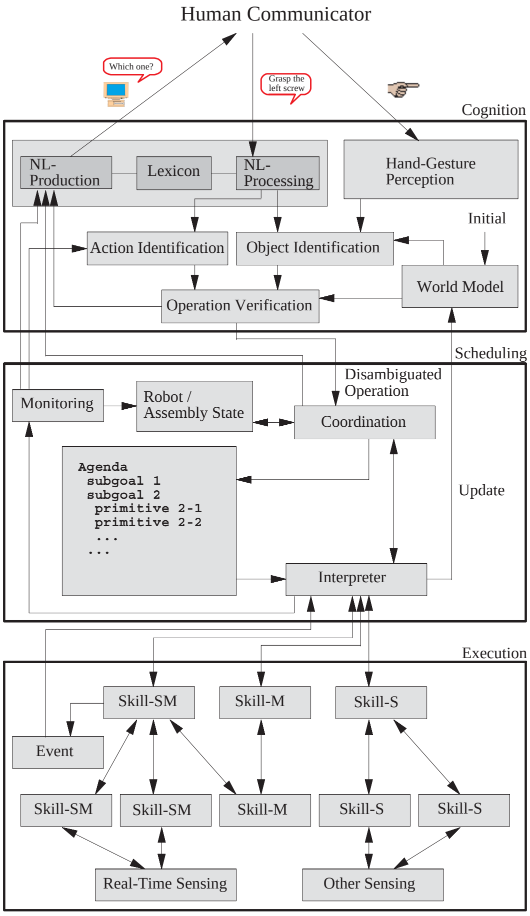


Figure 2 Control architecture of the communicative robot agent.

With the help of a hand gesture the operator can further disambiguate the object. The system may then use the geometric knowledge of the world to identify the right object.

The chosen operation is then verified to check if the intended instruction can be substantiated. If in doubt, the robot agent should ask for further specification or the right to pick an object by itself. Once the proper operation is determined, it is is given to the *coordination* module on the next level.

3.2 Scheduling Level

On the scheduling level, an assembly task of the toy aircraft or an aggregate of which is decomposed into a sequence of primitive robot operations. The final decision about the motion sequence depends on the instructions of the human user as well as the generated plan. The *coordination* module should not only be able to understand the human instructions, but also to learn from the human guidance and improve its planning abilities gradually.

The *coordination* module on the scheduling level receives a Disambiguated Operation (DOs) from the cognition level. By referencing the *robot/assembly state*, the *coordination* module chooses the corresponding basic primitive sequence for the operation. This sequence is a script of basic primitives, which is the implementation of the chosen operation. The sequence includes planning of the necessary trajectories, choosing the right robot or robots, higher level actions and also provides basic exception handling. The screw operation, for example, is based on the following (simplified) script:

1. Find contact between screw and nut.
2. Find the thread and insert the screw.
3. Find the notch point.
4. Screw in.

Other sample primitives are: *Find the hole in a ledge*; *Adjust hand*; *Close hand*; *Get location*; *Move to location*.

Sequences are executed by the *interpreter*, which activates different skills on the next level, the *execution*. The *interpreter* also receives an event report that is generated by the *execution* level. If the event is a failure detection, the *interpreter* has to handle this exception and to inform the *monitoring* module. The *monitoring* module updates the robot/assembly state. If it is found out that the robot agent can re-perform the operation, the *coordination* module will give it another try. Otherwise, the *monitoring* module asks the human communicator how to handle the exception and waits for an instruction. After the execution of each operation, the *world model* is updated.

If the operation is a simple intervention instruction such as “halt!”, it is directly forwarded to the *interpreter* and activates the corresponding motion command.

3.3 Execution Level

The *interpreter* in the coordination level uses the assembly skills from the execution level to perform a sequence. In our approach a skill is not an autonomous agent which can intelligently intervene in the scenario. Instead, it is a more powerful command than a simple movement of the robot. More powerful skills are composed of one or more basic skills. We classify three different kinds of skills:

Motoric skills: Motoric skills are single robot movements, which are provided by most commercial robots. Some examples are: *Open and Close gripper; Drive joint to; Drive arm to; Rotate gripper; Move arm in approach direction; Move camera.*

Sensor skills: A sensor skill takes one or more sensors and generates a usable information for the scheduling level or other sensormotori skills. These skills are divided into two groups: skills with real-time sensing (force control and visual servoing) and skills that use sensors in a non real-time environment. Typical sensor skills are: *Get joint; Get position in world; Get force in approach direction; Get torques; Check if a specific position is reachable; Take a camera picture; Detect object; Detect moving robot; Track an object.*

Sensorimotor skills: A sensorimotor skill is an encapsulation of sensing (processing of sensor feedback) and action (trajectories). The main types of force-sensor based skills can be: *Force-guarded motion; Force-supervised contact finding; Force-controlled rotation; Force-balanced two-arm carrying; Maintaining force along a motion on a surface.*

Vision-based motion skills are: *Vision-guided gross movement to a goal position; Visual servoing of the gripper to optimal grasping position; Appearance-based fine positioning;* etc.

Events: The skills on the highest level can also signal an event to the coordination level. These events can be for example: *A force exceeds a defined threshold; A camera detects no object; Singularity; Collision;* etc.

4 Learning Assembly Skills

We define a set of DOs (Disambiguated Operations) for an assembly task: *Grasp a part; Place a part; Grasp an aggregate; Place an aggregate; Put a part in; Put a part on; Screw; Align a part* (see Fig. 3). The robustness of these operations mainly depends on the quality of the different skills.

4.1 Learning by Practising – Approach for Acquiring Skills

Up to now, several sensor-based skills have been acquired with an automatic learning method. We view the problem of skill learning as finding an optimal mapping function from sensor pattern to robot motion. Since in most cases such a direct mapping function is non-linear, we adopt an adaptive B-spline model for the learning (Zhang and Knoll, 1998*b*). For vision-guided fine-motion, the appearance-based approach by using dimension reduction with PCA (principal component analysis) was proposed in (Zhang and Knoll, 1998*a*).

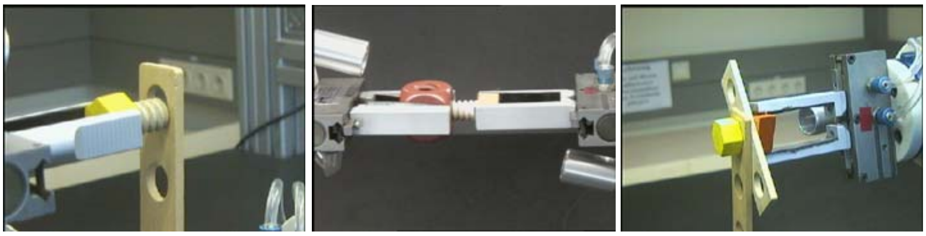
4.2 Grasping

To grasp an object at an arbitrary position and orientation, the main sensor data come from a vision system. The important vision-based skills are multicamera-guided gross motion (Scheering and Kersting, 1998) and optimal grasping using a hand-eye system (Meinicke and Zhang, 1996).

(a) Grasping a screw

(b) Placing an aggregate

(c) Grasping an aggregate with two robots



(d) Put a part in

(e) Screw

(f) Align

Figure 3 Examples of Disambiguated Operations.

4.3 Finding a Hole

A frequent operation for connecting parts is to peg a component onto a screw. This simple operation may fail, if the exact position of the hole is unknown. Fig. 3(d) shows a typical situation. To find the hole, we developed an approach using visual learning (Von Collani *et al.*, 1997). Fig. 4 show the visually guided position correction with the learned controller.

4.4 Screwing

Screwing with two arms is a frequently used operation in our assembly scenario. In order to realise the skills for screwing under diverse uncertainties, we proposed an on-line reinforcement learning method (Zhang *et al.*, 1997*b*). After repeatedly practising in the real world for a specified skill, a controller can find the optimal compliance parameter by itself.

4.5 Assembly of an “Elevator Control”

The assembly of the first aggregate, an “elevator control” of the Baufix toy aircraft, has been successfully performed under the a small subset natural-language

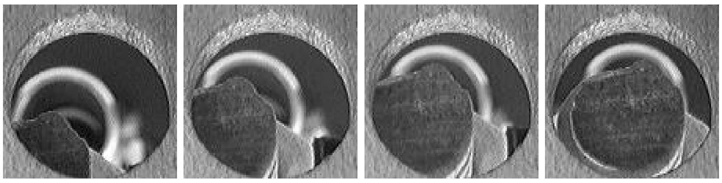

(a) (b) (c) (d)

Figure 4 Correction sequence by using a hand-camera guided motion.

instructions, (Knoll *et al.*, 1997; Zhang *et al.*, 1997*a*). In the current system, natural language instructions are given only by textual inputs.

5 Future Work

Future work on the development of the communicative situated robot agent can be summarised as follows:

Multisensor fusion: The complex manipulation tasks cannot succeed until multiple sensors are applied simultaneously. The mono-camera based technique for grasping separate parts would not be adequate for grasping an aggregate with geometric variances of the parts. We are developing a method for coding complex sensor patterns, (camera plus laser, for example) into an efficient form to be utilised for determining the correct grasping posture.

Learning assembly sequence: In the current implementation, no planning module has been integrated yet. In the future, the robot agent should not always need to be told the explicit assembly sequence. We will investigate how the robot can learn from the assembly sequences it has carried out and plan some subgoals by itself.

Coordination of multiple sensor/actor components: Concepts and control architectures are to be developed for delegating distributed sensor and actuator agents to fulfill common tasks. Modularity, flexibility and optimality are the main criteria to evaluate different approaches.

Comprehensive human-robot communication: Real understanding of natural continuous speech of human involves various aspects of psycholinguistics, dynamic naming, dynamic knowledge representation, etc. The results of the parallel research work in the framework of the SFB 360 will be further integrated into our architecture. More comprehensive situated dialogues will be performed between our robot agent and the human communicator.

6 REFERENCES

Knoll, A., B. Hildebrandt and J. Zhang (1997). Instructing cooperating assembly robots through situated dialogues in natural language. In: *Proceedings of the IEEE International Conference on Robotics and Automation*.

Laengle, Th., T.C. Lueth, E. Stopp and G. Herzog (1996). Natural language access to intelligent assembly robots: Explaining automatic error recovery. *Artificial Intelligence: Methodology, Systems, Applications*.

Lloyd, J. and V. Hayward (1989). Multi-RCCL Users' Guide. Technical report. McGill Research Center for Intelligent Machines, McGill University.

Meinicke, P. and J. Zhang (1996). Calibration of a "self-viewing" eye-in-hand configuration. In: *Proceedings of the IMACS Multiconference on Computational Engineering in Systems Applications*.

Scheering, C. and B. Kersting (1998). Using distributed sensing and sensor fusion for uncalibrated visual manipulator guidance. In: *Proceedings of the 4th International Symposium on Distributed Autonomous Robotic Systems*.

SFB360 (1993). Sonderforschungsbereich Situierte Künstliche Kommunikatoren - Finanzierungsantrag. Universität Bielefeld.

Thorissen, K. R. (1997). Communicative Humanoids - A Computational Model of Psychosocial Dialogue Skills. PhD thesis. MIT Media Lab.

Von Collani, Y., J. Zhang and A. Knoll (1997). A neuo-fuzzy solution for fine-motion control based on vision and force sensors. Technical report. Department of Technology, University of Bielefeld.

Zhang, J., A. Knoll, B. Jung, I. Wachsmuth and G. Rickheit (1997*a*). Experiments of robotic assembly instructed by situated natural language. In: *Video Proceedings of IEEE International Conference on Robotics and Automation*.

Zhang, J. and A. Knoll (1998*a*). Constructing fuzzy controllers for multivariate problems using statistical indices. In: *International Conference on Fuzzy Systems*.

Zhang, J. and A. Knoll (1998*b*). Constructing fuzzy controllers with B-spline models - principles and applications. *International Journal of Intelligent Systems (forthcoming)* **13**(2/3), 257–286.

Zhang, J., Y. v. Collani and A. Knoll (1997*b*). On-line learning of sensor-based control for acquiring assembly skills. In: *Proceedings of the IEEE International Conference on Robotics and Automation*.