Autonomous Mobile Robot Navigation Including Door Opening Behavior
—System Integration of Mobile Manipulator to Adapt Real Environment—

Keiji Nagatani
(JSPS Research Fellow)
Scaife Hall, Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
Email:keiji@andrew.cmu.edu

Shin'ichi Yuta
Intelligent Robot Laboratory
Information Sciences and Electronics
Univ. of Tsukuba, TSUKUBA 305, JAPAN
Email:yuta@is.tsukuba.ac.jp

Abstract

In this paper, we report a design and experimental results of one intelligent robot motion. This research is performed along Task Oriented Approach. In this approach, a concrete task should be set up, which is necessary for advanced robot motion, but it allows a constraint condition of it. Our specific task is Autonomous Navigation Including Door Opening Behavior using a Mobile Manipulator in known indoor environment. Once our task was fixed, we made efforts to realize the task step by step. In the process of realization, we found not only several insights, but also problems common to mobile manipulators.

1 Introduction

Our objective is to realize intelligent and robust behavior for an autonomous robot. Especially, we focused autonomous behavior by mobile manipulator in indoor environment.

Generally, robotics research is performed next two steps,
1) To propose a general theory or algorithm for robot
2) To confirm an availability by simulation or experiment

We call this approach "Seeds Oriented Approach". Along this approach, a number of good theories and algorithms are proposed for the realization of intelligent robot motion. However, only basic motions are realized in real environments under strict conditions, and an intelligent versatile robot is a very long way away.

To grow out of this dilemma, our research group followed "Task Oriented Approach" for robotic research. This approach is performed,
1) Set up a concrete task
2) Make efforts to realize the task step by step

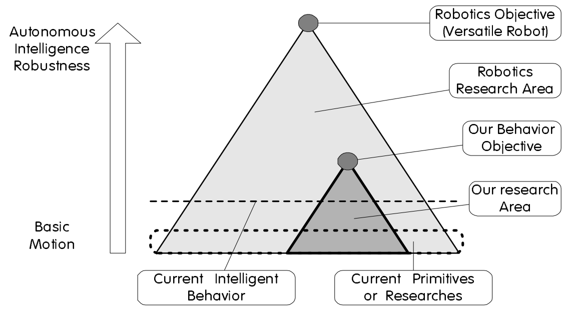
The task should be executed only by intelligent robot motion, however it allows a constraint condition. On the process of the realization, we may find new theory or algorithm which is directly available for actual problems.

Our objective along task oriented approach is shown in Figure 1.

In the task oriented approach, a task selection is very important. We fixed our research task as "Autonomous Navigation including Door Open Behavior", in an indoor environment. This task is not only one of basic skills for intelligent mobile robots, but it includes many research topics. In this task, to open the door by itself, the robot should be mobile manipulator. So motion planning and control are important topics. Currently, this topic is so popular that several research groups are researching about the topic with various approaches[O.Khatib *et al.*, 1996][Yamamoto and Yun, 1995][Azarm and Schmidt, 1994][Nagatani and Yuta, 1996].

We specified a target environment as our laboratory room and corridors in University of Tsukuba, and we assumed that a target robot possesses the environment information. The task of the robot is navigation from current position to a given goal. If the door is in the way of the robot, it passes through the doorway using its mounted manipulator and hand autonomously. Requirements of this task are (1) the environment should be real world and not be modified for robot motion (2) the robot should not hurt a human or the environment (3) the robot should cope with a unscheduled situation (e.g. door is locked).

In this paper, we analyze above a robot motion for the task realization, and describe the robot system of both hardware and software to realize the task. We also report several insights and problems for common mobile manipulators, which were found in the process of the task realization.


Autonomous Intelligence Robustness
Basic Motion
Robotics Objective (Versatile Robot)
Robotics Research Area
Our Behavior Objective
Our research Area
Current Intelligent Behavior
Current Primitives or Researches

NOTE: The top of this pyramid indicates one of the final goals of robotics; a "Versatile Robot". The whole pyramid represents the "robotics research area", and the base of the pyramid represents element technologies or foundation researches in robotics. A number of the element researches follow a Seeds Oriented Approach. However the level of robot intelligence is not so high in actual environment (Current Intelligent Behavior). In Task Oriented Approach, an advanced behavior is set as goal (Our Behavior Objective), which is supported by several research topics in constraint condition (Our Research Area).

Figure 1: Our Objective along Task Oriented Approach

2 Task Analysis

To realize the objective task, we divided the robot motion into two stages, "Motion Planning" and "Execution". In the Motion Planning stage, the robot plans a traveling trajectory and manipulator motion according to pre-knowledge information. In the execution stage, the robot executes the planned motion. To arrange the robot motion, we divided the execution stage into several sub-motions shown in Table 1. An analysis of each basic motion is discussed in following.

(1)	Traveling to front of Door
(2)	Grasping the Door Knob
(3)	Opening the Door
(4)	Releasing the Door Knob
(5)	Traveling to backside of the Door
(6)	Grasping the Door Knob
(7)	Closing the Door
(8)	Releasing the Door Knob
(9)	Traveling to Given Goal

Table 1: Mobile Robot Navigation with Door Opening

Trajectory Traveling

In stage 1, 5 and 9, an accurate trajectory following is required for the mobile robot. Ordinary, an odometry system is used for mobile robot navigation. However it generates positioning error caused by wheel slippage or rough surfaces, and the error is accumulated in proportion to the traveling distance. Therefore, external sensors should be used for adjustment of the robot position using the environment model.

Door Knob Grasping

In stage 2 and 6, the robot needs to find the door knob accurately by external sensors even if it knows the accurate position of the knob. The reason is positioning error of the mobile robot.

Door Opening / Closing

In stage 3 and 7, the robot should open the door using a mounted manipulator with the mobile base moving forward, because the general length of the manipulator for the mobile robot is not long enough. Also, a large force might occur at the grasping point of door knob due to the positioning error of mobile robot. Therefore a compliance mechanism is necessary for mounted manipulator control.

Releasing Door Knob

In stage 4 and 8, the robot releases grasping door knob by control of robot hand, and initializes the manipulator pose.

3 Motion Design and Construction of Robot System

According to task analysis, we designed each basic robot motion shown in Table 1.

In stage 1, 5 and 9, the mobile robot follows a given trajectory using its odometry system. To cope with the positioning error of odometry, the robot uses wall information which is detected by mounted ultra-sonic range sensors.

In stage 2 and 6, the manipulator is controlled to approach the knob, and vision feedback is used to adjust the approach trajectory of the hand.

In stage 3 and 7, cooperation control between locomotion and manipulation is necessary for door opening. Also, compliance control for the mounted manipulator is executed using a force / torque sensor.

According to above, the robot must possess the following functions for realization of the navigation task.

(a) Locomotion function
(b) Odometry system for Position Estimation
(c) Position Adjustment by ultra-sonic sensors
(d) Hand Control
(e) Knob's Position recognition by vision
(f) Pose Control of Manipulator
(g) Compliance Control of Manipulator

We developed a self-contained mobile manipulator system to realize above functions. The robot is named "Yamabico Ten ", shown in figure 2.

It is constructed from two parts, a mobile base and a manipulator. The mobile base has two driving wheels with encoders for the odometry system, and 8 directional ultra-sonic range sensors. The manipulator is light weight 7 degrees of freedom. The force sensor, vision sensor, and a two fingered robot hand are mounted at the top of the manipulator.

Each actuator or sensor was developed individually, and is controlled by each computer system which includes a real-time Operating System and low level program (e.g. motor servo software). We call each individual function mechanism as a "Function Module". All function modules are controlled by a decision making controller, called the "Master Module", which includes high level software to control the whole robot motion. The master module communicates with the other functional modules asynchronously through our original bus system. A transputer link (high speed serial link between CPUs) is used for a direct communication between function modules which is necessary for high speed information transferring. We call the whole controller network system as "Function Distributed Controller "[Suzuki *et al.*, 1993].

4 Software

The robot control software is structured hierarchically according to the function distributed controller. We classified the software layers as (1) function programs, (2) action primitives and (3) coordination program, shown in Figure 3. Each function program is executed on each control board to control the hardware directly. Several basic function programs have been developed in our research group in the past 10 years[Iida and Yuta, 1991][Watanabe and Yuta, 1990].

An action primitive is an execution program to perform one basic motion, corresponding to each stage in Table 1. It is implemented as one process on master module, and it communicates with function programs through BUS system. The coordination program has a charge of whole robot behavior. It is also implemented as one process on master module, and executes a suitable action primitive process by interprocess communication. The details of each programs are introduced in the next subsections.

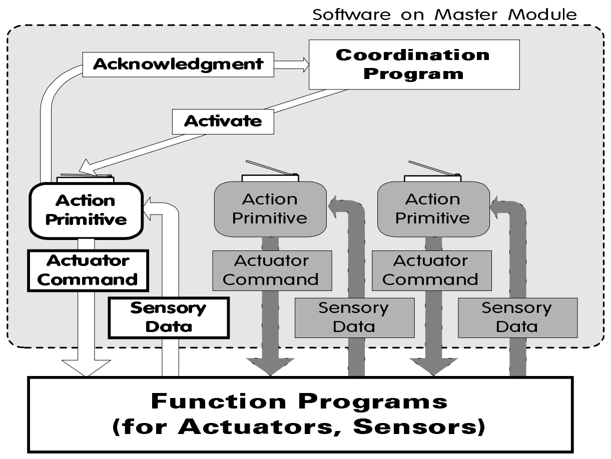


Figure 3: Software Structure

4.1 Function Programs

1. Manipulator Control Program

   The manipulator control program is a motor servo program for pose control of manipulator with compliance, located on manipulation module. Each reference joint angle is sent from programs on master module through bus system, and force information comes from force sensor module through transputer link. Basically PID control is used for pose control, and compliance of the manipulator is realized by sliding the reference position of the top of the manipulator along the force direction.

2. Locomotion Program

   Locomotion program is a motor servo program for driving wheels, located on a locomotion module to follow the trajectory issued by the program in the master module.

3. Position Estimation Program

   Position Estimation Program uses Kalman Filtering method [Watanabe and Yuta, 1990][S.Cooper and H.Durrant-Whyte, 1994] to estimate mobile robot position, which is located on the Position Estimation Module (POEM).

   Basically, the robot estimates it's position and calculates a dispersion by odometry system. Using positioning error information from external sensors (in our case, flat walls are assumed to be landmarks and ultra-sonic range sensors are used to detect them), the program adjusts the estimated position and orientation of the robot. Adjusted position informa-

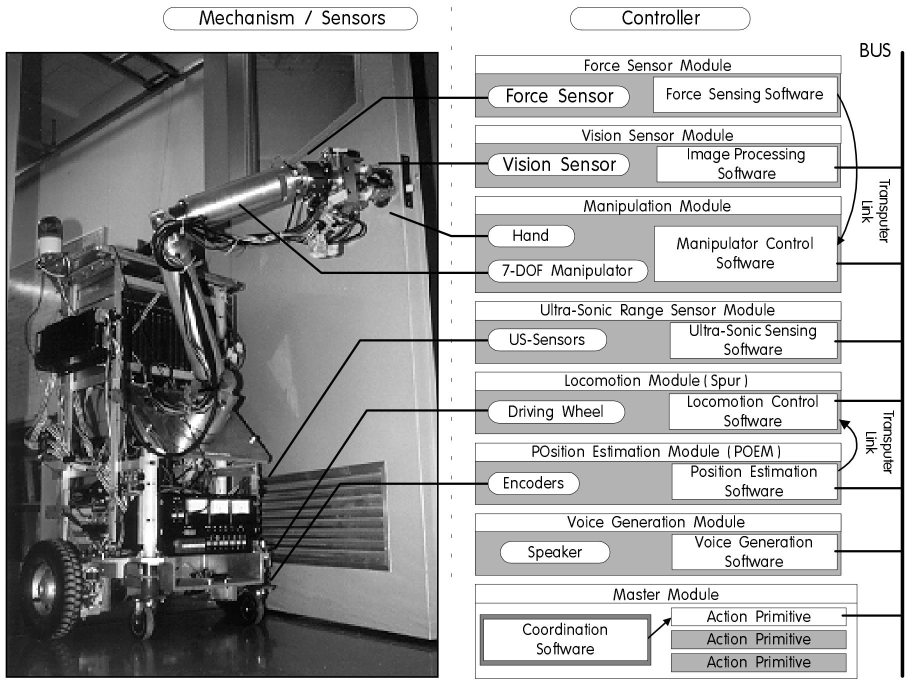


Figure 2: Function Distributed Controller on Yamabico TEN

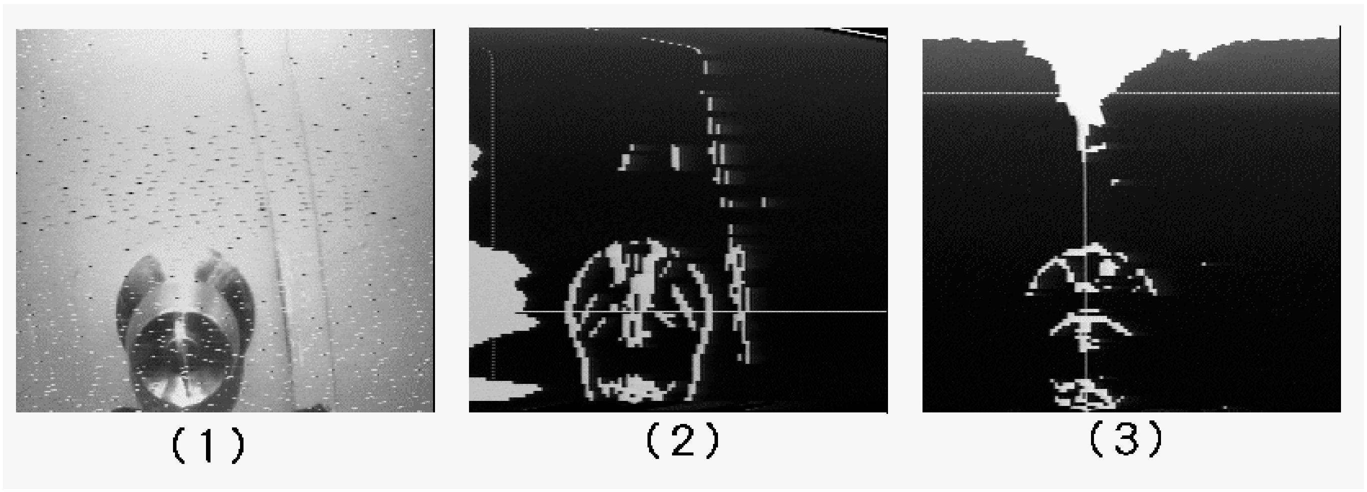


Figure 4: Image Processing of the Door Knob

tion is sent to the locomotion module for trajectory following.

4. Image Processing Program

   Image Processing Program is used for getting position data of a target door knob, located on vision sensor module. To abstract the center position of the door knob, we use differential image processing and calculate the center of gravity of it, which is shown in Figure 4. The result is referred to by programs on the master module.

5. Force Sensor Program

   Force Sensor Program is converting program from raw data to actual force data on hand coordinate. The converted sensor information is sent to manipulator control program directly through the transputer link.

6. Ultra-Sonic Range Sensor Program

   Ultra-Sonic Range Sensor Program is located on Ultra Sonic Range Sensor Module, and calculates eight directional distances to the objects using ultra sonic range sensors. Detected sensor data is referred to asynchronously by the master module through bus system.

7. Voice Generation Program

   The Voice Generation Program is located on the voice module, and generates verbal comments using a voice generator IC to describe the internal robot status for debugging purpose.

4.2 Action Primitives

An action primitive is an execution program for one basic motion decomposed along a time shown in Table 1. Each action primitive is implemented as one process on a master module, and can communicate with function modules through out our bus system. Each action primitive includes a sensor feedback system for robust motion.

1. Action Primitive "Trajectory Traveling"

   This action primitive is used to move the mobile robot along a planned trajectory.

   Basically, the mobile robot follows a planned straight line according to odometry information. The whole path of the mobile robot is expressed by an assembly of straight line segments and turning angles.

   Once this action primitive process is executed, it sends a suitable locomotion parameter to the locomotion program to follow the planned path. Also, it checks the environment information from the ultrasonic range sensor program. When the sensor detects an object, this action primitive compares the data with environment model. The difference between the sensor data and the internal model gives a positioning error, and the error is sent to the Position Estimation program to adjust the estimated position. When the action primitive detects an unscheduled obstacle in front of the robot, it issues a stop command to the locomotion program and waits until the object moves away. This loop continues until the robot arrives at a sub-goal.

2. Knob Approach

   This action primitive is to approach the robot hand to the knob and grasp it by controlling of mounted manipulator.

   This action primitive sends the joint parameters to the manipulator control program according to the planned hand approaching motion. During the motion, the action primitive gets location information about the target knob from the Image Processing program, and adjusts the grasping position several times. The motion continues until the hand contacts with the door which is detected by a force sensor. After that, this program sends a hand closing command parameter to the manipulator control program to grasp the knob.

3. Knob release

   This action primitive is to release door knob and to initialize a pose of manipulator. A sequence of the initialization is pre-planned and this action primitive sends the pose parameters to the manipulator control program step by step along the sequence.

4. Door Opening/Closing

   This action primitive is to open/close the door using locomotion function and manipulation function.

   The size of the target door and the size of the mounted manipulator is known in advance, so the cooperation motion between locomotion and manipulation is planned off line in advance. An example of a door opening plan is shown in Figure 5.

   The planned motion is expressed by 65 sets of robot positions and joint angles for the manipulator. At the beginning, the action primitive send a locomotion command to the locomotion program to follow the planned trajectory slowly. Then the action primitive sends the suitable manipulator pose parameter to the manipulator control program. The parameter is specified by the planned motion and estimated position data of the mobile robot from the Position Estimation program.

4.3 Coordination Program

The role of the coordination program is to plan a sequence of action primitives, and execute them step by

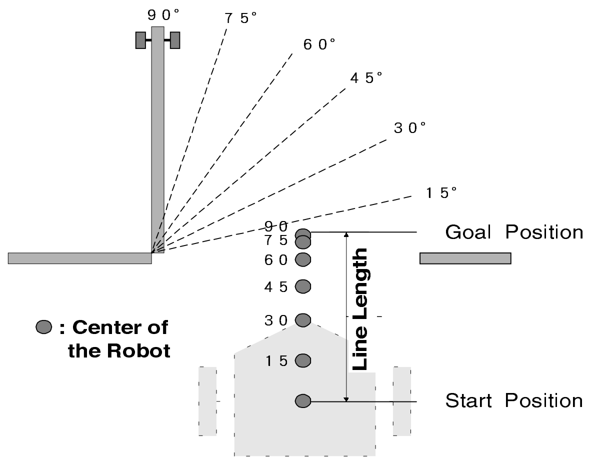


Figure 5: The Robot Motion of Door Opening

step along the sequence.

To plan a sequence of action primitives, we expressed every possible robot status as a network, called the "action network ". In the network, each node represents a robot status, in other words, it is a turning point of the robot motion. The nodes are connected by a arc which corresponds to one action primitive. Each arc includes an execution cost, parameters for robot motion, and environmental information to perform the action primitive. Once we represent the network, robot motion planning becomes a graph search problem on it. A example of the action network to pass through a doorway is shown in Figure 6.

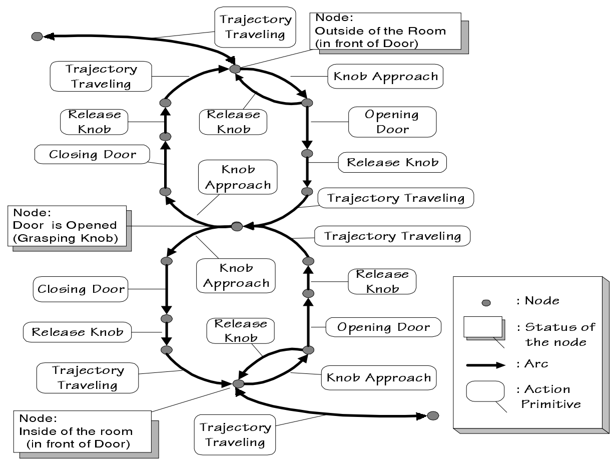


Figure 6: Action Network

Using the action network, the whole motion of the robot is executed by the coordination program along following sequence.

(1) The coordination program generates a sequence of action primitives by a standard graph search (Dijcstra Algorithm [E.W.Dijkstra, 1959]) on the action network to minimize the execution cost.

(2) According to the planned sequence of action primitives, the coordination program sends motion parameters to the action primitive process using interprocess communication.

(3) The coordination program waits until the acknowledgment comes back from the action primitive process.

(4) If the acknowledgment is "Normal", the coordination program continues to execute the next action primitive (go to (2)). If the acknowledgment is "Could not Complete", the robot can not continue the planned motion. In that case, the coordination program cuts the arc of the failed action primitive, and graph searches to generate another sequence of action primitives (go to (1)).

Above procedure is also shown in figure 7.

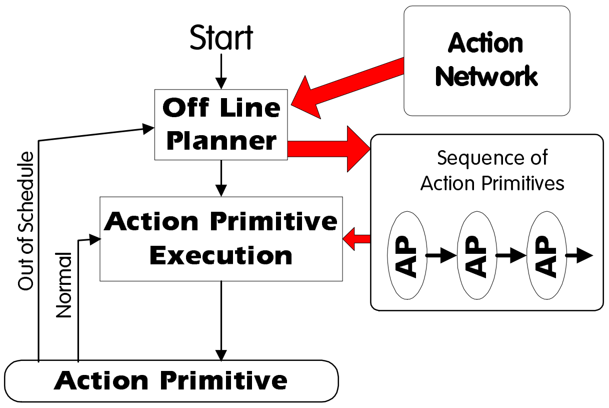


Figure 7: Coordination Software

5 Experiment

We implemented the hardware and software shown in the previous section, and realized autonomous navigation including door opening by Yamabico TEN. One example of the experimental result is showin in Figure 8.

The success ratio for completing this mission (Navigation from initial position to given goal) is about 90 percent.

6 Discussion and Conclusion

We realized autonomous, intelligent and robust motion of the mobile manipulator using a task oriented approach. We also found several actual problems and insights about mobile manipulator along task oriented approach.

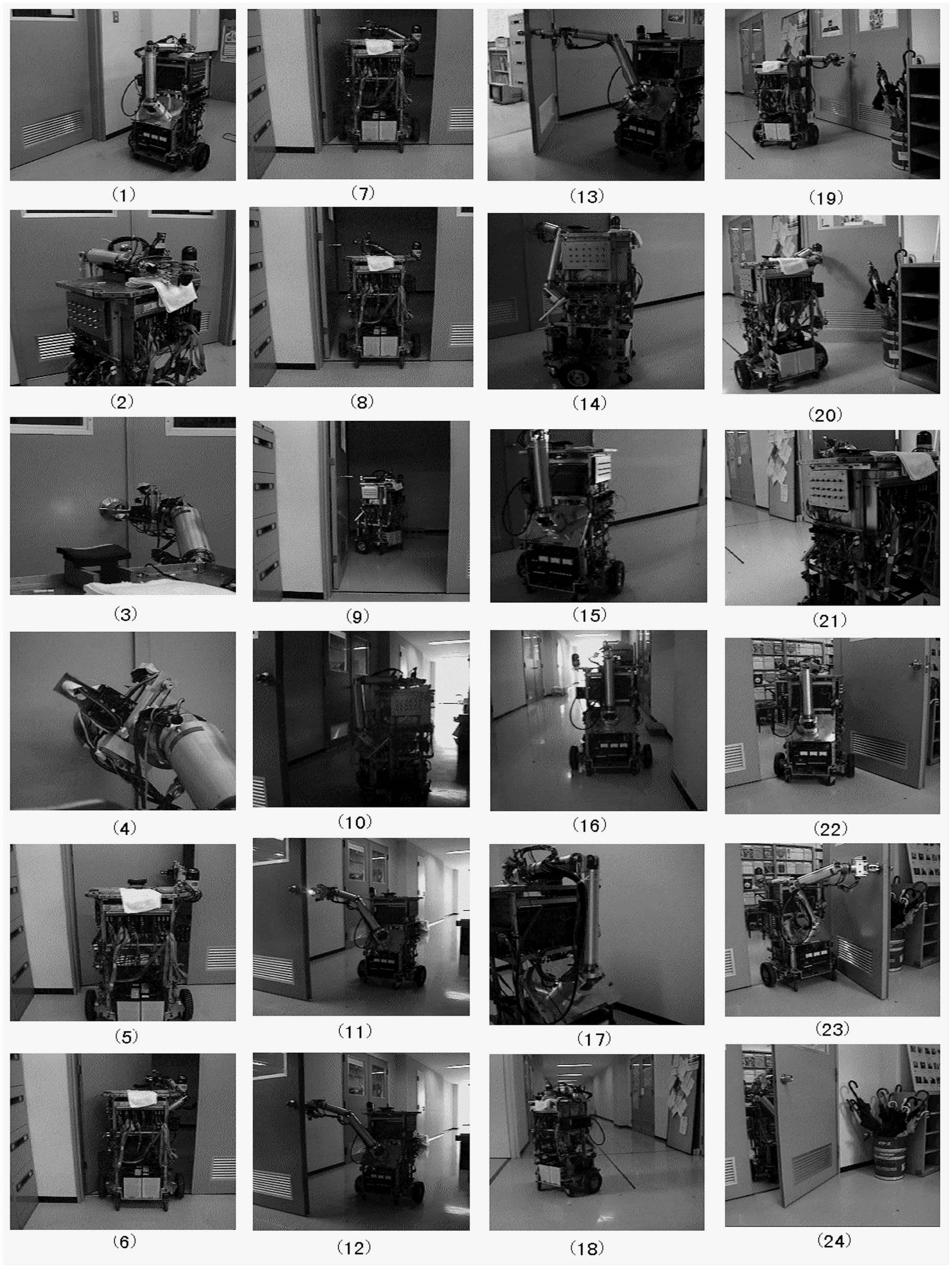


Figure 8: Autonomous Navigation with Door Opening Behavior

1. Positioning Error problem of mobile robot

   The biggest problem was positioning error of the mobile base, even if we coped with them in every stage. Most failures of the motion come from positioning error of the mobile robot. In our impression, motion planning of the mobile manipulator is a small part, and the actual problem is the positioning error problem of the mobile robot. Therefore, we can not discuss a mobile manipulator matter without positioning error.

2. Force Treating at the contact point

   The force at the grasping point is generated from positioning error, so it is the same topic as treating a positioning error problem in a wide sense. In this research, we treated the problem by realizing compliance control of the manipulator. However, the compliance motion may cause the problem in narrow environment, which is not considered in this research.

3. Grasping Problem

   Sometimes in the experiment, the robot could not grasp a knob tightly. It caused a failure to rotate the door knob, because the grasping sequence was open loop in our algorithm. Then the robot recognized that the door is closed even if the key was not locked. To aim at certain behavior, it is necessary to confirm every motion by sensors.

4. Importance of a light weight manipulator

   We developed light weight manipulator with a mounted force sensor, vision sensor and robot hand. In our experience of this research, we recognize that developing a small manipulator unit with these functions as a research platform is very important for mobile manipulator research.

5. Importance of function distributed controller

   This research was carried out as is one of the "Yamabico project", which developed a function distributed controller. Therefore we used several research resources of the Yamabico project (e.g. locomotion controller and sensor systems), and our system development time was shortened very much.

6. Importance of motion decomposed along actual time

   It is popular to decompose complex motion into several basic functions. However, it is very difficult for robust motion to decompose according to functions, because each function should cooperate with each other even if performing one basic motion. We devised the concept of the action primitive which is decomposed along in actual time, and verified the effectiveness of this approach for planning of whole motion, and robust behavior.

According to above results and discussion, we believe that the task oriented approach is necessary to realize an intelligent robot motion, and this research becomes a milestone of robotics research.

References

[Azarm and Schmidt, 1994] Kianoush Azarm and Gunther Schmidt. Integrated mobile robot motion planning and execution in changing indoor environments. In *Proc. of IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pages 298–303, 1994.

[E.W.Dijkstra, 1959] E.W.Dijkstra. A note on two problems in connection with graph. *Numerische Mathematik*, 1:269–271, 1959.

[Iida and Yuta, 1991] Shigeki Iida and Shini'chi Yuta. Vehicle command system and trajectory control for autonomous mobile robots. In *Proceedings of International Workshop on Intelligent Robots and Systems*, pages 212–217, 1991.

[Nagatani and Yuta, 1996] Keiji Nagatani and Shin'ichi Yuta. Designing strategy and implementation of mobile manipulator control system for opening door. In *Proc. of IEEE International Conf. on Robotics and Automation*, pages 2828–2834, 1996.

[O.Khatib *et al.*, 1996] O.Khatib, K.Yokoi, K.Chang, D.Ruspini, R.Holmberg, and A.Casal. Coordination and decentralized cooperation of multiple mobile manipulators. *Journal of Robotic Systems*, 13(11):755–764, 1996.

[S.Cooper and H.Durrant-Whyte, 1994] S.Cooper and H.Durrant-Whyte. A kalman filter for gps navigation of land vehicles. In *Proc. of IEEE/RSJ International Conf. on Intelligent Robots and Systems*, pages 157–163, 1994.

[Suzuki *et al.*, 1993] Sho'ji Suzuki, Jun'ichi Iijima, and Shin'ichi Yuta. Design and implementation of an architecture of autonomous mobile robots for experimental researches. In *Proc. of 6th International Conf. on Advanced Robotics*, pages 423–428, 1993.

[Watanabe and Yuta, 1990] Yutaka Watanabe and Shin'ichi Yuta. Position estimation of mobile robots with internal and external sensors using uncertainty evolution technique. In *Proc. of IEEE International Conf. on Robotics and Automation*, pages 2011–2016, 1990.

[Yamamoto and Yun, 1995] Yoshio Yamamoto and Xiaoping Yun. Coordinated obstacle avoidance of a mobile manipulator. In *Proc. of IEEE International Conf. on Robotics and Automation*, pages 2255–2260, 1995.