System Integration of a Daily Assistive Robot and its Application to Tidying and Cleaning Rooms

Kimitoshi YAMAZAKI, Ryohei UEDA, Shunichi NOZAWA, Yuto MORI,
Toshiaki MAKI, Naotaka HATAO, Kei OKADA and Masayuki INABA

Abstract— This paper describes a software system integration of daily assistive robots. Several tasks related to cleaning and tidying up rooms are focused on. Recognition and motion generation functions needed to perform daily assistance are developed, and these functions are used to design various behaviors involved in daily assistance. In our approach, the robot behaviours are divided into simple units which consist of 3 functions as check/plan/do, it provides us with high reusable and flexible development environment. Because sequential task execution can be achieved only after functions about failure detection and recovery, we also try to implement such functions in keeping with this approach. In addition to using simple behavior unit, multilayer error handling is effective. Experiments doing several daily tasks with handling daily tools showed the effectiveness of our system.

I. INTRODUCTION

In daily environments, various types of furniture and tools exist for human lives. Daily assistive robots are expected to work by handling such daily things to achieve helpful assistance. This paper describes a software integration and shows an application; "cleaning and tidying rooms" by a robot in daily environments.

In general, daily routine works include various object manipulations. Because recent robots have many of DOFs like a human, it can be said that such robot has sufficient potential to replace housekeepers in chores. However, a wealth of recognition and manipulation skills is also needed to the robot. The purpose of this research is to develop highly integrated software system which permits a robot to achieve plenty of daily works. In our system, 3D geometrical simulator is centered and essential functions of environment recognition and motion generation are combined by the simulator. In addition, hierarchical task management framework for failure recovery is also introduced.

This research copes with "tidying and cleaning rooms" task which includes some chores as follows:

- Carry a tray on a table to a kitchen,
- Correct clothes in rooms and put them into a washer machine,
- Clean a floor by using a broom.

Because these tasks need several manipulation skill such as dual arm manipulation, soft object manipulation, doors opening, button pressing and so on, good examples of daily assistance can be shown through their implementation.

Department of Mechano–Informatics, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan {yamazaki, ueda, nozawa, y-mori, maki, hatao, k-okada, inaba}@jsk.t.u-tokyo.ac.jp

This paper is organized as follows: Section II describes related works and our approach. Section III to VI introduces our integrated system and explains each functions. Section VII explains the task description and failure recovery system. Section VIII describes experimental results, and section IX concludes this paper.

II. RELATED WORKS AND OUR APPROACH

Daily assistive robots have been developed over several decades. Researchers have evaluated their control system, intelligent system or teaching system with applying their method to a single daily task in real environment [5],[7],[9],[16]. In the viewpoint of system integration, Petersson[15] et. al. developed a mobile manipulator system which could pick an instructed object up, convey, and hand it to a person.

In recent years, daily assistance by using humanoid robots becomes an active area of robotics research[1],[12]. Sugano et. al. presented assistance behavior by using a human symbiotic robot which has object manipulation skills[19]. We also have developed daily assistive robots provided perception, learning and motion planning skills. Several daily tasks or cooperative working etc. were implemented[14].

Generally speaking, effective daily assistance can be achieved by a single robot which has several abilities for daily tasks execution. Additionally, daily tasks should be performed continuously. For instance, "Cleaning and Tidying up rooms" aimed in this paper includes several series of works, and housekeepers carry out them one after another. Despite this fact, few robotics researches related to daily assistance have reported such sequential task execution. Our purpose is to develop and to proof an integrated system which can perform various tasks existing daily life.

III. SYSTEM INTEGRATION FOR BEHAVIOR GENERATION

We aim to build an integrated software system for a robot which has plenty of degrees of freedom like a human. This section describes the basic policy.

A. Previous knowledge

Manipulation targets satisfy following conditions:

- 3D geometrical model is given in advance. If the object has articular structure, the information is also added to the model.
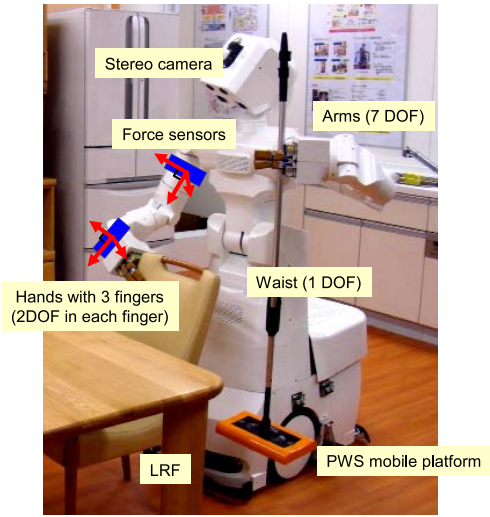- The pose of a target object is given in advance. However, a certain level of error is permitted because it is

Fig. 1.   A Daily Assistive Robot

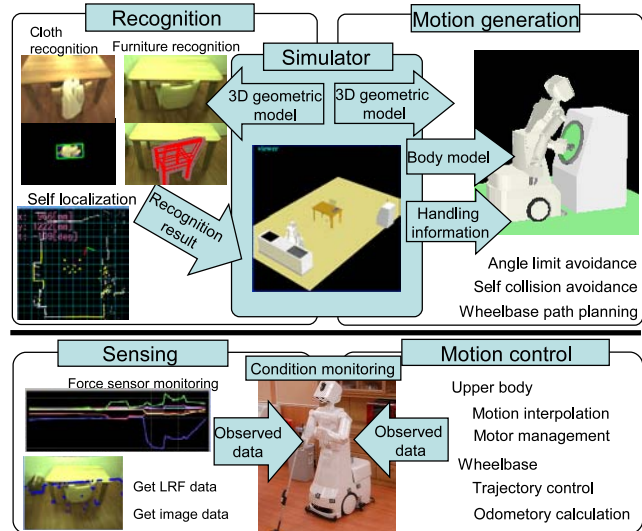| Deimension	H 1550 mm x W 500 mm x D 650 mm
Weight	130 kg
Head	Neck 3 DOF (Yaw-Pitch-Roll)
Arms	7 DOF (Shoulder Pitch-Roll-Yaw), (Elbow Pitch), (Wrist Yaw-Pitch-Yaw)
Hands	3 Fingers (Each of fingers has 2 pitch joints)
Waist	1 DOF (Pitch)
Mobile Platform	Two wheeled mobile base (Tred 500 mm, Wheel Diameter 170 mm)

Fig. 2.   Software Architecture

assumed that the robot estimates and corrects the error automatically.

- The robot has the basic knowledge of its manipulation target. This means that the robot knows what features can be used to recognize the target and which sensors should be used for effective recognition.

Although this policy indicates that environmental models are given, It is predicted that manufacturer will provide robots with these model data in the future.

B. A daily assistive robot

Fig. 1 shows a daily assistive robot in our use. Upper body consists of 2 arms with 7 joints and a head with 3 joints, and a waist with 1 joint. End-effectors equip 3 fingers and each finger has 2 joints. In order to grasp an object with palm, these fingers are fixed without locating to diagonal pair. The lower body is realized though a wheeled mobile platform with two active wheels and 4 passive wheels. (See Table.I)

On the other hand, this robot mounts a stereo camera (STH-MDCS3 made by VIDERE Design Inc.) on the head, and a LRF (Laser RangeFinder, LMS200 made by SICK Inc.) on the wheelbase. Force sensors are also equipped on the wrists and shoulders.

To develop robot system for achieving tasks as previously indicated, behavior generation functions constituted from mobility, dual arm manipulation and dextrous handling are needed. Meanwhile, recognition functions constituted from environment recognition, self monitoring and positioning should also be satisfied.

C. Software system overview

Fig. 2 indicates an integrated system. Recognition functions and motion generation functions are densely combined with 3D geometrical simulator. The simulator provides 3D shape and appearance with the recognition functions, and also provides handling information with the motion generation functions.

Meanwhile, a different layer is implemented to observe and to manage the robot state in realtime. For instance, collision checking of wheelbase by using LRF, measuring of joints load and so on. These functions drew upon plugin system described in [8]

"Pose" described in this paper means the state of combining its position and direction.

IV. ENVIRONMENT RECOGNITION

It is assumed that the handled targets while tidying and cleaning are 5 series of objects as follows: tray, chair, washing machine broom and cloth. Because former 4 objects can be regarded as solid objects, 3D geometrical models are used to recognize their poses. On the other hand, in the case of soft object as a cloth, the role of recognition function is to find the target in daily environments, and to detect its existing position.

A. Pose estimation based on a geometrical model

External sensors the robot equips are a stereo camera on the head and a LRF on the wheelbase. Pose estimation of large size furniture such as a chair and a washing machine uses both of these sensors. On the other hand, the stereo
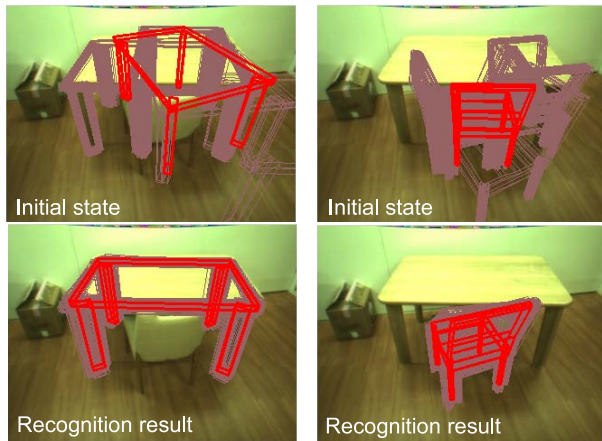
Fig. 3. Pose estimation of a table and a chair. Wireframe models illustrated as dark-red lines show candidates of target objects. Red wireflame models at lower figures show pose estimation results.

Fig. 4. Cloth detection based on wrinkly features in an image

camera is only used in the case of a tray, which cannot be observed by LRF. The matching procedure is as follows: firstly, 3D geometrical model is virtually placed in a simulator world, and its appearance is projected to an image which is captured by the stereo camera. By matching the projected model with several types of image features, the degree of estimation accuracy is evaluated. We apply particle filter to this estimation process according to following rules:

$$p(\mathbf{x}_t|Z_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|Z_{t-1})d\mathbf{x}_{t-1}, \quad (1)$$

This equation indicates prior probability which is calculated from an object pose $\mathbf{x}_t$, a sensor measurement $\mathbf{z}_t$. We denote $Z_t = \{\mathbf{z}_i, i = 1, ...n\}$.

The posterior probability $p(\mathbf{x}_t|Z_t)$ can be calculated obeying Bayes rule as follows:

$$p(\mathbf{x}_t|Z_t) \propto p(\mathbf{z}_t|\mathbf{x}_t)p(\mathbf{x}_t|Z_{t-1}), \quad (2)$$

where $p(\mathbf{z}_t|\mathbf{x}_t)$ denotes the likelihood at each time.

In addition to LRF data, because the estimation needs easily extractable image features like edges, line segments and colors, this method is suitable to recognize the pose of patternless furniture. Fig. 3 shows examples of pose estimation.

B. Cloth recognition based on wrinkly features[17]

A recognition approach without 3D geometrical model is needed to find clothes because they have a soft body. So we take an approach to find wrinkles in images. Our method involves image leaning to define the wrinkly features.

In the learning process, gabor filtering is first applied to several images which capture clothes placed on daily environment. Next, we crop partial images including cloth region. On the other hand, background regions are also cropped randomly. 20 bins of histograms are calculated from these regions, a vector of discriminant function is calculated by following equation:
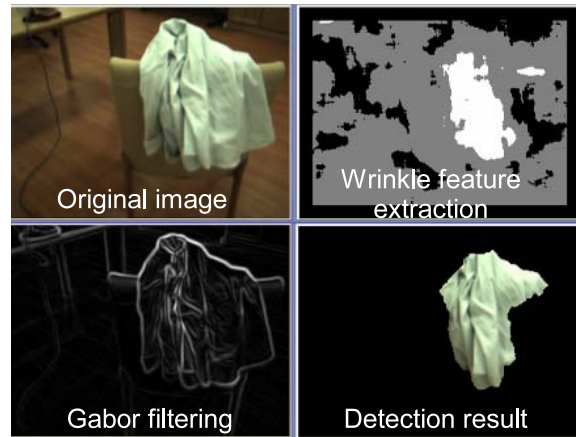
$$L(\mathbf{w}, h, \alpha) = \frac{1}{2}|\mathbf{w}|^2 - \sum_{i=1}^{n} \alpha_i \{t_i(\mathbf{w}_i^t\mathbf{x}_i - h)\}, \quad (3)$$

$L(\cdot)$ denotes Lagrange function. $\mathbf{w}$ and h are parameters of discriminant function. $\mathbf{x}_i$ denotes ith data for learning.

In the recognition process, an image is divided to 3 regions; (1) the region which can obviously be judged as a wrinkle region, (2) non-wrinkle region, (3) unclear region. These are calculated by using following equation:

$$y = \mathbf{w}^t\mathbf{x}_j - h, \quad (4)$$

the region belonging (3) is segmented by using graphcut[3] with regarding region (1) and (2) as seeds. Fig. 4 shows an example of a shirt detection.

This process results to judge whether clothes exist or not in the front of the robot. Moreover, because of using stereo camera, 3D position of the cloth is calculated.

C. Attention area extraction and change monitoring

For successful object manipulation, one of the effective ways is to visually confirm the object while doing the manipulation. Two type of functions are provided to estimate manipulation state. One uses specific color extraction, and the other uses differentiation between two images. Extracted regions through these methods are classified, their shapes and areas are utilized to judge whether or not the manipulation is well.

V. MOTION GENERATION

A. **Upper body motion**

In order to detect gazing and handling points, coordinates embedded in a 3D object model are referred. Jacobian-based inverse kinematics is calculated based on these coordinates. Especially, we utilize SR-inverse[11] which has a good track record in stability around singular point.

The equation to calculate angle velocities $\dot{\theta}$ is as follows:

$$\dot{\theta} = \mathbf{J}_w^{\#}\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}_w^{\#}\mathbf{J})\mathbf{y} \quad (5)$$
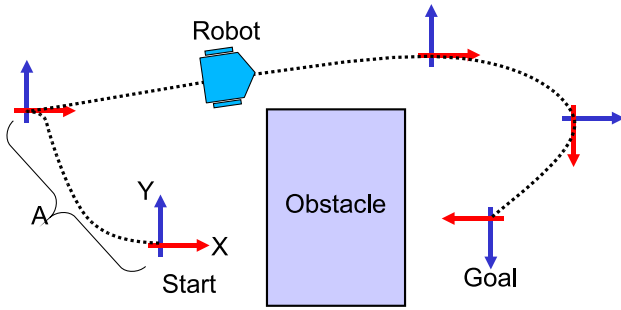
Fig. 5.   Wheelbase motion

Fig. 6.   Task structure in case of chair manipulation

where $\mathbf{J}^{\#}$ is a SR-inverse of $\mathbf{J}$, and $\mathbf{J}_w^{\#}$ is a multiplication result of $\mathbf{J}^{\#}$ and weight matrix $\mathbf{W}$. The diagonal matrix $\mathbf{W}$ is determined from following eq. [4]:

$$w_i = \left| \frac{(\theta_{max} - \theta_{min})^2 \, (2\theta_i - \theta_{max} - \theta_{min})}{4 \, (\theta_i - \theta_{max})^2 \, (\theta_i - \theta_{min})^2} \right| \quad (6)$$

In the equation(6), w_i is replaced on $1/(1 + w_i)$ when the value becomes smaller than pervious value. In reverse case, it replaced on 1. Such arrangement plays a role in making small weights when joint angles next to angle limits. $\mathbf{y}$ indicates an optimization function for avoiding self collision by using redundant degrees of freedom.

B. Wheelbase motion

Because we focus on doing several types of chores by a single robot, the robot has to shift its position where each task is performed. The wheelbase is controlled based on line trajectory tracking. Fig. 5 shows an example of navigation. Basically the trajectory of the platform is determined from a set of coordinates which are discretely allocated on the floor. The initial direction of the robot is the same as x axis of the initial coordinates. The controller outputs velocity v and angular velocity ω with considering relative pose of coordinates. In this case, the robot first goes back at interval $'A'$ in Fig. 5 and after that, goes in the goal coordinates via remained coordinates.

Initial wheelbase poses of every task are previously defined on an environment map. When the robot moves to an initial position of another task, or retry a task from the beginning, the motion generator provides the robot with a set of coordinates to shift toward target pose.

VI. LOCALIZATION AND STATE MONITORING

A. Wheelbase localization

Wheelbase localization is achieved by using LRF (Laser RangeFinder) mounted on the wheelbase. Environment map was generated by SLAM in advance, a present robot pose is calculated by means of scan matching[6].

In the map generation phase, we apply SLAM approach which combines ICP algorithm based on scan matching[2] and GraphSLAM[10]. Because this map is represented as dozens of reference scan and robot positions, ICP algorithm can be used to match between input scan and reference scans
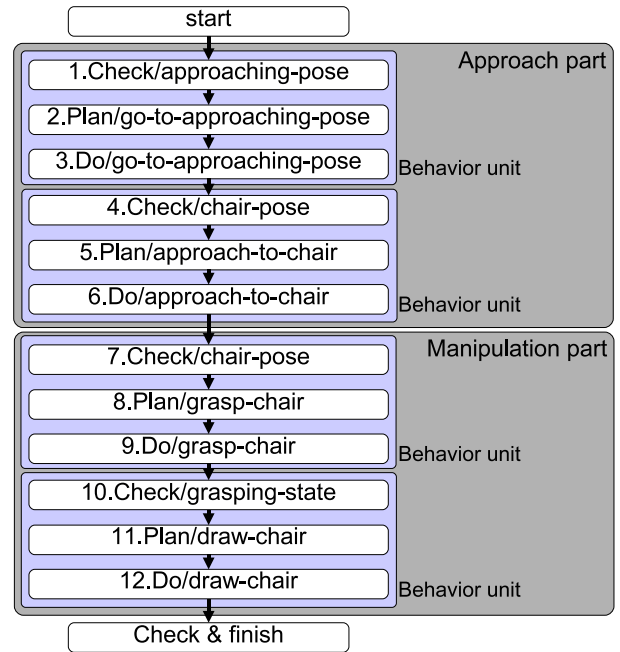
in the localization phase. However, this matching is subject to fail when the wheelbase rotates steeply. In order to eliminate such mismatching, the information of odometory changes from time $t - 1$ to t is added to the scan matching.

B. State monitoring

When the robot performs object manipulation, its state such as load to joints should be observed to detect manipulation failure. From this reason, monitoring functions are kept in good working order. For instance, (i)load monitoring using force sensors in arms, (ii)difference monitoring between joint angles of a present pose and that of reference pose. To ensure a collision-free navigation, collision risk of a wheelbase is also checked by using LRF.

VII. BEHAVIOR DESCRIPTION OF DAILY ASSISTIVE TASKS

To achieve various daily assistance by a life-sized robot, unified framework of behavior description is needed. Moreover, the framework should eliminate task failures. This section describes the policy to which we apply.

A. Basic configuration

A manipulation behavior consists of "approach" and "manipulation". The approach part has a role in finding its manipulation target and going near to it. Meanwhile, the manipulation part has a role in confirming the pose of the target, planning robot motion and executing it. Now we call the smallest description of behavior "behavior unit" which is a set of recognition, motion generation and motion execution. That is, daily assistive tasks are represented as approach phase and manipulation phase, and each tasks consists of several behavior units.
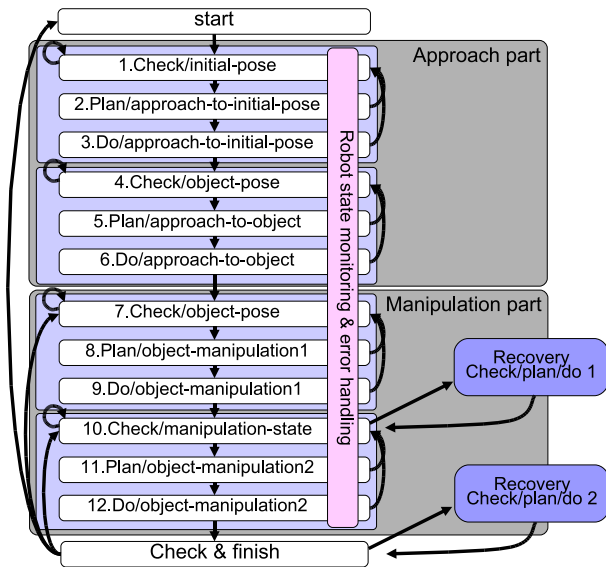
Fig. 7. Task structure including standard failure recovery

Fig. 8. Experimental Environment

Fig. 6 shows an example of behavior description in the case of chair handling. The advantage of this framework is that each of the behavior units can specifically be reused as the situation demands. For instance, when a hand which grasps a chair is isolated while pulling it, the error immediately detected by means of state monitoring such as force load to the wrist or joint angles of fingers. In such case, the task can be continued by transit function from behavior unit 12. to 7., see Fig. 7

B. Classification of failures

The word "Failure" in this paper means the condition that the result of sensor measurement is different from assumed one, and the fact adversely affects task execution. Because cleaning and tidying needs to perform several number of sub-tasks continuously, it is not approvable to stop the execution when one failure is detected. From this reason, failure detection and recovery are absolutely necessary.

There are many type of failures and many ways to recover from it. Examples in our task are that a cloth hangs out of a washing tab, or a broom lies down on a floor because the robot failed to grasp it. In order to establish basic structure for error recovery, we classify observed failures to 3 groups from the viewpoint of the levels of recovery intractableness.

- **(A) Failures observed before manipulation :** This occurs while approaching or at the beginning of object manipulation. One of the examples is that the robot cannot plan its handling pose because of wheelbase positioning error after it approaches to manipulation target.
- **(B) Failures observed after manipulation, without almost no changes of the manipulation target :** This failure occurs in a manipulation part and it is observed by checking sensing data whether or not the data is similar to the assumed one. If the robot can recover a
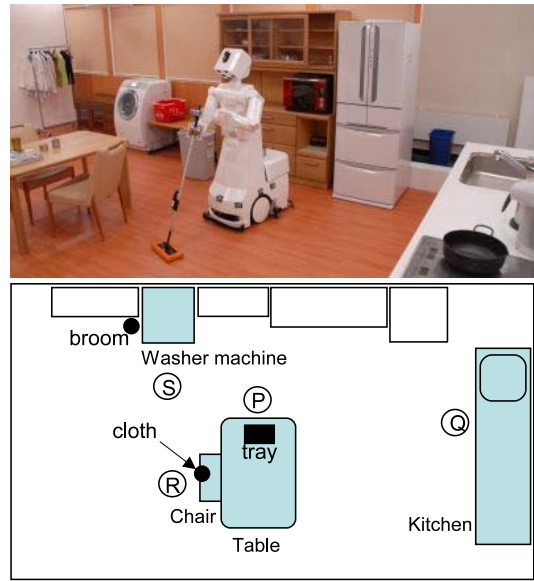
failure just by recalling the same behavior units again, it is classified to this group.
- **(C) Failures with changing the manipulation condition significantly :** Although this failure is observed by the same way as (B), the way of recovery is differ from (B) because simply behavior repetition does not make sense in this case. For instance, if the robot tries to grab a broom propped against a wall but drops it down, picking up motion is newly needed.

C. The procedure of failure detection and recovery

Fig. 7 shows state transition diagram with nodes for failure recovery. In the case of failures classified to (A), the system should find failures at "check" function or "plan" function, and calls the "check" function again. If an adequate result cannot be acquired through this process, a behavior unit which has already been called is used again. In the case of (B), recovery is achieved with moving from "do" function to "check" function in the same behavior unit. If it cannot, a behavior unit which has already been called is used as the same case as (A).

On the other case (C), a completely different behavior unit is called first. If the behavior succeeded, the system returns a process to a behavior which was originally executed. For instance, the robot finds a cloth which gets out of washing tab, a new motion is inserted to pick the cloth up. Next, original behavior unit for putting the cloth into the washer is called again.

Although failures described above can be found at "check" and "planning" functions, there is another type of failures. For instance, if the robot unexpectedly finds an abnormal state that force measurements go down while grabbing a manipulation target, it should be judged as a fail at that moment. Such sudden failures in the middle of running a function are managed "state monitoring & error handling"

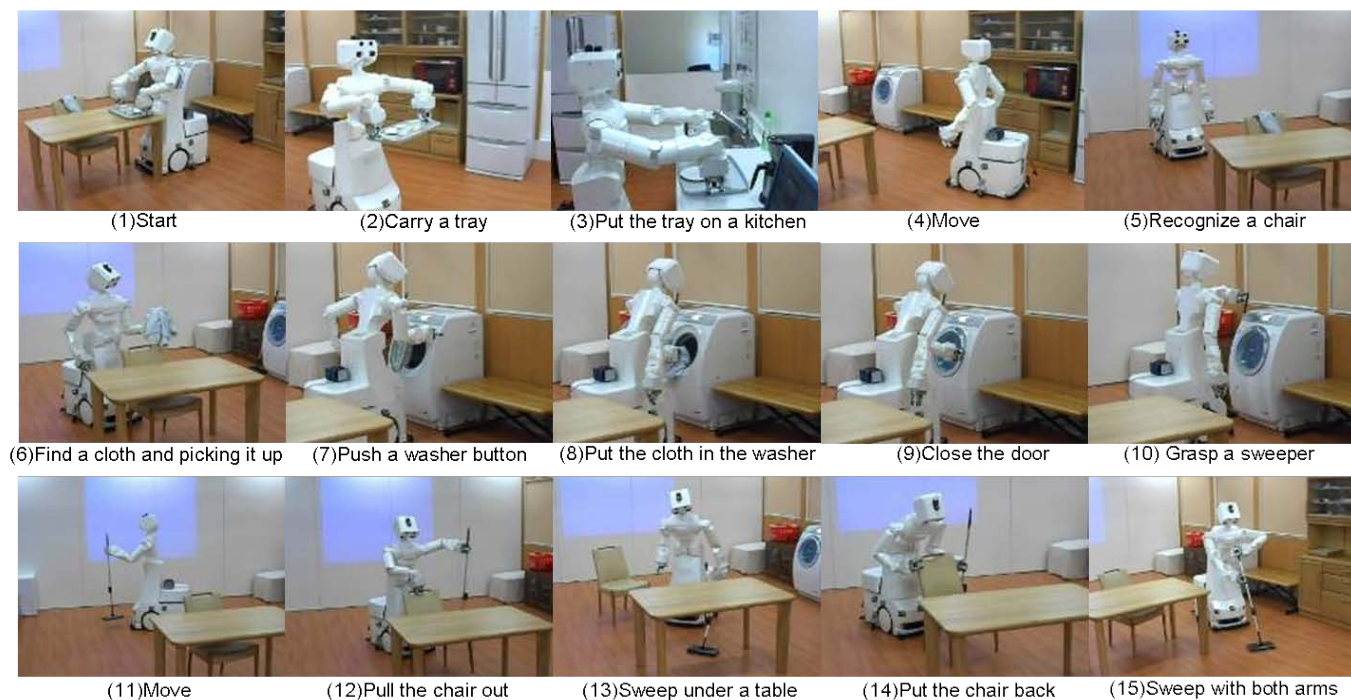|				
(1)Start	(2)Carry a tray	(3)Put the tray on a kitchen	(4)Move	(5)Recognize a chair
(6)Find a cloth and picking it up	(7)Push a washer button	(8)Put the cloth in the washer	(9)Close the door	(10) Grasp a sweeper
(11)Move	(12)Pull the chair out	(13)Sweep under a table	(14)Put the chair back	(15)Sweep with both arms

Fig. 9.   An experiment

layer which runs in parallel with main routine. A detected failure is coped as an error in the system, this layer interrupt main process and insert adequate functions to recover the error.

In addition this layer plays a part to prevent unnecessary iteration of same recovery routine. It manages how many iterations occur in one recovery process, and if the number of iterations is over a predefined threshold, it calls a function to rework the task fundamentally.

VIII. Experiments

A. Settings

Fig. 8 shows an experimental environment. The size of room was $5m \times 8m$. A set of furniture including a table, chairs and shelves was placed on the room, and some home electrical appliances as a frig and a washing machine were also equipped. Tasks imposed to a robot were (i) to carry a tray, (ii) to collect clothes and (iii) to clean a floor.

1) **Carry a tray**: The robot stands a front of a table (described Fig. 8, P) in the beginning, pick the tray up, move to point Q and put it on a kitchen.
2) **Collect a cloth**: The robot moves to the point R, finds a cloth placed on a back of a chair. After picking it up, put the cloth into the washing machine placed on the point S
3) **Sweep a floor**: The robot grabs a broom which is propped up against the washing machine, and move to the point R. it sweeps under the table after removing the chair, and then, moves around the room with sweeping as shown in upper figure at Fig. 8.

An environment map for self localization was generated in advance. A person who moved around the room pushed the robot from behind, scan matching was performed by using LRF data.

B. Implementation of doing chores

Basically, behavior units required to perform these tasks consisted of functions described in section IV, V and VI. As illustrated in Fig. 6, each manipulation task basically forms the following structure: "check" function finds manipulation target and returns relative coordinates between the robot and the target. "plan" function plans the robot motion based on the coordinates and geometrical model of the target, and returns pairs of robot pose stream and its execution time. "do" function executes the pose stream with running monitoring functions in parallel.

The action sequence needed for tidying and cleaning task was divided into 14 behavior units, 2 units for carrying a tray, 5 units for correcting a cloth and remained 7 units for sweeping a floor. Because the sweep task included a sub task to move a chair for cleaning under the table, it needed much behavior units.

In addition to the integrated software system based on geometrical simulator, the behavior unit framework permitted us to design simple programming on task level. Fig. 9 shows the outline of the tidying and cleaning experiment. A task execution took about 8 minutes. If some failures occurred, the time increased depending on its recovery.

C. Examples of failure detection and recovery

Fig. 10 shows an example of failure recovery with washer handling. In this case, the door of the washer tab was not
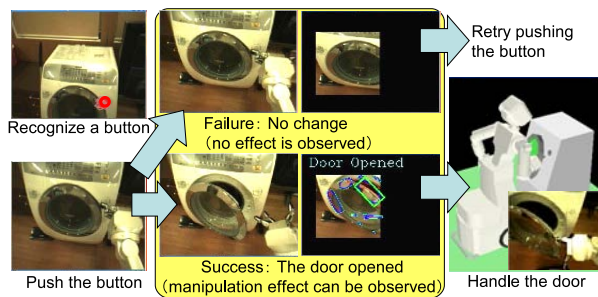
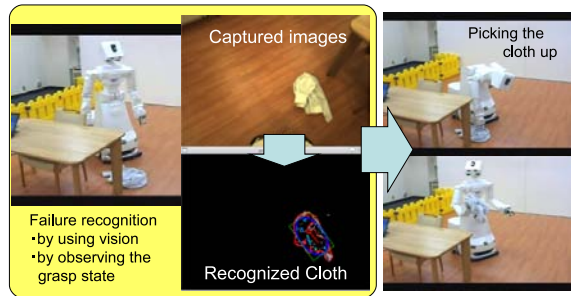Fig. 10. Failure Detection in washer door opening

Fig. 11. Failure Recovery by picking up a cloth

opened because the robot could not push the button on the washer normally. This was a failure classified to (B) described section VII.B. One of the methods to achieve secure manipulation is to observe effects on the target object after or in the middle of the manipulation. Visual functions which judged whether or not the washer door was opened. Whenever the robot found a trouble in the pushing, the task was retried from button recognition stage. If the retries successively failed, error handling layer caught an error and called a behavior unit of approach part. It means that the robot started over the task by changing its wheelbase pose.

Fig. 11 shows another example in the case of cloth handling. The figure indicates the condition that the robot could not pick up a cloth and dropped it down. This can be detected by gazing its hand or observing joint angles of fingers. Because it cannot be recovered only doing same motion again, the failure was classified to (C) described section 5.3. In this case, a novel motion but combining with basic behaviors was applied to find and to pick up the cloth.

IX. CONCLUSIONS

This paper described a software system integration of daily assistive robots. Several tasks related to cleaning and tidying up rooms are focused on. Simple task description was adopted to apply our software system which includes a plenty of recognition and motion generation functions. We also tried to detect and recover some failures based on the system. Experiments doing several daily tasks with handling daily tools showed the effectiveness of our system integration.

Future works, we try to develop more applicable functions to find failures automatically. In addition, it is predicted that automatic behavior unit generation is needed because there were worrying processes to divide a task into behavior units by manual. More feasible motion planner was also needed.

REFERENCES

[1] T. Asfour, et. al. "ARMAR-III: An Integrated Humanoid Plattfrom for Sensory-Motor Control," IEEE-RAS Int'l Conf. on Humanoid Robots, 2006.
[2] Besl, P. J. and McKay, N. D.: "A Method for Registration of 3-D Shapes," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239-256, 1992.
[3] Y. Boykov and V. Kolmogorov: "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," IEEE Trans. Pattern Analysis and Machine Intelligence, 26(9), pp.1124–1137, 2004.
[4] T. F. Chang and R.-V. Dubey: "A weighted least-norm solution based scheme for avoiding joint limits for redundant manipulators," in IEEE Trans. On Robotics and Automation, 11((2):286-292, April 1995.
[5] N.Y. Chong and K. Tanie: "Object Directive Manipulation Through RFID," Proc. Int'l Conf. on Control Automation and Systems, pp.22-25, 2003.
[6] N. Hatao, R. Hanai, K. Yamazaki and M. Inaba: "Real-Time Navigation for a Personal Mobility Robot in Environment with Pedestrians," in 18th IEEE Int'l Symposium on Robot and Human Interactive Communication, pp.619–626, 2009.
[7] H Jang et. al.: "Spatial Reasoning for Real-time Robotic Manipulation," Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robotics and Systems, pp.2632–2637 , 2006.
[8] K.Yokoi et.al: "Experimental Study of Humanoid Robot HRP-1S," The International Journal of Robotics Research, 2004.
[9] R. Katsuki et.al: "Handling of Objects with Marks by a Robot," Proc of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems, pp.130–135, 2003.
[10] F. Lu and E. Milios: "Globally consistent range scan alignment for environment mapping," Autonomous Robots, Vol. 4, No. 4, pp.333–349, 1997.
[11] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions with Singularity Robustness for Robot Manipulator Control," Journal of Dynamic Systems, Measurement, and Control, Vol.108, pp.163-171, 1986.
[12] E. S. Neo et. al.: "Operating Humanoid Robots in Human Environments", Proc. Workshop on Manipulation for Human Environments, Robotics: Science and Systems, 2006.
[13] K. Okada et.al: "Vision based behavior verification system of humanoid robot for daily environment tasks", 6th IEEE-RAS Int'l Conf on Humanoid Robots, pp 7-12, 2006.
[14] K.Okada et. al: "Vision based behavior verification system of humanoid robot for daily environment tasks," 6th IEEE-RAS Intl. Conf. on Humanoid Robots, pp7–12, 2006.
[15] L. Petersson, et. al.: "Systems Integration for Real-World Manipulation Tasks," Proc. of Int'l Conf. on Robotics and Automation, Vol.3, pp.2500–2505, 2002.
[16] T. Takahama, et.al.: "Motion Planning for Dual-arm Mobile Manipulator –Realization of "Tidying a Room Motion" –," Proc. of Int'l Conf. on Robotics and Automation, pp.4338-4343, 2004.
[17] K. Yamazaki and M. Inaba: "A Cloth Detection Method Based on Image Wrinkle Feature for a Daily Assistive Robots," IAPR Conf. on Machine Vision Applications, 2009. (to appear)
[18] K. Yamazaki, et.al.: "Modeling and Motion Planning for Handling Furniture by a Mobile Manipulator," Proc. of IEEE Int'l Conf. on Intelligent Robots and Systems, pp.1926–1931, 2007.
[19] http://twendyone.com/index.html