MOPS, a system for mail distribution in office type buildings

S. J. Vestli, N. Tschichold-Gürman
Institute of Robotics, Swiss Federal Institute of Technology, CH-8092 Zürich

1. Introduction

Over the past years, there have been significant developments with respect to the infrastructure of office and industrial environments. Various flexible electronic services are available. Documents on a computer can be shared amongst users, electronic mail can be sent all around the world and telephone conferencing is available. Some services have not been modernised to the same extent. For example the distributional methods of goods within buildings have remained the same for some time. At present, fixed installed distribution systems (e.g.: tube post) are readily available. These systems are difficult to reconfigure as the use of the building changes. Also the shape and size of the goods to be delivered are limited and when a failure occurs rectifying the system can be difficult.

Consequently, there are several research and development teams worldwide who have identified mobile post systems and other service robots as focus areas, please refer to [3], [8] and [9].

As a benchmark problem for the mobile robot project, we have chosen the task of mail distribution within a new building at the ETH, Zürich. To realise this mobile robot application, we have addressed several fundamental mobile robot research problems, the practical results of our work are described in the rest of this contribution. One very important issue of this robot project is obviously the interaction with humans, this has been treated within the Man-Technique-Organisation concept developed at the ETH which ensures that the technical system is properly integrated into the comple socio-technical system. This aspect of the project will not be discussed further in this contribution, the interested reader may refer to [12].

2. Task Description

In an institute building of the Swiss Federal Institute of Technology in Zürich (ETH) there are several separate departments each requiring separate internal mail distribution. Currently, mail distribution for the various departments at the ETH is done manually.

The departments are spread on 5 floors within the building and each department has a secretariat located more or less centrally to that particular department. Twice a day the mail arrives in a mail room centrally in the building and the mail for each institute is put into pigeon-holes, one for each department. At this central mail room there are also pigeon-holes for outgoing mail. Currently, somebody from each institute then picks up the institute's mail from the mail room and also deposits the institute's outgoing mail in the pigeon-hole for outgoing mail. The robot's task is to relieve the members of the institute of the daily "pilgrimage" to the central mail room.

Each institute has in its pigeon-hole a box in which the mail is stored. These boxes are loosely guided on rails and they can be gripped by a relatively simple mechanism. Next to the secretariat of each institute a smaller "mail-stop" is located with the same guide rails as in the main mail room. The robot can thus pick up the boxes at any of these mail rooms and deposit at one of the other mail rooms. Since this is a public building with unlimited access, protection against theft of the mail must be provided. This has been realised with motorised blinds for the pigeon-holes, blinds which can be opened by the robot, the secretaries and the postal service.

The building is approximately 70 m long and 15 m wide, the various mail stops are distributed on 5 floors.

3. Robot System

For the task described above a non-holonomic mobile robot platform with two diametrically opposed drive wheels and a free-wheeling castor (the same kinematics as the robot Newt, as described in [6]) is being used, see figure 1. The two drive wheels are located at one end (the front end) of the vehicle, and the support castor at the other end.

The basic platform contains a multi-processing controller for up to 6 servo motors (DC or EC) based around a VME bus . The drive wheels constitutes two servo axis, and the remaining 4 axis are consequently available for a manipulating mechanism. Such a mechanism has been integrated on top of the platform. Currently the robot is equipped with two 68040 single board computers which are used to control the six servo axis and to implement the application. The programming is done in the object-oriented real-time development environment D'nia [2].

The basic robot platform is equipped with two scanned laser range-finders (one at the front, the other at the rear), twelve ultrasonic range finders, and a bumper system with six segments. A radio modem enables wireless communication and various indicator lights and speech output module informs the environment about the robots immediate action. An infra-red communication system is used to communicate with the building infra-structure (elevator, pigeon-hole blinds, doors, etc.).

4. Navigation and Control

Motion planning is one of the most important problems in the realization of autonomous robots. At a first glance, motion planning looks relatively simple. The main reason for this attitude is the triviality of this task for humans. It is very easy for us to move or to reach an intended object within our environment. However, the operative intelligence used by humans for motion planning is very difficult to simulate in a computer program.

In this section a hierarchic behaviour based navigation system for mobile robots is introduced, which has been implemented for MOPS (figure 3). After a task is defined by the user, e.g. drive to the central mail room, the global planner generates a list of the intermediate positions that have to be driven to reach the final goal. This plan is calculated using the map of the robot's operation environment (see 4.1 for more details). The navigation behaviour of the robot on each of these path segments is then selected by a situation based behaviour selector (SBBS), which is discussed in 4.2. The different behaviours are implemented independently. Most of these behaviours are based on a position controller, which is discussed in section 4.3.

4.1 Global Planner

When the operating environment of the robot is known and represented in a graph structure, global planning can be realized using a graph search algorithm. Therefore, the operation environment of the MOPS is stored in a generic, directed graph. Each graph node contains the world coordinates of that point and the walls that can be seen by the robot at that position. A simplified pictorial example of this graph structure for one floor in our building can be found in figure 2.

Further information can also be stored in the graph edges which are associated to path segments. This allows us to define the behaviour as well as the reactions of the robot while traversing a particular segment. Thus the possible algorithms to be applied on a single path segment are defined by the user.

Operating on this graph, the global planner generates a list of intermediate path segments to be visited to reach a goal position. The local planner takes the next path segment from the list and tries to reach it using the information associated to this list element. An overview of this structure, including the SBBS module as the local planner, is depicted in figure 3.

For the planning of the robot's travel route, several graph search algorithms can be used. The graph representing an office building as described above is sparse (usually two, maximal four edges per node) and small (a few thousands nodes), therefore the breadth search algorithm [10] can be used, which always delivers optimal routes and it is efficient enough to be computed in real-time. It is also possible to use a weighted graph, to mark edges which are difficult to travel (e.g. routes that are usually crowded). In this case, a Priority-First Search algorithm can be used [10]. The use of a weighted graph is recommended in buildings with more than one corridor, enabling different routes to reach a location. In the case of MOPS, a weighted graph was implemented with the Priority-First Search algorithm to have a solution which can also be used in more complicated buildings.

4.2 A Situation Based Behaviour Selector

As mentioned in the previous section, the global planner generates a list of the path segments to be traversed by the robot to reach a predefined goal. The situation based behaviour selector is responsible for scheduling the appropriate navigation algorithms as indicated by the information contained in the path segment data packet and the current sensor data (figure 4).

The standard case is that the subgoals defined by the global planner can be reached on simple trajectories, often straight lines. Thus the robot must be able to identify its position in the world coordinate system and update it continuously (see section 5). It is then an easy task to travel on the desired trajectory using classical control techniques, e.g. a position controller (see section 4.3). However, in a real world environment the path defined by the global planner may be impassable at certain times, e.g. it has to drive around an obstacle. Thus the robot has to recognize the situation and react properly.

The SBBS utilizes two dimensional range data stemming from two laser range sensors (figure 1) in order to recognize situations for the mobile robot. A situation is defined as a recognized scene within an environment, if measurements of the sensors at a pre-defined position of the robot correspond to a set of measurements memorized before. Recognized situations are then used as a basis to provide a certain behaviour (a particular algorithm competent in handling the situation) for the mobile robot (figure 4). It is also possible to use a neural network (e.g. RuleNet) as a behaviour scheduler [11]; in the realized system on the MOPS, a rule based system has been used as the behaviour scheduler, because the number of possible behaviours in a situation is small, therefore the SBBS results in a small rule-base that can easily be defined by a programmer. The behaviours can be implemented using different algorithms, including neural, fuzzy and classical control techniques. Some of the standard and reflex behaviours are depicted in figure 4.

The main advantage of choosing a modular control architecture like the SBBS is the possibility of an easy extension of the robot system. As the implementation of the behaviour modules are independent of each other, it is possible to change or improve a particular one without affecting the others. It is also easy to add new behaviour modules to the system; only the behaviour selector module has to be extended and the user can introduce this new behaviour in the path segments of the global planner. All the existing behaviour modules are not influenced by this extension. On the other hand, if one single module would have to handle several behaviours, changing one behaviour could have an uncontrollable influence on other behaviours.

4.3 Position Controller

In the previous sections we have described how the MOPS performs global and local navigation. One step of the navigation involves generating a sequence of points (x, y and θ) that should be visited, one after the other, by the robot (see global planner in 4.1). To enable MOPS to execute such trajectories composed of a number of points a *position controller* has been developed.

The position controller employed for the MOPS project is an improvement of the controller described in [1]. Due to the construction of the manipulator mechanism and the guide rails in the pigeon holes, the position controller must have a very high accuracy. Therefore, the position controller has been implemented so as to be able to reach a given goal point within an accuracy of less than 1 mm (x, y) and 0.1 degree (θ). Further details on this position controller can be found in [4], [5] and [14]. This controller ensures that the MOPS follows the planned trajectories, and avoidance manoeuvres, and its accuracy is high enough to enable reliable docking by the pigeon holes. A typical trajectory for this controller when executing a sideways translation of 1 m is visualised in figure 5.

Positioning the robot by the pigeon holes so that the mail-boxes can be manipulated is achieved using the above mentioned position controller in co-operation with position update methods (see section “Position Update”), and through the use of special behaviours for the via-points involved in the docking. In figure 8 this process is visualised. A photo of the robot unloading a box is also shown in figure 10.

5. Position Update

Unless specific measures are taken a mobile robot does not automatically have any knowledge of its own position. A commonly used method for calculating the position is to measure the rotation of the wheels and from this deduce the position through the usage of the robot’s kinematic equations, a method known as odometry or dead reckoning. This method is always incremental, an initial estimate is used and the relative motion over a short time interval is added. Due to this incremental nature it is inevitable that even small systematic errors such as inaccurate knowledge of the wheel radii will over time add together and take on unacceptable proportions. This problem is similar to the problem of rowing across a lake from your boat house to a colleague on the other side in thick fog. You know your start point (your boat house) and how long and in what direction you have to row to reach your colleague. But if you try you will probably fail, because you don’t know exactly how many metres you row per stroke, and maybe your compass had a small offset in it.

A new position update method had to be developed that provides: real time performance; high accuracy and robustness. We would like to impose a maximum time limit for the calculations for a position update of 1 second. The position update system should have a reputability of 10 mm. And it should be possible to wilfully “disturb” the robot without any significant decrease in the accuracy (reputability) of the estimated position.

The method we propose is based on matching geometric structures in the sensor data to modelled geometric structures. A central part of this method is thus to find geometric structures in the sensor data. We propose the following method: fit a straight-line, cylinder, corner, etc. to the data points that minimises the sum of the squared distances from the points to the structure. As it is unlikely that all the separate measurements (points) in a whole range-finder scan will fall on the same straight structure only a subset of the scan is considered for the least squares fitting. This subset of o measurement points is “slided” along the complete set of n measurement points in the scan, in increments of one, until all the points have been processed.

This process can be viewed as a transformation of the range data to equally many points in the parameter space of the results from the structure fitting, in the case of the straight line r (the perpendicular distance to the LSQ line and a) and α (the angle between the x-axis the perpendicular). In this rα-space “clusters” of line fitting solutions will form, these clusters indicate that many sub-sets of measurement points were lying on a structure which was straight (or they were lying on co-linear straight structures). In general a cluster is a collection of observations (in this case rα values) that lie “close” together and represent some underlying common feature (in our case a straight line), i.e. all these observations belong to the same “class”. The clusters are in other words evidence for the presence of correspondingly many walls in the environment. The clusters contain the co-ordinates of the walls in the robot co-ordinate system {R}, and hence contain all the information we need to calculate the robots position.

Consider the example scan and environment of figure 7. The odometrically estimated position for this scan was $x = 1.2$, $y = 1.2$ and $\theta = 0.101$, the two modelled lines have parameters $r_1 = 0.1$, $\alpha_1 = 0.0$, $r_2 = 0.1$, $\alpha_2 = 1.571$. After preprocessing and least squares line fitting we have the rα-space of figure 6. The correct position $x = 1.13$, $y = 1.12$ and $\theta = 0.0204$, can then easily be found. The accuracy of this approach is 3 mm and the processing time is less than 0.5 of a second for a single pass, please refer to [13] for more details.

6. Summary

This paper has described: the robot system, the navigation and control architecture, the position controller, and the localisation method that are in use on the MOPS for its task of distributing the mail at the Swiss Federal Institute of Technology, Zürich.

We have shown that the system has a highly accurate localisation and position control system, which in turn enables the robot to "dock onto" the pigeon holes and to load/unload boxes of mail. Furthermore the position update process is tolerant towards unmodelled obstacles and the position update is also fast enough for utilisation in the real world.

The hybrid navigation scheme combining graph-searching with a situation based behaviour selectors and appropriate behaviours has proved itself as robust and efficient. The generated paths are close to being optimal, and the tolerance towards obstacles is high.

The MOPS currently serves the whole building (70 by 15 metres) on five different floors using the existing elevator to change floors. In figure 9 the first 200 metres of a run is visualised together with the modelled environment. A rack is located approximately at x=5 and y =18.5, the doors connecting the main corridor and the room in which the rack is located are 1.1 m and the robot is 0.6 m wide. This only leaves 25 cm each side of the robot for entering the room and unloading/loading boxes (ignoring additional complication due to the asymmetric robot construction). Nevertheless, with the methods described in this paper no problems has been encountered during operation.

6.1 Future work

In the near future we plan to integrate the MOPS with the computer infrastructure of the building. This will enable the MOPS to communicate via internet protocol to any other internet capable computer in the building (and world-wide for that matter). With this scheme it will be possible for all users of the system (secretaries and postal-service) to query the status of the system or to issue simple commands. In order to ensure that the interface is user-friendly it will be implemented as pages on the "world wide web".

Currently the system relies upon an a-priori map. Investigations are being undertaken with the goal of eliminating this costly, and unnecessary step.

Another issue receiving our attention is the extroperceptive sensor configuration. In the last few months new laser range-finders has become available. One such system is currently being tested, and will at a later stage be integrated in order to enhance the navigation and localisation of the MOPS.

The use of a vision system for the recognition of "important" scenes, e.g. for reading the room numbers, is also being examined.

7. References

[1] Astolfi A., "Exponential stabilization of a mobile robot" 3.rd European Control conference, Rome, Italy, 1995.

[2] Diez D., Vestli S. J., "D'nia an object oriented real time system", Real-Time Magazine, Real-Time Consult Publication, rue de la Justice 23, 1070 Brussels, Belgium, August, 1995.

[3] Evans J. M., "HelpMate: a service robot success story", Service Robot: An International Journal, Vol. 1 No. 1, pp. 19-21, MCB University Press, Bradford England, 1985.

[4] Kaiser O., Pfiffner R., Vestli S., Astolfi A., "Positionsregelung für nichtholonome mobile Roboter" 11. Fachgespräche über autonome mobile Systeme, Karlsruhe, Germany, Nov. 1995 (in German).

[5] Meyer M., Suppiger M., "Positionsregelung des mobilen Postroboters", Diploma Thesis, Institute of Robotics, Swiss Federal Institute of Technology, 1996 (in German).

[6] Muir P. F., Neuman C. P., "Kinematic Modelling of Wheeled Mobile Robots", Carnegie Mellon University, the Robotics Institute, Technical Report, VMU-RI-TR-86-12, 1986.

[7] N.N., "mecos Mobile Robot, Differential Drive (MM-DD)", commercial information, mecos Robotics AG, Technopark Zürich, 8005 Zürich, Switzerland, 1995.

[8] Pellerin C., "The service robot market", Service Robot: An International Journal, Vol. 1 No. 3, pp. 17-20, MCB University Press, Bradford England, 1985.

[9] Schoefield M., "Cleaning robots", Service Robot: An International Journal, Vol. 1 No. 3, pp. 11-16, MCB University Press, Bradford England, 1985.

[10] Sedgewick R., "Algorithms", Addison-Wesley, 1988.

[11] Tschichold N., "RuleNet - A Knowledge-Based Neural Network Model with Application Examples in Mobile Robotics", Ph. D. Thesis Number 11356, Swiss Federal Institute of Technology, Zürich, 1995.

[12] Scweitzer G., "Mechatronics for the Design of Human Oriented Machines", to appear in: IEEE/ASME Transactions on Mechatronics.

[13] Vestli S. J., "Fast, accurate and robust estimation of mobile robot position and orientation", Ph. D. Thesis Number 11360, Swiss Federal Institute of Technology, Zürich, 1995.

[14] Vestli S. J., Tschichold N., "MOPS, a system for mail distribution in office type buildings", Submitted to IROS 1996.

Figure 1: The MObile Post System MOPS. The upper part of the system constitutes a 3-DOF manipulator / storage mechanism for two mail boxes. The lower part is a MECOS [7] mobile robot platform equipped with two laser scanners, 12 ultrasound sensors, a tactile bumper, infrared communication system, wireless modem, speech output system and a VME-based multiprocessing robot controller for 6 servo axis (2 used for propulsion, 3 for manipulation, 1 spare).

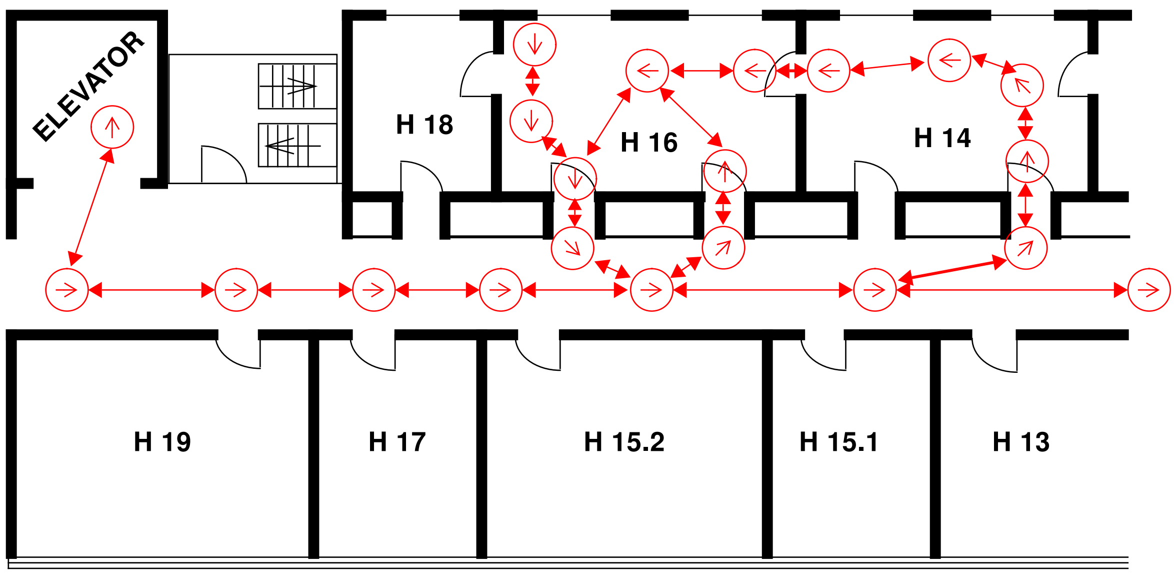

Figure 2: Schematic view of the robot's map of the building.

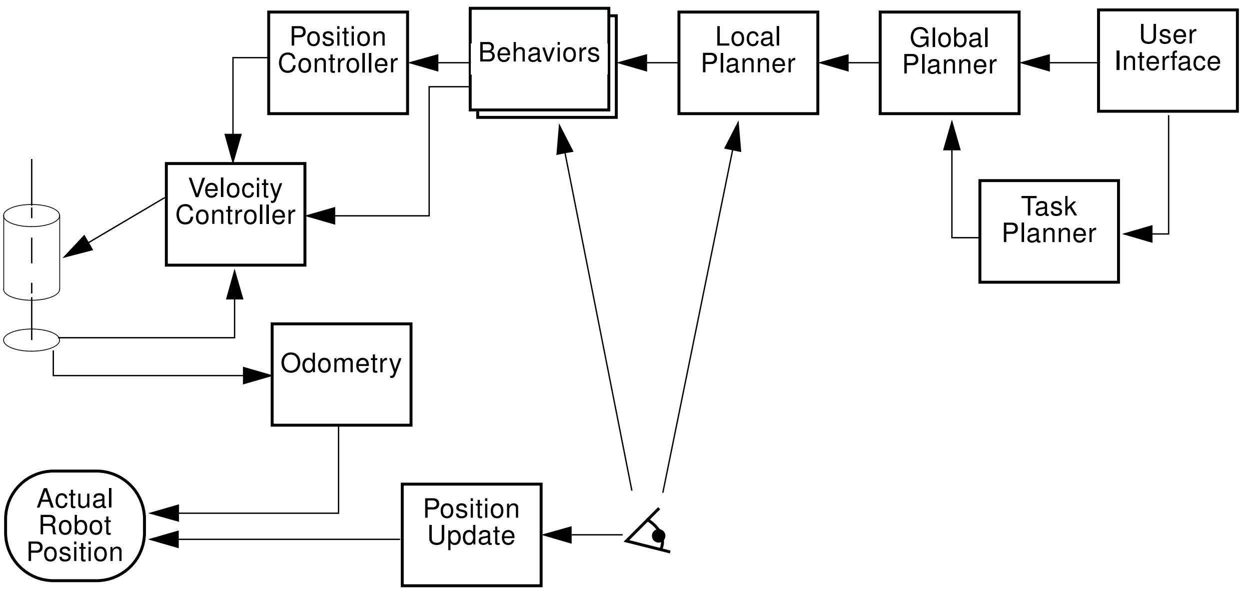

Figure 3: Control structure and information flow in the autonomous mobile robot system MOPS.

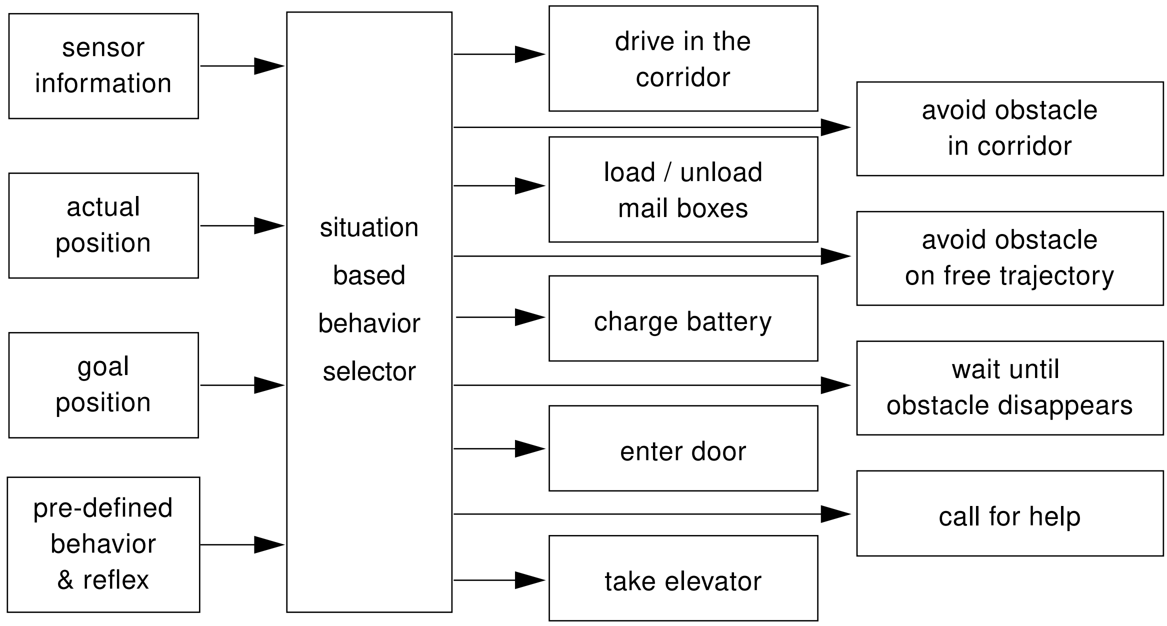

Figure 4: The situation based behaviour selector, associated behaviours on the right and input information on the left.

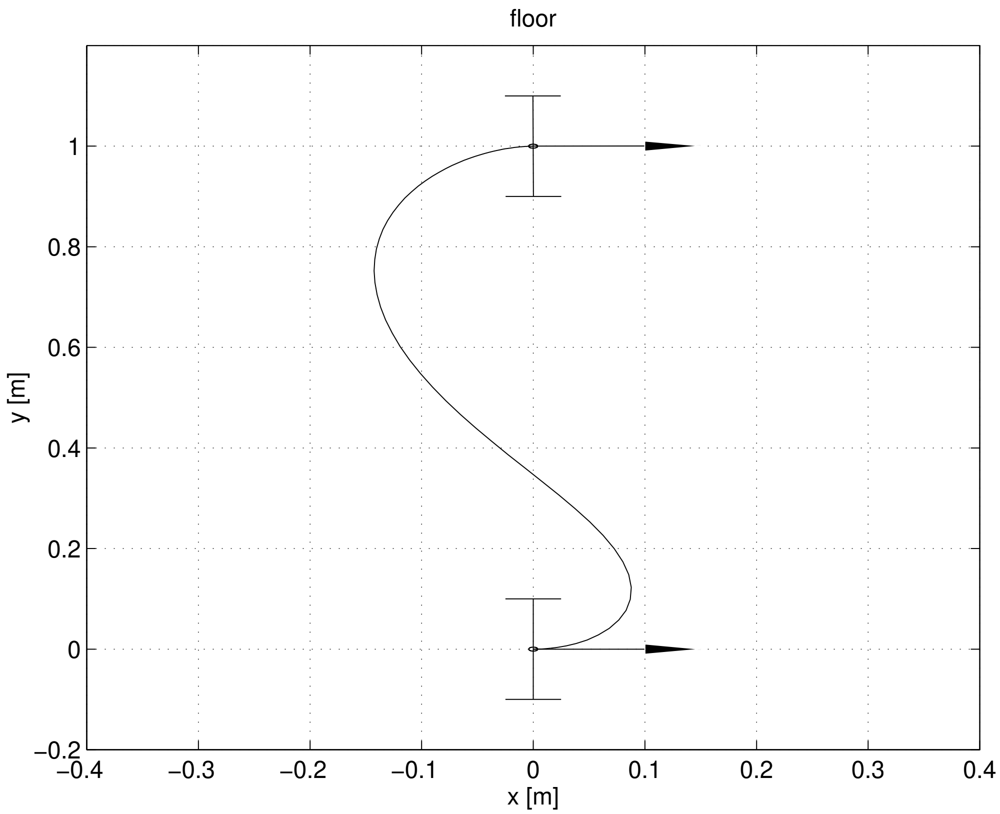

Figure 5: Trace of the robot x and y world co-ordinates for a one metre translation in (robot) y direction.

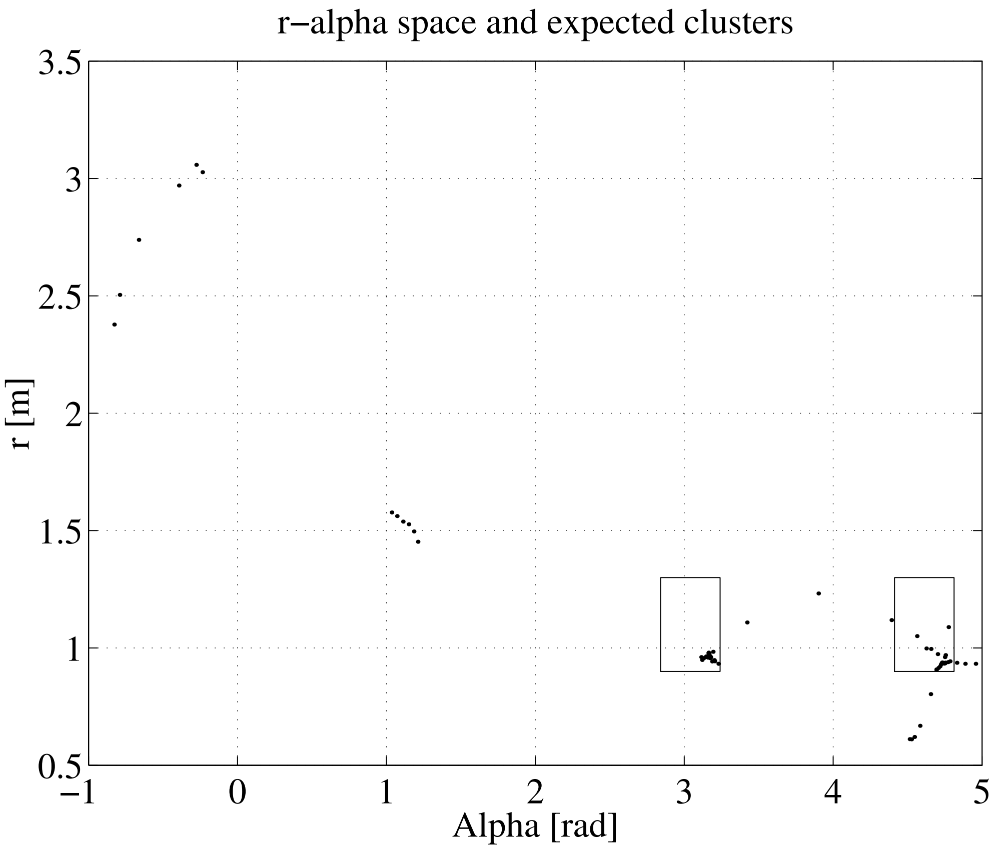

Figure 6: rα-space of the sensor data in figure 7 (dots).

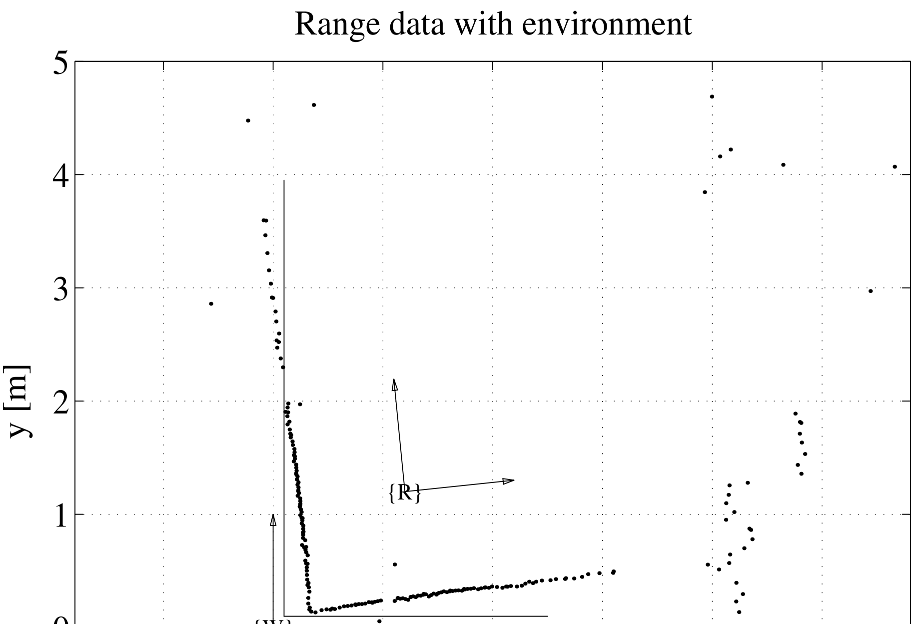

Figure 7: Example of raw sensor data (points) together with the modelled environment (continuous lines).

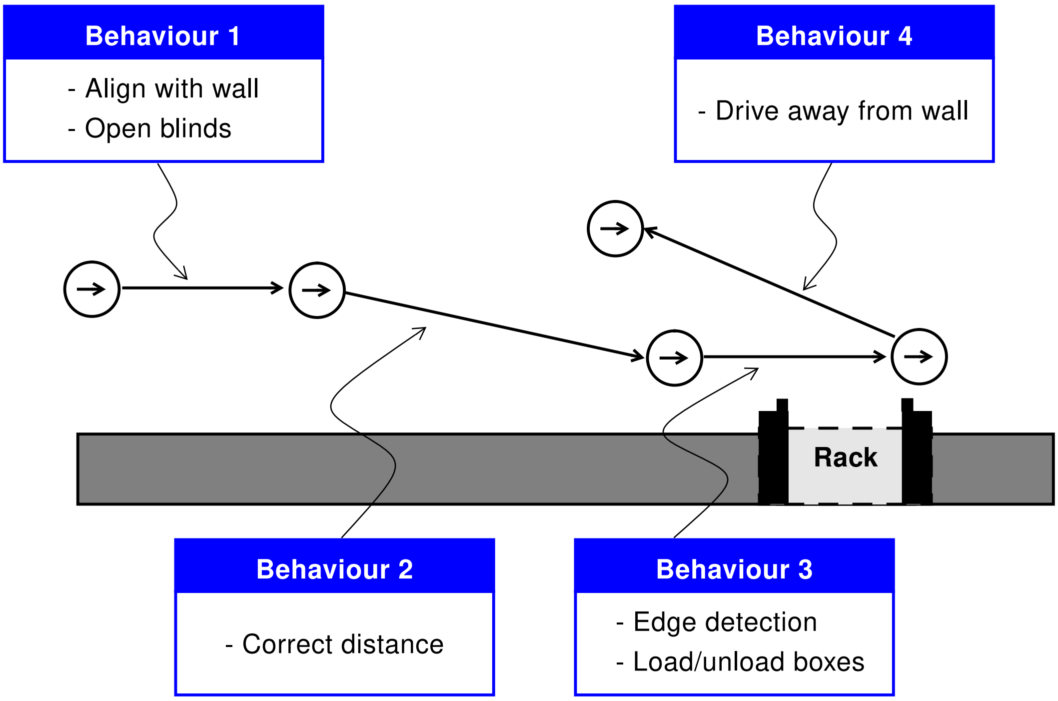

Figure 8: Schematic view of the docking process.

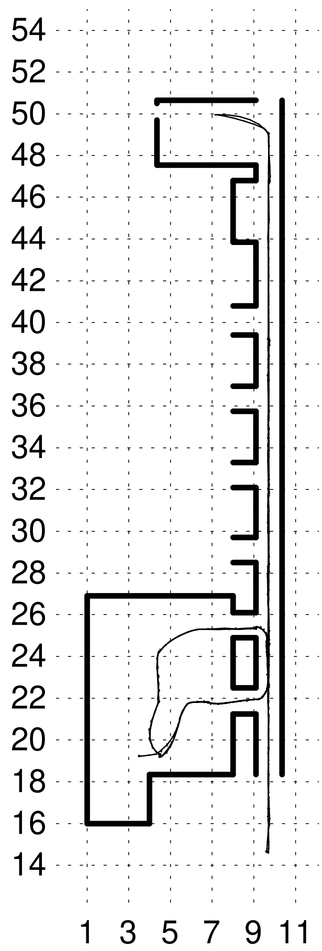

Figure 9: The xy trace of a run in our building, co-ordinate in metres.

Figure 10: MOPS by unloading a box of mail.