

Control and Localisation of a Post Distributing Mobile Robot

M. Adams, N. Tschichold-Gürman, R. Müller, S. Neogy, L. Ruf, S. Vestli, D. von Flüe
Institute of Robotics,
Swiss Federal Institute of Technology
Zürich,
Switzerland

Abstract—

Mobile robotics has been labelled as a subject which has promised so much but produced so little. In response to this we present here various research and development issues which we are addressing in order to realise an internal post distributing mobile robot. To realise this application we have implemented a behavioural control system which is capable of recognising scenes within a known environment. The behaviours are defined as sensor data fueled algorithms capable of dealing with both expected and unexpected situations.

An algorithm for localising the mobile robot in the presence of uncertain odometric information is also presented. By tracking predicted features from the environment with a sonar range sensor, we show that the position of the vehicle can be accurately estimated whilst simultaneously building a map of the environment.

We also present some mechanical concepts which we have considered for the distribution of post itself within an indoor environment.

I. INTRODUCTION

We present here various issues which we are addressing in order to develop a mobile robot that can be viewed as part of the logistical services of large buildings such as office blocks, hospitals or factory floors. The term logistical services is used here in the context of distribution, cleaning and similar services.

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. In this building there will be several separate departments each requiring separate internal mail distribution. Currently, mail distribution for the various departments at the ETH is done manually. The project has the potential to save many man hours in terms of collecting and distributing post.

To realise this mobile robot application, we are currently addressing several fundamental mobile robot re-

search problems. We describe here three particular areas that we have addressed, namely the overall control structure of the mobile system; a trajectory following control algorithm which generates smooth paths for the mobile robot to follow and an algorithm which at present manipulates ultrasonic range data in order to track environmental features (corners, walls) in order to update the position of the mobile robot.

In section II. we present a mobile robot control system which manipulates two dimensional range data (ultrasonic and optical [12]) in order to recognise “situations” for the mobile robot. We define a situation as a recognised scene within an environment, meaning that some kind of preplanned action can be taken. Recognised situations are then used as a basis to provide a certain “behaviour” (a particular algorithm competent in handling the “situation”) for the mobile robot, under a neural network decision making algorithm. Results using this technique in indoor environments are shown in the form of planned paths of the mobile robot.

Planned paths provided by the decision making algorithm are passed to the “trajectory following controller”, results of which are shown in section III. We show that the steering angle and front wheel velocity of the mobile robot can be modelled as a non-linear feedback controller. By analysing the motion of the robot under such a non-linear control scheme, smooth trajectories can be achieved.

As the mobile robot follows its desired trajectories, it is essential that it can update its position reliably and accurately. In section IV. we offer a solution to this *localisation* problem which, at present, manipulates two dimensional ultrasonic range data. The algorithm extracts certain features from an indoor environment (corners, walls) and keeps track of them as the mobile robot moves. We show in this section that positional errors caused by odometric estimates can be compensated for as the algorithm generates an optimal state estimate of the robot's position, after each scan of the environment, and further quantifies

the probabilistic certainty of this estimate.

Finally in section V. we show some of the mechanical considerations which we are using in order to realise our particular application. To deliver internal post as efficiently as possible, the mobile robot needs to be able to receive, transport and deliver as much post as possible for the given size of the robot. In this section, a mechanism is presented for achieving this aim.

II. MOBILE ROBOT CONTROL STRUCTURE

When the operating environment of the robot is known, global planning can be realised using a graph search algorithm [4]. Most of these graphs contain only information on specific positions in the building [7, 9, 10].

A graph search algorithm, such as A*, selects the nodes to be visited and the robot traverses them using a navigation algorithm. We propose to use available, *a priori* information about the building as much as possible including information which can be delivered by the user. For this purpose we interpret the graph nodes as path segments and save all available information which characterise the path segments in the graph nodes. This allows us to define the behaviour as well as the reactions of the robot while traversing this segment. Thus the possible algorithms to be applied on a single path segment are defined by the user. Operating on such a graph, the global planner generates a list of intermediate “path segments” to be visited to reach a goal state. The local planner takes the next “path segment” from the list and tries to reach it using the information associated with this list element. An overview of this structure including the local planner is shown in figure 1.

A path segment contains its start and end coordinates, the standard behaviour on the path and the reflex behaviour for exceptional cases (obstacles). The positional information can be defined in absolute world coordinates. It can also be a sensor defined position, this means, some sensor data has to be detected at the specified position (e.g. information from passive sensors (see section IV, or a particular scene to be detected from the optical system). It is also possible to use a combination of these. The standard behaviours will be:

- Following the right wall.
- Following the left wall.
- Driving in the middle of the corridor.
- Driving “blind” along a virtual trajectory (useful in larger rooms where little sensor data is available).
- Passing through a door or entering an office.

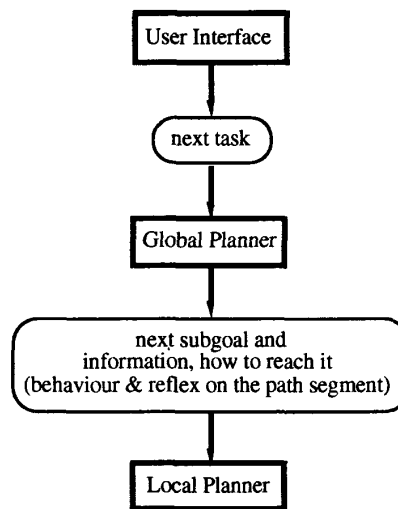


Figure 1: Control structure of the mail distributing robot

- Exact positioning (e.g. in front of the post-station to get the post).
- Loading / unloading the post.
- Using the elevator.

The reflex behaviours are needed to select an algorithm in case of exceptions such as obstacles in the path. Four of the main reflex behaviours are:

- Trying to avoid an obstacle.
- Not trying to avoid an obstacle (e.g. in a narrow corridor);
- Just wait until the obstacle disappears (only possible for moving objects).
- Use an alternative path.

The appropriate standard or reflex behaviour is selected by the situation based behaviour selector, this process is shown in figure 2.

A. Situation based behaviour selector

The situation based behaviour selector is responsible for scheduling the appropriate controllers as indicated by the information contained in the path segment data packet and the current sensor data. A first prototype of the behaviour selector has been realised with a neural network based decision mechanism known as “RuleNet” [11]. For

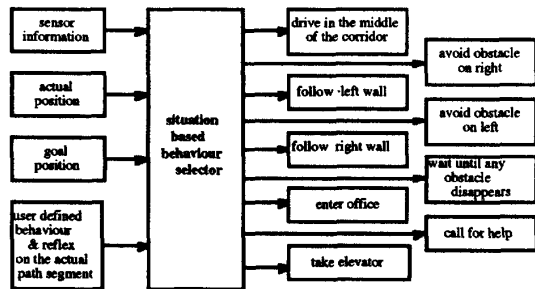


Figure 2: The situation based behaviour selector, associated behaviours on the right and input information on the left.

this purpose, the network has been trained with a subset of the laser scanner range data as input (9 inputs) and the appropriate behaviour as desired output. The information from the global planner about the behaviour and the reflex for the path segments were ignored. After training the network with 659 randomly generated examples it was tested on 346 newly generated examples. The network was able to recognise all training examples correctly. On the tested examples its success rate was 89.6%. The network was therefore able to recognise most of the situations correctly and select the appropriate controller.

An analysis of the training examples has shown that the distinction of the situations based only on features stemming from sensor data is very difficult as only small changes in the input data lead to significant changes in the outputs. This makes a classification based on the approximation of the pattern clusters with geometrical primitives very difficult. Therefore preprocessing the sensor data to produce useful qualitative information is necessary to achieve better results. The controllers of the behaviours take the sensor information and the path segment data packet to navigate the mobile robot. The sensor information is delivered from the sensor data interpretation module. Separate preprocessing of the sensor data makes the navigation algorithm independent of the robot's sensors. For example a controller for wall following needs the distance to the wall and the orientation of the robot to the wall. This reduced information leads to efficient controller algorithms because the relevant information is reduced to two values and is directly accessible. The preprocessing of the sensor data usually has the effect that similar inputs demand similar navigation steps, so any classification algorithms, including neural networks and fuzzy control algorithms can be used efficiently. Another advantage of the preprocessing is that the algorithm can

be used on different robots with different sensors. The only part that must be changed is the pre-processing of the raw sensor data to extract the relevant information for the navigation algorithm. As the controllers are totally independent of each other it is possible to change or optimise a particular behaviour 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 has to handle several behaviours, optimising one behaviour often has an uncontrollable influence on other behaviours.

The behaviours are implemented as separate processes within the multitasking real-time system XOboron [2] using different control methods such as neural, fuzzy, and classical control techniques. An example run of the mobile robot controlled by the above described control scheme is illustrated in figure 3.

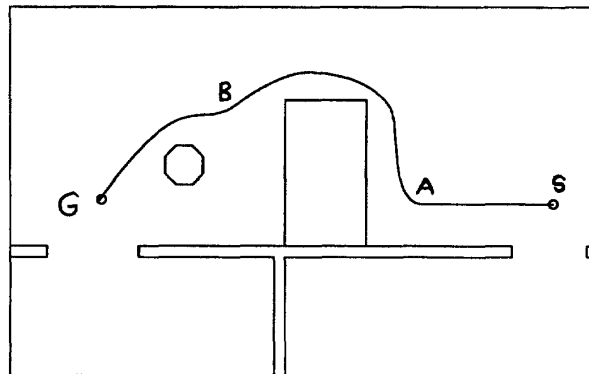


Figure 3: Performance of the robot control scheme. Between the start point S and point A and between B and G the behaviour "trajectory control" was used. Between A and B the reflex behaviour "obstacle avoidance" was used.

III. TRAJECTORY FOLLOWING CONTROL

As described above it is also desirable to follow virtual trajectories, for instance when moving across a large open space. We have implemented a behaviour for following virtual trajectories as a non linear feedback controller. Consider the robot relative to the path segment as in figure 4 below.

The coordinates of the new position is provided by the global planner along with the information that this behaviour should be utilised. The virtual trajectory is a straight line from the last co-ordinate visited (old position) to the next coordinate to be achieved (next position). The goal of the behaviour is to keep the robot on

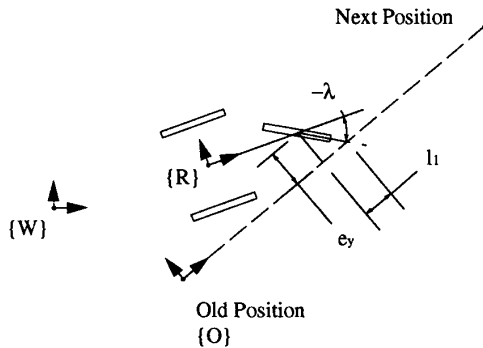


Figure 4: Mobile robot parameters shown relative to its virtual trajectory

this trajectory. The behaviour is then implemented as a feedback control law according to equations 1 and 2.

$$\lambda = -\tan^{-1}\left(\frac{e_y}{l_1}\right) \quad (1)$$

$$\omega = k_1 \quad (2)$$

Where λ is the steering angle and ω is the front wheel velocity. The steering angle and front wheel velocity are then achieved using two separate single input single output (SISO) controllers, the performance of which are described in [12]. A block diagram showing the operation of the control system is shown in figure 5.

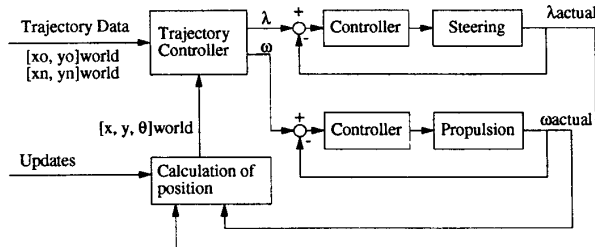


Figure 5: Trajectory controller block diagram showing inner steering angle and wheel velocity controllers.

The parameters for a stable controller can be chosen and an actual run with the mobile robot utilising such a controller is depicted in figure 6. The two solid lines show the desired trajectory, as output from the situation based behaviour selector. The control system in figure 5 produces the smooth path shown by the dashed line,

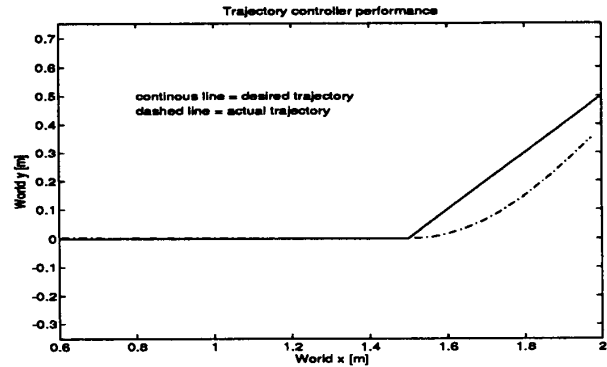


Figure 6: Performance of the controller implementing equations 1 and 2 when moving onto a new path segment with a 45° rotation relative to the last segment.

which brings the mobile robot back on to track in a smooth curve. Note that the maximum deviation from the desired path in this case is approximately 15 cm.

IV. LOCALISATION OF THE MOBILE ROBOT

As the mobile robot follows its given trajectories, it is essential for accurate navigation that it can update its position at all times. The problems of relying solely upon the odometry are well documented [8] and we will show here that wheel slippage, uneven floor surfaces and inaccurate calibration cause the positional uncertainty to grow indefinitely when using only odometry.

Following the work by Leonard [6] we describe here our results of using an approach to mobile robot navigation that unifies the problems of obstacle detection, position estimation and map building in a common multi-target tracking framework.

As the mobile robot moves it continuously tracks naturally occurring indoor targets or "beacons". Predicted targets (found from the known environmental map) are tracked in order to update the position of the vehicle. Newly observed targets (those *not* predicted) are caused by unknown environmental features or obstacles from which new tracks are initiated, classified and eventually integrated into the map.

We have implemented the above technique, at present using real sonar data. We note that a good sensor model is crucial for this work and in order to predict our expected observations (from the sonar data) we use the sonar model presented by Kuc and Siegel [3].

Figure 7 shows *regions of constant depth* (RCD's) which have been extracted from 15 sonar scans recorded from each cross (x) [5]. The model from Leonard's work suggests that RCD's such as those recorded at the positions marked A in the figure correspond to planar surfaces;



Figure 7: Regions of constant depth (RCD's) extracted from 15 sonar range scans. Note the motion of the RCD's at plane surfaces (marked as A) and corners (marked as B).

RCD's marked B rotate about a point corresponding to a 90° corner and RCD's such as C which cannot be matched correspond to multiple reflections of the ultrasonic wave.

A. Position Updating

The position of the mobile robot is estimated under the framework of an extended Kalman filter algorithm [1]. The algorithm requires two models:

1. A *plant* model which describes how the vehicle's position changes with discrete time in response to a control input (from the trajectory controller) and a random noise disturbance.
2. A *measurement* model which mathematically expresses a sensor measurement in terms of the vehicle's position and the geometry of the beacon being observed.

The aim of the algorithm is to continually produce estimates of the position of the robot (the estimated states of the Kalman filter) and the associated variances (the error

covariance matrix of the Kalman filter) based upon the previous estimates, the control input and the new beacon observations. The mathematical model for this algorithm is given in [6].

Figure 8 shows the results from the Kalman filtering algorithm using the single corner beacon marked 'C' in figure 7. The vertical crosses (+) show the true posi-

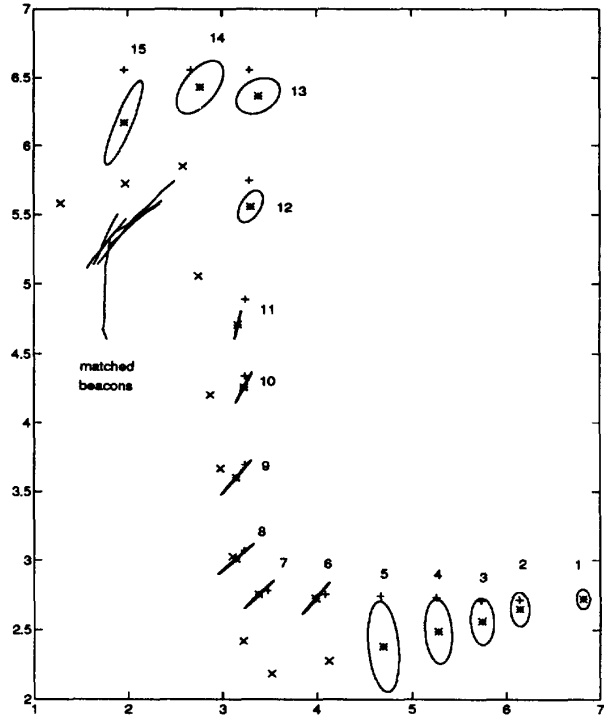


Figure 8: True, odometric and estimated positions of the mobile robot using a single corner beacon for localisation.

ons of the mobile robot as measured manually. The diagonal crosses (x) show the positions estimated from the vehicle's odometric system. Note that there is a systematic error between the true and odometrically measured positions. The stars (*) represent the estimated positions output from the extended Kalman filter algorithm. Centred upon each estimate is a 'certainty ellipse' which is known to contain the actual position of the robot with some constant probability. The ellipses have as minor and major axes the eigenvalues of the Kalman error covariance matrix and are drawn in the directions of the eigenvectors of the error covariance matrix [1]. Note that during the first 5 scans (recorded at true points 1 to 5 in figure 8) the corner beacon was not visible and therefore the Kalman filter estimates correspond to the odometric estimates. Note also that the certainty ellipses grow from points

1 to 5 as our estimates of the vehicle position get worse since we have no observations.

At position number 6 in figure 8, the corner beacon is observed and has been used to estimate the mobile robot's position. Note that the estimated and true positions virtually coincide and the certainty ellipse has shrunk, as we are more confident in our position. Between positions 6 and 11, the beacon is tracked by the algorithm and the certainty ellipses shrink further. The minor axis of each ellipse has an approximate direction along the line joining the sensor and the beacon. It is in this direction that we are most certain of the robot's position. At position 12 in figure 8, the beacon is lost (the robot has traversed the door and the corner is no longer visible). The effect on the positional estimates and certainties can be clearly seen.

Figure 9 shows the same mobile robot run, this time localising from *two* sensed beacons, namely the wall beacon at D and the wall at E in figure 7. It can be seen

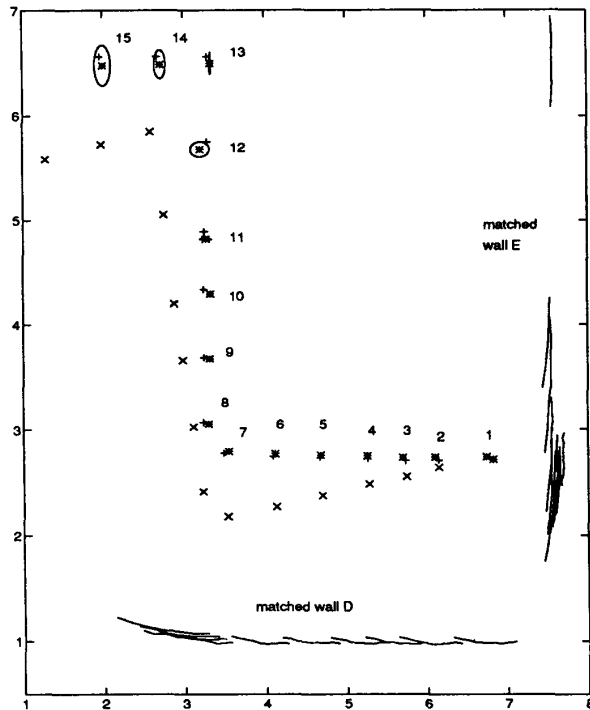


Figure 9: True, odometric and estimated positions of the mobile robot using two planar (wall) beacons for localisation.

that the algorithm is capable of producing highly accurate positional estimates of the robot and simultaneously building a map of its sensed environment as we become more confident of the nature of our features.

V. POST DISTRIBUTION - MECHANICAL CONCEPTS

In order to address the application of post distribution we have considered various mechanical concepts capable of receiving, transporting and delivering post. Our present vehicle is 0.75m wide and 1.10m long and is to operate in a building with corridors 1.4m wide. It is therefore essential that the space on board the mobile robot is used as efficiently as possible, so that the robot can deliver the maximum amount of post for its given size.

The following possibilities were considered:

1. A Robot manipulator arm and gripper to retrieve and collect the postal sacks.
2. A sliding and lifting mechanism to retrieve and deliver trays containing post from pigeon holes.
3. A tilting and rolling mechanism which, upon docking, allows the trays carrying the post to roll onto the mobile robot's transport system.

The third option above has the advantages that nothing needs to protrude outside of the working space of the mobile robot (unlike both other options) and that the mechanism itself can be simple, not requiring the use of high torque motors. Figure 10 shows the method to be used for transferring mail boxes from the central pigeon holes to the various departments within the building.

VI. CONCLUSIONS

We have presented here three of the research issues which we are currently addressing in order to realise an internal post distributing mobile robot.

We have shown that a "situation based behaviour selector" under the control of a neural network decision mechanism can provide a useful path planning technique. The mobile robot reacts to its environment with certain behaviours or reflex behaviours depending upon the input sensor data. The results of injecting a desired trajectory into the mobile robot's control system has also been demonstrated.

As the mobile robot moves it is capable of updating its position under an extended Kalman filtering algorithm using, at present, sonar range data. The algorithm is capable of finding naturally occurring geometric beacons and tracking them as the mobile robot moves. As new beacons (from obstacles) are found (ie: not predicted) they cause new tracks to be initiated, classified and eventually integrated into the map. The results show that successful localisation and map building is possible using this algorithm.

Finally we have outlined some of the mechanical concepts necessary to implement our application. The mechanical system presented offers an efficient solution to

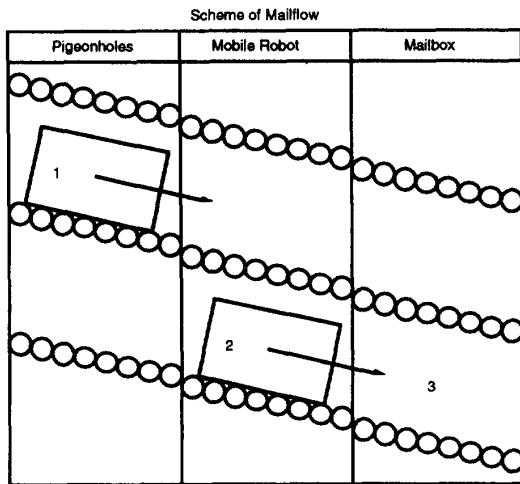


Figure 10: Mechanical concept for internal post transfer via the mobile robot. When the mobile robot has docked with the central pigeon hole point, the mail box in position 1 is released and can slide onto the robot's transport mechanism (position 2). After docking with the mail boxes (at each department) the tray is released from the mobile robot and can slide into position 3.

internal mail distribution in terms of mobile robot space usage.

The coordination and implementation of this work is currently under way in Zürich.

REFERENCES

- [1] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
- [2] D. Diez and G. Schweitzer. Real time systems for mechatronics. In *Second Conference on Mechatronics and Robotics*, Sept. 1993.
- [3] R. Kuc and M. W. Siegel. Physically Based Simulation Model for Acoustic Sensor Robot Navigation. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 766-778, 1987.
- [4] Jean-Claud Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [5] J. Leonard and H.F. Durrant-Whyte. Navigation by Correlating Geometric Sensor Data. Technical Report OUEL 1788/89, Oxford U. Robotics Research Group, 1989.
- [6] J. Leonard and H.F. Durrant-Whyte. Application of Multi-Target Tracking to Sonar Based Mobile Robot Navigation. In *International Conference on Decision and Control*, 1990.

- [7] T. Lozano-Pérez and M. A. Wesley. An Algorithm for Planning Collision Free Paths Among Polyhedral Obstacles. In *Communications of the A.C.M.*, volume 22, pages 560-570, 1979.
- [8] Winston L. Nelson and Ingemar J. Cox. Local Path Control for an Autonomous Vehicle. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1504-1510, 1988.
- [9] J.S. Singh and M.D. Wagh. Robot path planning using intersecting convex shapes: Analysis and simulation. In *Proc. IEEE Int. Conf. Robotics and Automation*, volume 3, 1987.
- [10] O. Takahashi and R.J. Schilling. Motion planning in a plane using generalised voronoi diagrams. *IEEE Trans. on Robotics and Automation*, 5(2), 1989.
- [11] N. Tschichold-Guerman. Rulenet: An artificial neural network model for cooperation with knowledge based systems. *Special Issue of the Machine Learning Journal on Symbolic Knowledge and Neural Learning*, 1994.
- [12] S.J. Vestli, N. Tschichold-Guerman, M.D. Adams, and S. Sulzberger. Integration of Path Planning Sensing and Control in Mobile Robotics. In *IEEE J. Robotics and Automation*, 1993.

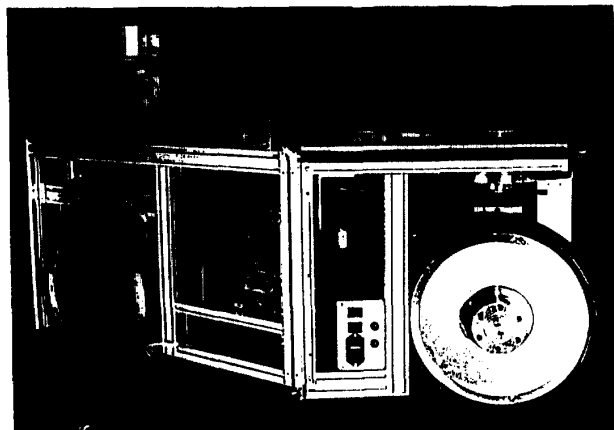


Figure 11: The mobile robot platform transporting the ultrasonic scanning mechanism.