

# Integration of Path Planning, Sensing and Control in Mobile Robotics

Sjur J. Vestli, Nadine Tschichold-Gürman, Martin Adams, Sandra Sulzberger  
 Institute of Robotics, ETH Zurich, 8092 Zurich, Switzerland  
 <vestli@ifr.ethz.ch>, <nadine@ifr.ethz.ch>, <adams@ifr.ethz.ch>, <sandra@ifr.ethz.ch>

*Abstract - Mobile MODRO is a modular mobile robot developed as a test platform for indoor applications at the Institute of Robotics, ETH Zürich. The modularity is ensured for all components of the robot: electronics, sensors, software and mechanics. The benchmark task for the modular mobile robot is planned to be the distribution of internal mail in the new institute building of the Institute of Robotics, currently under construction.*

*This paper reports the progress thus far with respect to hardware, sensors and software system.*

## 1.0 Introduction

In recent years, the construction and analysis of robot navigation algorithms which guarantee the safe convergence of the trajectory of a mobile robot from a point of origin to a point of destination, has become a growing concern. The main stream of the research done so far is the development of control technologies to enable mobile robots to operate within simple laboratory environments. On the other hand, industrial mobile robots of today address only a limited aspect of the overall navigation problem as they are usually restricted to following previously positioned beacons. Furthermore, there is often a lack of communication between industrial development and academic research in mobile robotics. In an attempt to bridge this gap, we intend to realize a mobile robot for mail distribution in office type environments.

## 2.0 Hardware

For the sake of simplicity and ease of alterations, sensor mounting and adaptation of electronics, the mechanism of the robot is built up using standard industrial profiles. The configuration of the robot can be seen in fig. 1. The robot is actuated through the front wheel, where the steering angle  $\lambda$  and the tangential velocity of the wheel  $\omega r$  is controlled. This gives the coordinate transformations in Eq. 1 and Eq. 2.

As the mobile MODRO is intended for indoor applications, and as in our experience discontinuities (such as door thresholds) are a major cause of problems when

$$\begin{bmatrix} \frac{dX_r}{dt} \\ \frac{dY_r}{dt} \\ \frac{d\theta_r}{dt} \end{bmatrix} = \begin{bmatrix} \cos \lambda \\ 0 \\ (\sin \lambda) / l \end{bmatrix} \omega r \quad (\text{Eq. 1})$$

$$\begin{bmatrix} \frac{dX_w}{dt} \\ \frac{dY_w}{dt} \end{bmatrix} = \begin{bmatrix} \cos \theta_r & 0 \\ 0 & \sin \theta_r \end{bmatrix} \begin{bmatrix} \frac{dX_r}{dt} \\ \frac{dY_r}{dt} \end{bmatrix} \quad (\text{Eq. 2})$$

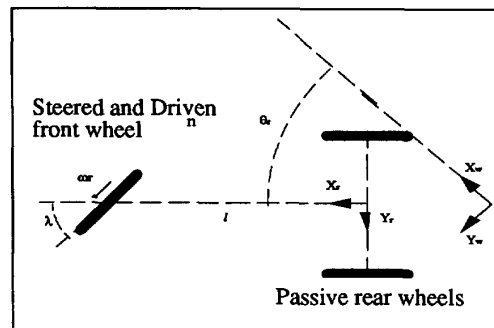


Figure 1. Mobile MODRO configuration.

navigating in buildings, the wheels were developed so that such floor discontinuities are absorbed through a spring damper system, similar ideas are expressed in [1]. The wheel is shown in fig. 2 together with the associated two mass oscillator model. The parameters of the model are as follows:

- $m_{\text{vehicle}} = 50 \text{ kg}$
- $m_{\text{wheel}} = 10 \text{ kg}$
- $c_s = 15 \text{ kN/m}$  (rubber spring spring constant)
- $c_t = 10 \text{ kN/m}$  (tyre spring constant)
- $d_s = 0.5 \text{ kNs/m}$  (rubber spring damping factor)
- $d_t = 0.65 \text{ kNs/m}$  (tyre damping factor)

A simulation of the wheels hitting a step in the floor of 0.04 m at a velocity of 1.0 m/s can be seen in fig. 3. The disturbance input is the vertical component of the trajectory a solid wheel would follow when traversing

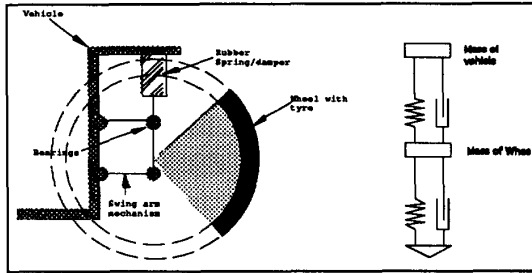


Figure 2. Wheel mechanism and associated model.

such a discontinuity assuming that the horizontal velocity stays constant. Clearly the shock is absorbed by the spring damper system. Analysis of the second derivative of fig. 3 shows that the acceleration of the vehicle frame is reduced by a factor of 5 compared to a wheel mechanism without a suspension system. This is a useful feature since it reduces the forces exerted on any component mounted onto the vehicle frame.

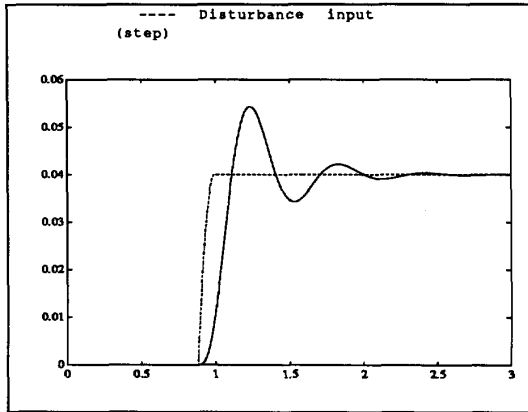


Figure 3. Response of the vehicle to a 0.04 m step on the floor.

## 2.1 Sensor Configuration

The sensor system is designed for maximum reliability in three layers:

- low level bumpers
  - high level bumpers
  - LIDAR (Light Intensity Detection and Ranging)
- The low level consists of contact bumpers all around the vehicle, which provide a failure free last resort sensor system.

The next level consists of a virtual bumper with a range of 0.5 m. This is realized with a Ultrasonic (US) system employing several US sensors actuated in parallel. This sensor provides a signal when an obstacle is within the sensor range, and an indication as to where

the obstacle was detected. Since all the sensors are actuated in parallel the update time of the sensor system is greatly reduced compared to conventional US sensor systems. Since the sensors are all active simultaneously the problems of specular reflections [3] are reduced.

## 2.2 Amplitude Modulated Continuous Wave Range Estimation

Part of the mobile robot's sensing structure consists of an amplitude modulated continuous wave (A.M.C.W.) lidar described in [10]. The sensor produces two analogue output signals:

1. The *phase* difference between the transmitted and detected A.M.C.W signals, thus giving a range estimate.
2. The *amplitude* of the returned signal which is dependent upon both target range and surface reflectivity.

### 2.2.1 Noise Propagation within the Sensor

In reality the phase estimate produced by such a sensor becomes more useful when it is combined with the amplitude or strength of the returned light signal. The strength of the returned signal has a large effect upon the behaviour of the sensor. It gives an indication of the confidence we can place in the phase measurement. At the phase comparator the range is estimated from the phase difference shown in fig. 4.

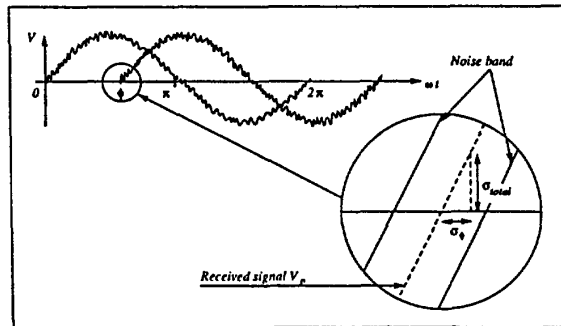


Figure 4. Phase estimation in the presence of noise.

The relationship between the standard deviation in the absolute *phase* of the received signal  $\sigma_\phi$  to the combined electronic and photon noise standard deviation  $\sigma_{total}$  transmitted by the receiver electronics, can be seen graphically as a *noise triangle*, [11] which, from fig. 4 gives:

$$\frac{\sigma_{total}}{\sigma_\phi} = \left| \frac{\partial}{\partial(\omega t)} V_p \right|_{V_p=0} \quad (\text{Eq. 3})$$

where the right hand side of Eq. 3 is the magnitude of the change of received signal voltage with phase at the zero crossing - i.e.: when the wave cuts the  $\omega r$  axis.

Under this analysis it is possible to derive the relationship between the range variance  $\sigma_r^2$  and the amplitude of the received signal  $V_r$  [10]:

$$\sigma_r^2 \approx \left( \frac{\lambda \sigma_n}{4\pi} \right)^2 \left( \frac{1}{V_r} \right)^2 + \sigma_e^2 \quad (\text{Eq. 4})$$

where  $\lambda$  is the modulating wavelength,  $\sigma_n^2$  the constant receiver diode shot noise and  $\sigma_e^2$  the constant additive electronic noise effect caused after the signal is received.

### 2.2.2 Lidar Depth Maps

The left hand scan in fig. 5 shows a 360° scan result, taken in a laboratory environment. The triangle shows the position of the mobile robot transporting the sensor and each small cross represents a data point.

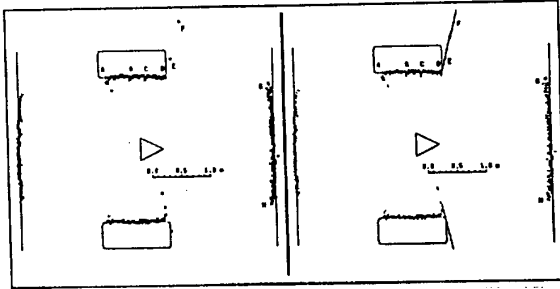


Figure 5. The left hand plot shows data produced by the lidar. In the right hand scan, lines of length  $2\sigma_r$  are centred on the range observations to show the uncertainty associated with each range sample.

Each range sample in fig. 5 is accompanied by a range variance estimate, so that data can be accepted or rejected according to uncertainty associated with each point.

### 3.0 Software Structure

Considering the benchmark application, the software (and thus the robot) must have the following abilities:

- safely interact with humans
- be robust (i.e. have extensive error recovery features).
- adapt to changes in the environment

We intend to achieve these properties by using artificial neural networks and fuzzy logic as well as classical information processing techniques (figure 6). For this purpose, we have developed new ANN-models for optimization of rule based systems (M-RCE) and for fuzzy controllers (FUN).

To allow a safe software development for different information processing methods, a modular software architecture is used [2]. The software architecture is grouped in three main parts: the controller, sensor data processing and task planning.

The controller is realized in three hierarchic layers: basic controllers, local navigator, global planner. The algorithms for the realization of these are described in the following sections.

The sensor data processing makes the filtering and the basic interpretation of the sensor signals. The procedures in this module return the distance to the next object or a binary value specifying whether there is an object in the sensor measurement range or not. There is such a procedure for each sensor. In the next step, algorithms for combining the partly contradictory informations from the different sensors (sensor fusion) will be developed.

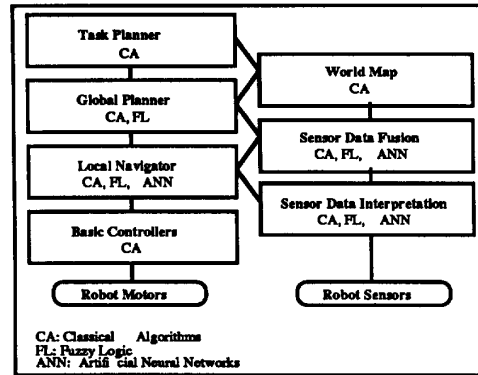


Figure 6. The software architecture .

The task planning will be realized according to the distribution task of the robot.

### 4.0 Controllers

#### 4.1 The Basic Controllers

The steering angle  $\lambda$  and the wheel tangential velocity  $\omega r$  is controlled using a PD and P controller respectively. In fig. 7 and fig. 8 the step responses for velocity and angle is shown. Notice that the actual velocity is able to follow the desired signal within 2.3 seconds lag. Notice also that the controller is able to turn the wheel at maximum speed during at least 90% of the rotational trajectory. Hence minimum time steering control is almost attained.

## 4.2 The Local Navigator

We have developed several algorithms for the local navigation of mobile robots and we have implemented them in our own developed simulator environment. In the following sections, these algorithms and some simulation results are presented.

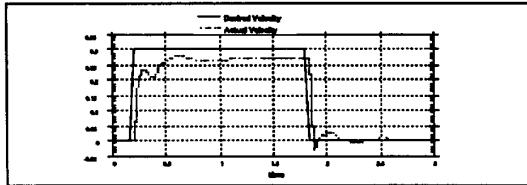


Figure 7. Propulsion step response ( $\alpha$ )

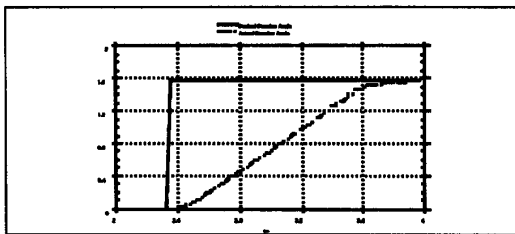


Figure 8. Steering step response ( $\lambda$ )

### 4.2.1 Rule Based System

A very efficient method for the realisation of the local navigation for a mobile robot is to work with a rule base. We have set up a rule base with the following strategy: The information from the sensors in the directions front, left and right are divided in three regions: near, medium and far. This partition defines a 3-dimensional area in the sensor space. To have a consistent rule base, rules for each region in this area has to be defined. The resulting rule base successfully drives the mobile robot to a goal position.

### 4.2.2 Artificial Neural Networks (ANN)

The rule base as explained in the last section sometimes produces non satisfactory trajectories. A well known problem is oscillations of the robot while driving parallel to a wall. The reason for an oscillation is the following: While moving towards a goal, if the robot detects a wall, it makes a movement to the right or to the left, depending on the rule base, to avoid it. The next moment after it turns off the wall, if it is not near enough to the wall, the sensors on the side of the robot do not detect the wall any more, so the robot turns back and detects the wall again, etc.

For solving this problem, we have used the ANN-model M-RCE (Modified Restricted Coulomb Energy)

[4] for optimizing the rule base to eliminate such problems. M-RCE is a feedforward ANN model with a supervised learning strategy, a dynamic architecture and discrete outputs. Its main idea is to approximate classes with hyperrectangles. As each rule in an n-dimensional feature space can be described with an n-dimensional box (figure 10), this ANN model is very well suited for unequivocal rule translation into and from the net.

The non-optimal rule base was translated into a M-RCE network and it was trained with examples showing how to behave at several distances from a wall. After only 2 training epochs (i.e. each example was presented 2 times), the net was able to drive the robot without oscillations parallel to a wall. For efficiency reasons, the knowledge of the network was translated back into a rule base afterwards.

The learning algorithm of M-RCE splits the three sensor regions (near, medium and far) until a finer resolution is reached thus resulting in non-oscillatory motion in the vicinity of the wall. In a next step, this rules based system will be extended to operate with the data delivered by the optical sensor discussed in section 2.2. The ANN model M-RCE will be used as an optimisation algorithm for the rule base

The optimization of rule bases with M-RCE networks compared to manual modifications of rules has the big advantage that the rule base never gets inconsistent, as the training algorithm of M-RCE does not allow rules with different THEN-parts if there are overlap between their IF-parts. So it is not possible to have contradictory rules. This allows a fast and safe improvement of the rule base.

### 4.2.3 Fuzzy Logic (FL)

As an alternative to the rule based system described above, which works with binary logic, a fuzzy controller has been developed for the local navigation of the mobile robot. For this purpose, the fuzzy development tool FLIE (Fuzzy Logic Inference Engine) has been implemented.

The inputs to the fuzzy controller are: goal direction, measurements of three sensors (front, left, right) and the change in these measurements, the actual driving direction and the actual steering angle. Its output is the steering angle.

The results of the simulation are very satisfactory, because the robot was able to reach the goal with noisy sensor data and inexact position update.

### 4.2.4 FUN: Combination of ANN & FL

Experience with Fuzzy Logic show that this technique allows a fast development of controllers. How-

ever, due to the large number of parameters and little knowledge about their exact influence on the behaviour of the controller optimization is difficult.

Research on the possibilities of combining these two techniques has been done for several years, especially in Japan [5], [6], [7]. The common factor of these methods is that they do not allow a simple bijective mapping from the rulebase into the net. Where a learning method is used, there is no possibility to extract the result from the net. There is no solution to the problem, where both rules and membership functions are learned simultaneously.

We have developed a new neural network model, FUN [8], which is very well suited to represent logic expressions. In order to enable an unequivocal translation of fuzzy rules and membership functions into the network, special neurons have been defined, which, through their activation functions, can evaluate logic expressions.

The network consists of an input, an output and three hidden layers. The neurons of each layer have different activation functions representing the different stages in the calculation of fuzzy inference. The activation functions can be chosen individually for different problems (e.g. AND can be implemented as the MIN-function, the dot product function, or any other function).

The network is initialized with a fuzzy rulebase and the corresponding membership functions. The network can then be trained with supervised or reinforcement learning (RL) strategies to optimize the rules and / or the membership functions. FUN contains two different training algorithms for the optimization of the rules and the optimization of the membership functions.

It is very difficult to create learning examples for the mobile robot navigation problem, so the RL strategy was used for the optimization of the fuzzy controller. For RL a cost function must be defined. The cost function provides us with an evaluation of the path, and hence the rule base. The learning algorithm then attempts to optimize the cost function by adapting the network / rule base. The execution continues until either: the goal is reached, or the robot collides with an obstacle, or a predefined time,  $T$ , has elapsed. At the end of the execution the cost function is calculated as given below:

$$Cost = \begin{cases} Constant & \text{if robot drives into wall} \\ |\vec{Goal} - \vec{ActualPos}| & \text{otherwise} \end{cases}$$

The input-output signals of the fuzzy controller optimized by FUN are shown in fig. 9. Clearly, for having better results, the inputs must be extended with informa-

tions described in the previous chapter. Our aim was to test the performance of FUN with a small example.

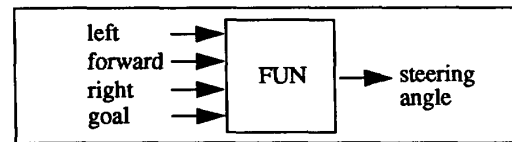


Figure 9. Inputs and outputs of FUN

An untrained net with an a priori defined rule base would typically perform as seen in fig. 10, after optimization the controller performed as in fig. 11. The oscillation in fig. 10 is a result of contradictory rules in the rule base. A more demanding environment and the corresponding performance of the (optimized) controller can be seen in fig. 11. The path produced by the controller for the environment given in fig. 11 is clearly not optimal with respect to length. However, since the cost function takes no account of the path length this was expected.

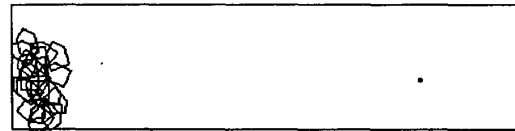


Figure 10. Path of the mobile robot before learning

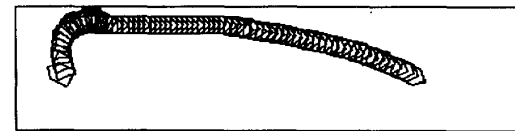


Figure 11. Path of the mobile robot after learning

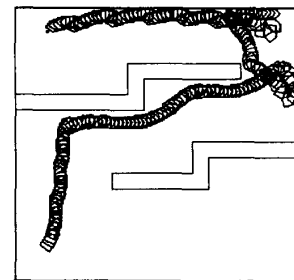


Figure 12. Path of mobile robot in a more complex environment (after optimization with FUN)

### 4.3 The Global Planner

The road map used for the mobile MODRO is represented in a partly connected, directed graph. The graph nodes describe points in the building where the mobile

robot can drive, e.g. corridors, doors, etc. Graph nodes contain information about regions represented by them, e.g. distance to the next node, a measure for the difficulty to reach the next node (narrow corridor, crowded region, doors on the way, etc.)

Trajectory planning is realized with the classical graph search algorithm A\*, which uses the data in the graph nodes as heuristics for choosing one of several possible trajectories. The heuristics are defined using linguistic rules and linguistic sets [14] and [15].

The output of the global planner is a sequence of knot points which the robot has to attain (in succession) in order to arrive at its destination. Between each knot points the robot employs a local planner, as described in the previous sections.

## 5.0 Conclusions

We have presented a modular structure for the design of a mobile robot for autonomous navigation in indoor environments. At the lowest level we have considered the stability of the platform in the presence of floor discontinuities and have shown that it is possible to achieve near optimal steering and speed control.

By also considering noise sources and their propagation through an optical sensor, we have derived a model for an A.M.C.W. lidar which produces range uncertainty information with each range estimate.

At the highest level, efficient global and local path planning can be achieved using traditional search methods in combination with ANN and fuzzy logic based controllers.

## 6.0 Acknowledgements

Thanks to Einar Nielsen for many valuable suggestions, and to "The Royal Norwegian Council for Scientific and Industrial Research" for supporting Sjur Vestli.

## 7.0 References

- [1] K. Tani, et. al., "A computer-controlled active suspension for a wheeled terrain robot", *International journal of computer applications in technology*, Vol. 3, No. 2, pp. 100-104.
- [2] D. Diez, R. Roshart, "A versatile, expandable configurable controller for a modular robot", 23rd international symposium on Industrial robots, Barcelona, 6-9 October 1992.
- [3] P. J. Parniz, P. J. Drazan, "Recognition of unreliable ultrasonic range data in a robotic environment", *Robotica*, Vol. 7, pp. 223-229, 1989.
- [4] N. Tschichold-Gürman, M. Ghazvini, D. Diez, "MRCE: A self configuring ANN with Rule-Extraction Capabilities", *Artificial Neural Networks, Proceedings of the 1992 International Conference on Artificial Neural Networks (ICANN-92)*, Brighton, 4-7 September 1992, Vol. 1, pp. 269-272.
- [5] Kosko B., "Neural Networks and Fuzzy Systems", Prentice Hall, 1992.
- [6] Takagi H., Hayashi I., "NN-Driven Fuzzy Reasoning", *International Journal of Approximate Reasoning*, Vol 5, No 3, May 1991, pp. 191-212.
- [7] Enbutsu I., Baba K., Hara N., "Fuzzy Rule Extraction from a Multilayered Neural Network", *International Joint Conference on Neural Networks*, Seattle, WA, July 8-12, 1991, pp. II 461 - II 465.
- [8] S. Sulzberger, N. Tschichold-Gürman, S. Vestli, "FUN: Optimization of Fuzzy Rule Based Systems Using Neural Networks", submitted to the IEEE International Conference on Neural Networks (ICNN-93), San Francisco, California, March 28 - April 1, 1993.
- [9] S. Boviere, B. Demaya, A. Titli, "Fuzzy logic control compared with other automatics control approaches", 30. IEEE-CDC, Brighton, December 11-13, 1991.
- [10] M. D. Adams, "Optical Range Data Analysis for Stable Target Pursuit in Mobile Robotics" DPhil thesis, University of Oxford, U.K., 1992.
- [11] G. L. Miller, E. R. Wagner, "An Optical Rangefinder for Autonomous Robot Cart Navigation", technical report, A.T.&T. Bell Laboratories, 1987.
- [12] J.-C. Latombe, "Robot motion Planning", Kluwer academic publishers, 1991.
- [13] P. H. Winston, "Artificial Intelligence", Addison Wesley, 1992.
- [14] L. A. Zadeh, "Fuzzy Sets", *Information and Control*, 8, pp 338-353, 1965.
- [15] H.-J. Zimmermann, "Fuzzy Set Theory - and Its Applications", Kluwer academic publishers, 1990.