# ACHIEVING HIGH PERFORMANCE SONAR-BASED WALL-FOLLOWING

**Carlos Puchol***
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188. USA

**Abstract.** In this paper we develop a technique to achieve robust high performance real-time wall-following behavior of a mobile robot in an indoor office environment, more specifically, in a corridor environment. The mobile robot achieves increasingly better performance by learning the environment's (most important) features in successive runs through it. This allows the robot to perform the task repeatedly, reliably, increasing the speed at which it is done after every step, without losing accuracy. We are basing our approach in the Spatial Semantic Hierarchy [Kuipers et. al. 1993].

**Keywords.** Robotics, Navigation, Mapping, Real-Time Control

## Introduction

The problem of wall-following can be described as that of *tracking* a so-called "wall" (an *object* in the world, for our purposes) until a certain condition, usually derived from the environment, holds true. Most of the time, in real applications, the condition is encountering some object in front of the robot or detecting the ending of the object being tracked.

To be able to achieve high performance, we decompose the problem and map it into the layers of the Spatial Semantic Hierarchy (SSH) [Kuipers et. al. 1993]. By using the word "track," we are introducing the problem as one of *maintaining some measure* associated with the object, while the subject or the object, or both are in motion. We consider the problem at this level as a control problem, where the reference signal used is a desired *approximate* distance from the wall. The distinctive places are defined based on the variations in the distance measured to the object being tracked. In our scenario, the procedural level of the SSH is trivially reduced to following a straight path until one end of the corridor is reached, registering the distinctive places found along the way. At the topological level, we learn (in the broad sense) the relative positioning of the features in the environment. The metrical information is extracted from the environment and the robot's own odometry information and annotated on top of the topological structure.

In the following sections we describe what are the assumptions made, the sonar sensor model used and the control laws used at the lower levels of the structure. Next we describe how we achieve higher performances than the direct control algorithms would provide, followed by a discussion about the performance of the algorithm used and an evaluation of the odometry information provided by our robot.

## Assumptions

Obviously, in office-like environments the walls are not likely to be straight and usually have doors, elevators, stairways, etc. The difficulty of the problem stems from this fact as well as from the noise and inaccuracies in the sonar measurements. Objects, like tables and chairs are placed in the environment. However we rely on the 2-D assumption, i.e. the three dimensional geometry of the environment is orthogonal to the plane of the sensory horizon of the robot [Leonard et. al. 1992].

Another assumption that we impose in the environment is that the features in it are stationary while the robot is performing the exploration.

Again, our goal is to establish a performance measure of the behavior in the restricted, known environment described above. Assuming dynamic features would require significant changes to our approach.

The physical robot used in the paper is an RWI™ (Real World Interface) with a one foot diameter, three wheel base

*Sponsored by a graduate Fulbright Scholarship.

and an enclosure on top, with twelve sonar transducers, a G96 (Gespac) backplane, and a Motorola 68000-based microprocessor board. The base has its own 68HC11-based microcontroller board, which communicates with the 68000 board through a serial channel.

## The Sonar Sensor Model

The sonar sensor model used is a simplification of the theoretical model introduced in [Leonard et. al. 1992] for our task and environment conditions. Essentially, our model is based in the following observations[1]:

- *Strong returns are very accurate, with low error distributions.*
- *Weak returns can be caused by "time-delay" range errors in which the range measurement can be up to* $19cm$ *too far, while no more than* $1cm$ *too close.*

With these observations, from the upper layer point of view, we have implemented a "return" as a pair $(value, uncertainty)$, where the value indicates the range, with it's corresponding degree of uncertainty.

We filter the sonar returns to obtain the $value$ by taking the *minimum* of several repetitive samples. The $uncertainty$ is based on the variance of the returns sampled.

This is a simple and efficient way to obtain accurate readings from the sonar sensors for reasonably short ranges. This approach would only fail to give us accurate readings if all the measurements had been caused by weak returns, in which case the variance grows and so does the $uncertainty$ value.

The experiments performed confirm the good performance of this approach for the task of wall-following. In our case, we find the sampling 3 sonar returns, gives good accuracy and good levels of certainty. Increasing this number hurts the overall performance of the control algorithm, since it every sonar return takes a significant amount of time to be sampled. Thus with this setting, we can treat the sonar as an ideal ray-trace scanner.

## Control Laws

The robot has advanced control algorithms in the base that make it unnecessary to focus on low level details such as straight line control laws, acceleration/deceleration ramps, etc., thus we do not concentrate our attention on the lowest level of implementation.

Initially, the robot is left in the corridor, without any knowledge of what its position is with respect to the environment.

[1] The definition of strong returns and weak returns can be found in [Leonard et. al. 1992] as well. Fundamentally, the later can be due to "specularities," in a short-range sensing mode. In other words, we cannot use the densely sampled data model and we do not use long-range sensing (we don't have local support, thus the "regions of constant depth" (RCD) approach cannot be used either.

The first local control law applied at this point is to locate the nearest wall and get close to it. Once the robot is facing the wall, we enter in a proportional control loop to make its distance from the wall equal to the reference. Since the base control algorithms are quite accurate this loop usually terminates in two steps. This imposes that initially the robot has to be located near to a wall (not a another place, like a doorway) since, this is the reference the robot is going to be tracking all the rest of the time.

At this point, before the robot starts the straight line following procedure (each time it gets ready to do so in every round to the corridor), we have found that a very important factor for the performance of the procedure is that the the robot is facing the right direction as accurately as possible, since this is the the initial direction ti is going to follow, and a bad start can drive the control algorithm out of a stable state. To do this, we have implemented a discrete PD regulator to locate the body of the vehicle as parallel as possible to the wall.

## Achieving High Performance Behavior

As pointed out in [Leonard et. al. 1992], navigating at higher speeds always demands some degree of anticipation and capacity of reaction. We argue that the rate at which the sonar observations with appropriate filtering can be taken using the low-cost off-the-shelf technology currently available (e.g. Polaroid sonar ranging systems) effectively limits the speed the robot can travel at, for a given environment detail scale. Another factor that affects the achievable speed obtained is the density of the scan. In our case (a 12 sonar ring of sensors) approximately 3 samples per second can be taken. These features in the system practically limit our speed to about $180mm/s$. At higher speeds and using a direct control algorithm, the behavior of the robot becomes unstable.

Were we to discern the regions of constant depth as defined in [Leonard et. al. 1992] with our current hardware, the robot would have to perform many slight rotations to achieve a similar degree of resolution, which would take a significant amount of time, severely limiting our speed. We find however, that with our current hardware, minor filtering and the environments we are working with, a reasonable degree of resolution can be obtained with speeds of up to $100mm/s$. Figure 1 represents the measured distance from an "artificial" corridor with one door, inside the AI laboratory. Figure 2 represents the measurement in a real corridor with a double door to a stairway. In both cases, the results have quite good accuracy.

We base our efforts for high-speed navigation in two stages, an *exploration* phase in which the vehicle produces an initial annotated topological map of the corridor wall and a proper *navigation* phase in which modifications of the topological map can be performed based on the returns from the sonar sensors. We elaborate on these in the following sections. The exploration phase starts right after the vehicle has approached the nearest wall, has positioned at the goal distance and has aligned itself with the wall. We call this node the "home" base. The exploration phase is ready to

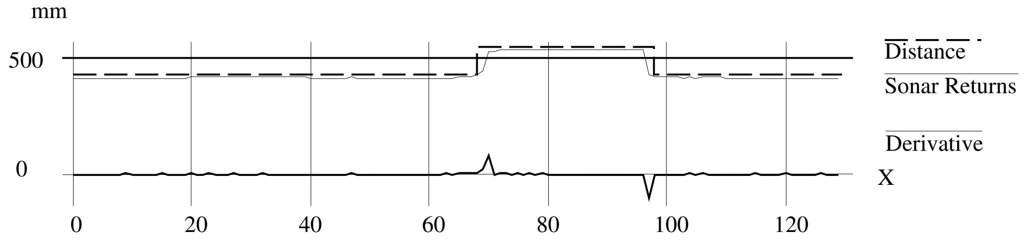


Figure 1: Artificial corridor measurement with a 12-sensor sonar ring.

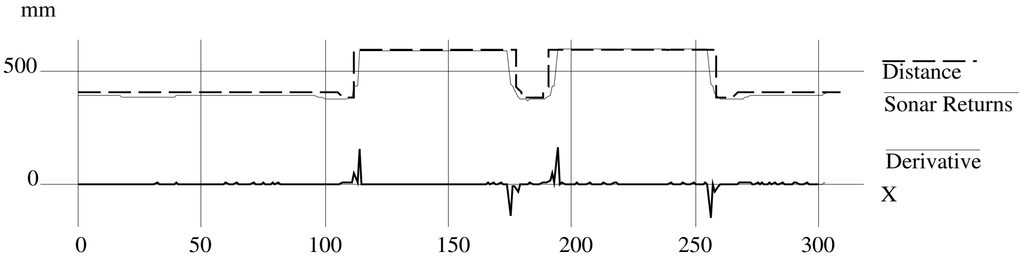


Figure 2: Real corridor measurement with a 12-sensor sonar ring.

start, and the topological graph is created with only one node, in which all the odometry information is registered.

## Initial Exploration

At this stage, the algorithm performs a straight line following procedure, with a PD control regulator to follow the wall, repeatedly detecting all the distinctive places along the way, adding them to the topological graph.

The vehicle performs the wall following procedure until the vehicle enters a "no-control" zone. These are zones in which the sonar sensory system has detected that some feature in the environment has disrupted the "flatness" of the wall, at which time it creates a new node (and an arc linking it to the previous node visited) in the topological graph. The node is labeled with the odometry information relative to the home base. The arc is labeled with the distance to the wall that has been followed.

The distinctive places are detected by computing the derivative of the distance to the wall along with the sonar returns and comparing it against a threshold, rather than performing a hill-climbing strategy that might imply having the vehicle stop. At this point, the motor system is left only following a straight line command. This state remains in place until the end of the features is found. This is detected only when the derivative remains constant (within some window) for a number of sonar samples.

The vehicle is now past a place, following a linear trajectory. The algorithm falls into the wall-following mode again, but this time, the reference distance at which to track the object is not the original distance to the wall, but the actual distance that has been detected. Usually, the doors or other features of the environment are not large, relative to the length of the corridor, and it doesn't take a long time until the wall is back again. If this doesn't happen within a window, then something is declared to be wrong, and it is annotated as an uncertainty value for the current arc and node. A new original distance to wall goal is set at this point.

Assuming that the environment feature is surpassed eventually, the algorithm keeps performing this loop until the end of the corridor is detected (or a collision is detected).

The algorithm then rotates 180 degrees and labels the node in the graph as an endpoint and starts a new odometry annotation, for the following run (the annotations per node are two, one in each direction, relative to the endpoints). The next run is similar to the one just described, except that the vehicle is verifying the topological/metrical map along the way, possibly making modifications to it, if discrepancies are found.

The navigation phase starts when the vehicle returns to the "home" base with (few or) no uncertainties annotated in the topological graph.

## Actual Navigation

The actual navigation is performed by anticipating the sonar returns in the graph with metrical information and contrasting them against the odometry information observed. The basic procedure is to keep increasing the speed of the vehicle to a limit, unless the sonar returns are not in accordance with the topological map, in which case the speed is reduced until the sonar returns agree with the topological map. If the speed is reduced below a threshold, the algorithm enters in an exploration mode identical to the one performed at the beginning.

The actual control implemented is a discrete PD regulator with the reference given by the annotations in the arcs in the graph and the actual signals given by the sonar returns. If the return from the sonar falls within the boundaries of a distinctive place, the measurement is discarded. If the return belongs to a flat surface and some degree of uncertainty is returned along with it or it is found to reveal that the vehicle is beyond some bounds of the control algorithm, the speed is reduced.

## Performance Evaluation

One of the problems found during the process of implementing the algorithm here described is that a critical point in the algorithm is when the robot enters a zone where there are several distinctive places, with a "bad" entry angle with a mildly high speed. The robot heads in the wrong direction, but since the observations from the sonar sensor can be potentially uncertain or noisy within the that zone (thus cannot be interpreted without a lot of processing), the vehicle can end up in a position quite far from the right position intended for tracking the object. While in a corridor this may not happen as long as the relation between flat surfaces and edges in them is relatively big, the main problem resides in the end points of the corridor, specially if there are any doors or edges near them.

In a similar vein, another critical point is when the vehicle's body is tried to be positioned parallel to the wall to follow. The rotation primitives in the mobile base are not as precise as we would have liked.

Unfortunately, we have not been able to test the navigation at high speeds due in part to technical difficulties accessing the odometry information at high speeds as well as due to time constraints. Another difficulty that we have had is related to the run-time system in place in the computer board in the RWI. This system does not provide a real-time timer system. With real-time timers, the tracking of the actual position of the robot could be done much better, by being able to allocate in the time line the instants where the control actions and the sensory inputs needed to be performed. Unfortunately, much of the development time spent in this project has been dealing with technicalities associated with the actual hardware. We would have liked to show a performance graphs of the actual development of the algorithm as well.

## Conclusions and Future Work

We have successfully mapped the problem of wall-following into the SSH architecture. For this task, some of the levels in the SSH reduce to trivial procedures or elements, but still, the general approach can been applied.

Some work needs to be done regarding the reliability of the interface with the hardware. For instance, a primitive call whose result would return the odometry information attached to one particular sonar return would be particularly helpful. The actual higher speed runs have not been performed, only speeds up to $180mm/s$ have been developed. Due to the unreliability in the odometry information, it was not safe to try any higher speeds.

Although the control regulators presently implemented in the system are in fact quite robust, it would be interesting to try this approach, which has worked quite fine in the "office corridor" environment to a more general office environment.

## Appendix: Accuracy of Odometry

One of the issues involved in the upper layers of the SSH is finding out how much accuracy we can rely upon when obtaining odometry information from the robot base.

Several experiments have been performed to find out what is the average cumulative error that our algorithm can expect from the odometry information provided by our mobile robot base. The measurements have been performed to find out the straight-line distance error with respect to the translation commands issued to the base. The error obtained is linear with respect to the length of the translation and has an *average* of $0.449\%$. The *standard deviation* of the error is also linear with the translation and it is $17.09\%$. This standard deviation over the error can be considered high, but we believe the measurements have this degree of uncertainty because they have been done by hand, not because of defects in the mobile base control algorithms.

We have performed several measurement runs that show that the above results are essentially independent of the speed as well as the acceleration/deceleration ramps at which the base is commanded to move.

Given these results, and given the nature of the task at hand, the error obtained doesn't impact in the performance of the algorithm, for two reasons, one being that the metrical information is always taken to be approximate (this error is taken care of explicitly in the upper layer of the SSH (metrical)). The second is that at the end points of the corridor, the internal accurate localization is achieved, thus the error does not accumulate (in effect, even though for mapping purposes it is essential to have reliable odometry information, for the wall following task, internal representation does not have to be accurate with respect to the outside reference, as long as it is repeatable)

## References

[Kuipers et. al. 1993] Kuipers, B. J., Froom, R., Lee, W-Y., and Pierce, D. *The Semantic Hierarchy Approach to Robot Learning.* In Connell, J. and Mahadevan, S. (Eds.) *Robot Learning.* Kluwer Academic, 1993.

[Leonard et. al. 1992] Leonard, J., and Durrant-Whyte, H. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic, 1992.