SELF-LOCALISATION IN THE 'SENARIO' AUTONOMOUS WHEELCHAIR

P.D.Beattie and J.M.Bishop,

Department of Cybernetics, University of Reading,

Whiteknights, READING,

Berkshire, UK.

email: Paul.Beattie@royalmail.co.uk, J.M.Bishop@uk.ac.rdg

Abstract

This paper introduces the Focused Stochastic Diffusion Network as a novel method of self-localisation for an autonomous wheelchair in a busy, complex environment. The space of possible positions is explored in parallel by a set of cells searching in a competitive co-operative manner for the most likely position of the wheelchair in its environment. Trials of the SENARIO autonomous wheelchair project indicate the technique is practical and robust.

Keywords

Localisation, Neural Networks, Stochastic Diffusion

Categories

1, 3, 5.

1. Introduction

This paper discusses the development and application of a novel neural network used to determine the position and orientation of an autonomous wheelchair. A brief overview of other localisation and self-localisation techniques is given as an introduction to the basic localisation problem. The (x, y, θ) space of possible locations is explored in parallel by a set of cells searching in a competitive co-operative manner for the most likely position of the wheelchair in its environment. Trials of the prototype system in a noisy environment indicate the technique is practical and robust.

The simple Stochastic Diffusion Network (SDN) is reviewed, as an understanding of its operation is a prerequisite to understanding the Focused Stochastic Diffusion Network (FSDN). Practical problems resulting from the use of laser range data to localise a wheelchair in a large, real environment are also outlined, and working solutions to them provided.

1.1 SENARIO

The development of an autonomous wheelchair, which is able to self-localise, is one of the goals of the EEC TIDE (Technology and Innovation for the Disabled and Elderly) project named SENARIO. The overall aim of the autonomous wheelchair is to provide a device that people within a hospital or other such institution may wish to use to assist them to independently travel around within a known environment (Beattie, 1995; Katevas *et al*, 1995).

The wheelchair control system consists of several subsections: the user interface uses voice recognition to obtain commands, or goal locations from the user. The interface is taught the user's voice for each of the control commands and goal locations. The sensor subsystem, using ultrasonic sensors and a bumper, detects local obstacles to allow the risk avoidance subsystem to determine a local path along which to navigate. The risk avoidance subsystem co-ordinates the activities on the wheelchair and also calculates a global path that is to be followed. The positioning subsystem provides the risk avoidance subsystem with the current location of the wheelchair within its environment. This allows the risk avoidance subsystem to determine is progress along the current path and set new routes to follow.

1.2 Localisation

Many localisation methods described in the literature only function if an initial, exact or approximate, position and orientation are provided. Thus the term self-localisation has been established to indicate an entirely autonomous localisation system. Self-localisation is necessary if the current location is not known due to re-initialisation or an accident (Madarasz, 1986). The position estimation problem is the primary problem that must be solved for autonomous guided vehicles (AGVs) working in structured environments, and this is particularly difficult if passive or active beacons are not used (Cox, 1991).

A simple method for non-self-localisation is used by Evans *et al* (1992) and Galles (1993) who used triangulation from detected landmarks to localise, and then used wall following routines between landmarks. Thus absolute position was only known in regions where landmarks existed. Wilkes (1994) has improved on this by using a current position estimate from odometry to select two features from an environment model. A vision system then scans the image for the two features, and when found triangulation is used to locate the robot. Thus the robot can update its location from the features when required, and provide an estimate from odometry at other times.

Cox (1991) used an infrared range finder and odometry on his AGV, Blanche, but provided a fairly accurate initial position estimate which it was able to accommodate. He assumed that the robot was always near its previously calculated location. His method of localisation selects the nearest map line for each range. The location that then minimises the sum of the squared differences between the range point and the nearest wall is determined. The image is moved to accommodate the error vector, and the process is repeated until the error vector is within a tolerance value. The location is then deemed to be the previous location plus the sum of the error vectors. Bourhis (1994) and, Freund & Dierks (1993) also used Cox's (1991) localisation algorithm, the later with an infrared range finder.

Rather than using features Durieu *et al (1989)* used active beacons to improve a position estimate obtained from odometry data. They use active beacons which transmit a coded infrared signal, when triggered from ultrasonic transmitters on board the robot. The time of flight of the ultrasonic and infrared signals is measured to provide a distance to the beacons. The position estimate is improved by minimising the difference between the measured beacon distances and the expected beacon distances.

Self-localisation is divided into two methods, one using uniquely identifiable beacons (Byler, 1995), the other using model matching (Bilgic, 1995) between the sensor data and the environment description in the AGV. These methods of self-localisation, however, require that the AGV knows the environment that it is to operate within. The environment can be provided (Drumheller, 1987) or self-taught localisation (Figueroa, 1994) allows an AGV to determine its position without needing to know its previous position. This enables the AGV to be switched on at any location within its environment and determine a navigational path to a goal position, without operator intervention.

A self-localisation method is needed for an autonomous wheelchair in a hospital environment to ensure that the wheelchair is robust enough to be able to determine its location if it is moved manually to an unknown position.

1.3 The Wheelchair's Sensors

Figure 1 shows the SENARIO wheelchair in its test environment. The sensors employed on the wheelchair for the localisation sub-system were two 180° infrared range finders, mounted back to back on the wheelchair at a height of 192cm to avoid the majority of temporally variant obstacles within the environment. The range finders provided a 720-element vector of ranges to the nearest obstacles around the wheelchair. The range finders are able to detect objects up to a range of about 50m, which meant the walls of most rooms are within range. Passive radio beacons were also used to provide a unique code when the wheelchair was within a metre radius of a beacon. Odometry information :- the distance travelled by each of the two wheelchair drive wheels - is available through the wheelchair risk avoidance subsystem.

1.4 Neural Networks for Localisation

Neural networks provide a technology that can be used to determine the correlation between input data and a predetermined map, to obtain the location of a robot within the environment. A neural network is required when triangulation is impractical due to the volume of data being processed, for example when the environment is large and the desired resolution of localisation is high.

Tarassenko *et al* (1991), Marshall & Tarassenko (1994) and Bishop *et al* (1995) used very large scale integration (VLSI) neural network techniques to localise an AGV in a small environment by training their network at each of a large number of evenly distributed positions within the environment. It is not, however, obvious as to how they determined the orientation of the AGV from their network output. The system is impractical beyond a single room environment as all possible positions and orientations need to be taught to the network.

Townsend *et al* (1991) used a Radial Basis Function (RBF) network, operating on data from an infrared range finder. They did not however, use the range values returned directly from an infrared scanner. Instead they reduced the dimensionality by extracting seven key features from the data. The seven features chosen were: the shortest range; the median range; the longest range; the magnitude of the largest discontinuity; the magnitude of the second largest discontinuity; the energy in the scan; and the length of the longest wall segment. The number of corners detected was not chosen as an input feature, as for most positions the number of corners detected would not change. However, for a small number of regions a small movement would greatly alter the number of corners detected. By selecting features from the range vector they provided the RBF with a rotationally invariant input. An average error of 23cm within an environment of 4.5m by 5m was obtained, but only position was determined, not orientation.

Using extracted features from a sensed environment has the disadvantage that if an obstacle obscures a feature, localisation becomes more difficult, in a similar manner to triangulation using beacons. Neural networks, however, have the advantage of being fault tolerant of the input provided and generalising from the training that they have received (Lippmann, 1987).

In our case, on the SENARIO wheelchair the localisation resolution required by the risk avoidance subsystem was ±1cm in each of the x and y directions and ±1°.

Hence we can quantise the number of possible positions in a single 4m by 4m room as (400x400x360) 57,600,000 positional orientations. The initial SENARIO test environment was an industrial workshop measuring 1920 x 1813 cm - which translates to (1920x1813x360) 1,253,145,600 positional orientations.

Due to the magnitude of the problem a special type of neural network was required, which needed to be able to handle both the complex target pattern (the 720 element range vector produced by the scanner) and the large number of possible target instantiations (the 1,253,145,600 positions of the wheelchair) in the environment.

Initially a Stochastic Diffusion Network (Bishop, 1989) was investigated as a possible candidate. Previous research (Bishop & Torr, 1992) had indicated that a SDN is capable of solving best fit constraint satisfaction problems with an efficiency dependent on the ratio of the number of neurons, or cells, employed to the search space size. However as the search space is so large in this application, the available computing power (a 100MHz 486DX PC card) was not sufficient to locate the wheelchair in the time available. To speed up the process a focussing mechanism was added, enabling the network to find the best fit of the scanner data to the environment model in real-time - the *Focussed Stochastic Diffusion Network* (FSDN).

2. The Stochastic Diffusion Network (SDN)

A general description of the SDN operation is given below - the FSDN is a simple extension of it. Note that in this description references to position are with respect to the abstract SDN search space, rather than the real world position of an AGV.

SDN is used to locate the best fit of a given data model, in the case of SENARIO data from the laser range sensors, within a given search space - here the set of possible wheelchair positional orientations in the environment. Both search space and model are defined by ‘Atomic Data Units’ (ADUs) - here range vectors, which are the micro-features from which they are both comprised. In the example of Figure 2, the model is defined by a set of eight such range vectors. A simple Boolean function on a distance measure between specified ADUs of the model and the ADUs at the same position in the search space will thus define the presence or absence of the micro-feature at that location. In the example in Figure 2, it is easily observed that micro-feature [8] is not the same length in the model and the example mapping, hence the feature is said to be ‘absent’ at that mapping.

A naive search to locate the model (range data from the scanner) within our search space (quantised positional orientations in the environment map) would thus involve evaluating the distance metric at each possible positional orientation of the model in the search space - the best fit of the model being the location with the smallest distance measure. In Figure 2, the map is quantised into 23x23 [ie. 529] possible XY positions (illustrated by the grey dots). Assuming eight possible angular orientations, this would result in a search space of thus 529x8 [ie. 4232] positional orientations. Note, the real SENARIO test environment at ZENON SA was defined by a search space comprising of 1,253,145,600 positional orientations. Using a naive sequential search on a space this size, (even assuming 10,000 comparisons per second) it would take approximately 1.2×10^6 seconds to find the ‘best-fit’ position.

Stochastic Diffusion is performed in parallel by a finite set of simple processing units called cells. Each cell is characterised by its activity and a pointer to a position in the search space. A cell is active if there is evidence that the model is present at this mapping in the search space.

Stochastic Diffusion Search can be considered as a competitive co-operative process in which all cells independently evaluate partial solutions to the problem by looking for specific model ADUs at specific positions in the search space. In the example shown in Figure 2 a partial solution is generated if an individual ADU at the position pointed to by the cell mapping (e.g. [8]) is the same length as the corresponding ADU from the scanner ([8]).

Once a partial solution is found by a cell, the cell becomes active and competes for greater allocation of search resources. A mapping with good fit to the data model has a higher chance of attracting more cells than a mapping with poor fit. In this way cell competition transforms smoothly into cell co-operation, as more and more cells are attracted to explore mappings with good fit to the data model.

The competition for co-operation mechanism ensures that potential positions of the target are examined independently, with the most promising mappings, (those active over a number of iterations), attracting most of the computational resources (i.e. being processed by most cells).

The ‘best-fit’ of the target will emerge from independent, parallel evaluation of potential positions in the search space, as cells gradually disregard misleading partial matches. It follows that cells begin to cluster over interesting positions in the search space as soon as the first cells pointing to these positions start to transfer information to others. Stochastic Diffusion, the mechanism responsible for this spread of information, is probabilistic in nature and consists of stochastically transferring mapping information from active ones to inactive cells. It involves several stages and can be summarised in the form of the following algorithm:

```
INITIALISATION PHASE;
WHILE NOT TERMINATION DO
    BEGIN
```

TESTING PHASE;

DIFFUSION PHASE;

END;

- The **initialisation phase** is performed by assigning to the cells random positions in the search space.
- During the **testing phase**, positions pointed at by cells are evaluated by comparison of randomly chosen ADUs from the data model and from the search space pointed to by the cell. If the comparison is successful the cell becomes 'active', otherwise it remains 'inactive'.
- During the **diffusion phase**, positions of active cells can cross to inactive cells. Positions pointed at by inactive cells are redefined as follows:

  "*each inactive cell chooses randomly another cell, and if the latter one is active, its position is copied by the former; otherwise a new, randomly chosen position in the search space is assigned to the cell*".
- The process iterates until **termination** conditions are fulfilled. Bishop & Torr (1992), proposed the following equilibrium based termination condition:

  "*if, in an iteration, the number of cells pointing to the same position within the search space exceeds a given threshold and remains constant within specified bounds over a number of iterations, then the network is said to have reached equilibrium and the solution is given by the common position of these cells*".

2.1 Implementation

Below, a description of an SDN is given, applied to the wheelchair localisation problem. The SDN consists of a number of testing units, or cells, which evaluate possible localisation mappings independently of each other. These mappings are initially uniformly randomly selected from (x, y, θ) space. Cell mappings are tested by comparing map information at (x, y, θ) with current ‘real’ system range data from the scanners. A cell is defined as ‘active’ if all the difference between the ranges is within a predetermined tolerance.

After testing active cells diffuse information to other cells across the network. Each ‘inactive’ cell polls a cell at random and *if this cell is active*, copies its mapping. If it is not it selects an entirely new random location from (x, y, θ) space.

Active cells define possible locations of the wheelchair in the environment, however, as each cell only tests one random micro-feature of the model at the position defined by the cells mapping, activity does not *necessarily* define the globally best solution. However, as at each iteration a cell evaluates a different model micro-feature at its specified mapping, with each iteration that it remains active, the probability that this position is the best, rather than one causing false positive evaluations, increases. Also, the longer that cell remains active the more likely it is to propagate its mapping to other cells via the Diffusion mechanism.

It has been shown that without noise the best-fit mapping will rapidly diffuse to all cells and that in noisy conditions the best-fit location is defined by the largest group of active cells with the same mapping at network equilibrium (Nasuto & Bishop, 1997).

2.2 Termination Condition

Let S denote a given search space. We assume that S is discrete, e.g. there exists a finite number of possible instances of the range vector within the search space. Further, let M denote the size of the given data model. Let f_n^s denote the number of active cells pointing to the same location s in the search space S in the nth iteration of the algorithm. It is easy to see that the following condition is fulfilled:

$$\sum_{s \in S} f_n^{\,s} \leq M$$

SDN and FSDN function by assigning potential locations of the range vector within the search space to cells or equivalently by assigning to locations in S a number of active cells. Let z_n denote the maximal number of active cells pointing to the same location, $s_n^{\,z} \in S$ in the search space, i.e. $z_n = max_{s \in S}\ (f_n^{\,s})$. Then, (from Nasuto & Bishop, 1997), the definition of convergence of SDN and FSDN have the following formulation:

$$(\exists\ a > 0)\ ($$

In practise a Stochastic Diffusion Network will have reached an equilibrium if there exists a time instant n_0 and an interval (specified by **a** and **b**) such that after n_0 iterations the maximum number of cells pointing to the same location will enter and remain within the specified interval for a predetermined period.

ie.

$$\exists\ ((2b < M) \text{ AND } (b + a \leq M) \text{ AND } (a - b \geq 0))\ \exists\ \ \forall\ \ (|z_{n\,-\,a}| < b)$$

Note also that the above definition does not require convergence of the process to a fixed point. Indeed, the interval specified by **a** and **b** defines a tolerance region. All fluctuations of the maximal number of cells pointing to the same location in the search space are discarded as not important if they occur within this interval. The conditions for **a** and **b** exclude the trivial case in which we would ask only, that

$0 \leq z_n \leq M.$

3. The Focused Stochastic Diffusion Network (FSDN)

FSDN is an extension of the SDN, but provides a faster solution to the localisation problem. Here, the search space is mapped onto a multi-resolution pyramid (see Figure 3). The size of the search space determines the number of ‘focusing levels’ that are required such that the effective search space is evaluated in real-time - the example map in Figure 3 shows the coarsest resolution to comprise of 4x4 (16) regions. In the real SENARIO environment map, focussing levels are defined such that the search space covered at the highest level is smaller than the smallest room - ensuring that it remains possible for cells to become active if the wheelchair is in that room.

Note that the FSDN does not try to find the exact wheelchair position. Instead it attempts to find a **region** that contains the position. FSDN search progresses by focusing - using ever decreasing region sizes, and reducing the comparison tolerance.

3.1 Initialisation

Where no positional cues are given *a priori*, the FSDN initialises each cell with a uniform random mapping into (x, y, θ) space. However, any suitable information that is available may be used to bias the uniform distribution. In SENARIO’s case, the passive radio beacons are placed within the environment at known (x, y) positions, and if one is detected then a number of FSDN cells are set to this position with the highest focus value in order to prime the search.

3.2 Testing

Cells initially randomly select a location within the search space. Positional information from each cell is used to generate artificial range data from the map. Thus data can be compared with the ‘real’ data from the laser range finders. The tolerance value that is used for comparison ranges depends upon the level of focusing currently employed by the cell. If the comparison is within tolerance and the cell is not already at the limit of focussing, it moves down a level. If the comparison is out of tolerance the cell moves up a focusing level. A cell that produces an out of tolerance comparison *at the top level of focusing is termed inactive*; otherwise it is active.

3.3 Diffusion

Diffusion occurs as in the SDN however in diffusion only mapping information is transferred between cells - the cell requesting the mapping begins its search centred around this location but at the top (coarse) focus level.

3.4 Termination criterion

Termination of the FSDN is determined using the same basic criterion as SDN (see 2.2), but counting the number of testing units that have reached the ‘finest’ (most accurate) focus level. At termination the final positional estimate is made by averaging active mappings at this level.

3.5 System Trial

The FSDN localisation sub-system was installed on a 486DX 100MHz processor card on the SENARIO wheelchair and evaluated initially at the headquarters of ZENON SA - the co-ordinating partner for SENARIO. The trial environment was a large industrial workshop (approximately 19.2m x 18.13m - see Figure 1). The time required to obtain the initial position to the required tolerance (±1cm; $\pm 1^0$), without using any a priori information, was 120 seconds. Subsequent position updates to the same tolerance (giving the FSDN knowledge of the previous position) took 8 seconds.

4. Discussion

The SENARIO wheelchair project has established that an FSDN, in conjunction with laser range sensors, is a viable, novel, and interesting method of solving the AGV localisation problem. However in general, there are several parameters that need to be finely tuned when using an FSDN in this domain. The most important being the number of cells used in the network, as this affects the number of iterations required for one cell to locate the correct wheelchair position. Implementing FSDN on a parallel machine, there is a linear speedup with the number of cells employed by the network, however on the single processor test machine a population of 200 cells was found to give the fastest convergence to wheelchair position. This FSDN parameter is highly coupled to the specific problem under investigation and would need to be empirically re-optimised if the technique were to be applied in another application.

4.1 Practical Problems

The most time-consuming operation in determining the location of the wheelchair is the generation of simulated range data for a given target position in the search space. Current cycle times for positional update in the prototype wheelchair are two minutes for initial localisation and eight seconds for subsequent positions, albeit this could be improved by using a more modern processor card.

The generation of simulated range vectors from the environment map is fundamental to the test phase of FSDN and although considerably speeded up by pre-processing, still dominates network convergence time. (The pre-processed map data provided the simulator with a list of just the walls that needed to be evaluated from a specific position - those within line of sight - rather than calculating the ranges to all walls within the wheelchair environment).

A fundamental problem concerning the use of range data to solve the localisation problem is symmetry. This occurs when the wheelchair is in a symmetric room, when data from the laser range sensors alone is not sufficient to obtain an unambiguous position. The solution to this problem currently used on the SENARIO wheelchair is to specify that the door into the wheelchairs room must be open when the first location is being determined. This method breaks the room symmetry and is only required when the wheelchair is first switched on, (or when it has been moved manually as the last known location is stored for retrieval when the wheelchair is switched on).

When using this system to solve the localisation problem within any large institution (e.g. a hospital), there is a further problem related to that of symmetry, caused by the physical similarity of rooms - as many will produce very similar range vectors. The method adopted to overcome this problem in the SENARIO project has been to use inductively coupled radio beacons, which can transmit a unique identity to the wheelchair. The location of each beacon is pre-determined, so when a beacon code is received, a number of cells are loaded with this location to prime the FSDN to the correct (x, y) region of the search space.

4.2 Further Research

Research is ongoing to determine a formal description of both the simple Stochastic Diffusion Network (SDN) and the Focussed Stochastic Diffusion Network (FSDN), which will enable the optimal selection of all network parameters (such as the number of cells) to be made rigorously, rather than by the empirical methods currently employed.

5. References

Beattie, P.D., 1995, SENARIO: SENsor Aided intelligent wheelchaiR navigatIOn, IEE Power Division, Colloquium on “New developments in electric vehicles for disabled persons”, 1995/055, pp. 2/1-2/4.

Bilgiç, T. and Türksen, L.B., 1995, Model-based localization for an autonomous Mobile Robot Equipped with Sonar Sensors, IEEE Int. Conf. Systems Man & Cybernetics, Intelligent Systems for the 21st Century, Vol. 4, pp. 3718-23.

Bishop, J.M., (1989), Stochastic Searching Networks, Proc. 1st IEE Conf. Artificial Neural Networks, pp. 329-331.

Bishop, J.M., Keating, D.A. & Mitchell, R.J., (1995), A Compound Eye for a simple Robotic Insect, Proc. Weightless Neural Network Workshop 1995, pp. 41-47.

Bishop, J.M. and Torr, P., 1992, The Stochastic Search Network. in Neural Networks for Images, Speech and Natural Languages, (eds. Linngard, R., Meyers, D.J. and Nightingale, C.), Chapman & Hall, New York.

Bourhis, G., Horn, O. and Agostini, Y., 1994, Location and high level planning for a Powered Wheelchair, IEEE International Conference on Systems, Man and Cybernetics, pp. 2629-2634.

Byler, E., Chun, W., Hoff, W. and Layne, D., 1995, Autonomous Hazardous Waste Drum Inspection Vehicle, IEEE Robotics and Automation Magazine, Vol. 2, No. 1, pp. 6-17.

Cox, I.J., 1991, Blanche - An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle”, IEEE Transactions on Robotics and Automation, Vol. 7, No. 2, pp. 193-204.

Drumheller, M., 1987, Mobile Robot Localization Using Sonar, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 2, pp. 325-332.

Durieu, C., Clergeot, H. and Monteil, F., 1989, Localization of a Mobile Robot With Beacons Taking Erroneous Data Into Account. *IEEE Proceedings, Conference on Robotics and Automation.* Vol. 2, pp. 1062-68.

Evans, J., Krishnamurthy, B., Barrows, B., Skewis, T. and Lumelsky, V., 1992, Handling Real-World Motion Planning: A Hospital Transport Robot”, IEEE Control Systems, pp. 15-19.

Figure 1. ZENON SA. The first wheelchair test environment.

Figure 2. Example environment map showing the actual position of the wheelchair and range vectors from its scanners together with the mappings from an SDN network of ten cells - example range vectors for one mapping being illustrated.

Figure 3. The Multi-Resolution Pyramid. Focus regions initially cover a large area of the search space. Regions (delineated by thin lines) and are sub divided as the cell focuses in upon the correct location until the desired resolution is obtained.