Monograph:
Nehmzow, U., Kerr, D. and Billings, S.A. (2009) Accurate robot simulation. Research Report. ACSE Research Report no. 992 . Automatic Control and Systems Engineering, University of Sheffield

Accurate Robot Simulation

Ulrich Nehmzow[1] **Dermot Kerr**[1] **S A Billings**[2]

[1]Intelligent Systems Research Centre, University of Ulster, Londonderry, BT48 7JL, UK

[2]Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, S1 3JD, UK

Abstract

Robot simulators are valuable tools for researchers to develop control code in a fast and efficient manner without spending time setting up physical experiments. Most simulators, however, do not model the real world accurately. As a consequence, when a program is run on a real robot it may behave differently from when run in simulation. In this paper we present a method of developing a robot simulator that models the operation of a real robot in a real environment accurately, using real robot data and system identification to construct the simulator's model.

1. Introduction

When considering the behaviour of a robot, there are three components that influence the system: (i) robot hardware, (ii) the program being executed, and (iii) the robot's environment. Thus, the interaction of these three components, as illustrated in Figure 1, constitutes a highly complex system from which the overall robot behaviour emerges.

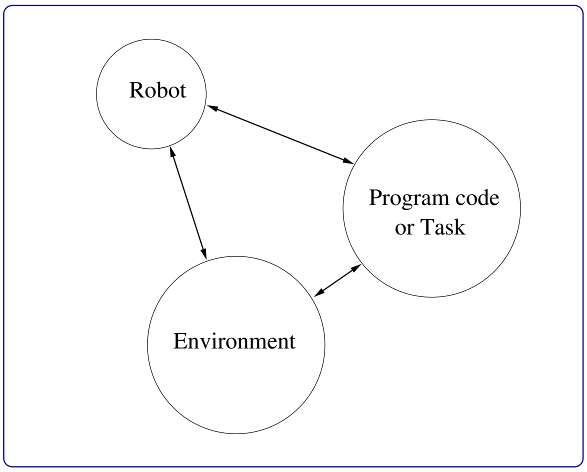

Figure 1: Overall robot behaviour emerges from the interaction between robot, task and environment

The common procedure to develop a control program is for the robot programmer to write the program "as best as possible" whilst considering the desired task. However, it is unlikely that the desired behaviour will be obtained immediately, as the robot programmer has to make (often idealistic, simplifying) assumptions about the robot's hardware and properties of the environment. Typically, the desired robot behaviour is eventually obtained by a process of iterative refinement of the control program.

Robot simulators are therefore useful for researchers to develop and test new controllers. Simulators can save time and effort by concentrating on the development of the controller, rather than spending time on i) setting-up real world experiments and ii) waiting for experiments to complete, as using a simulator allows the researcher to run experiments faster than real-time. Additionally, simulators are the only way to provide a consistent repetition of experiments.

Traditional trial-and-error-based approaches of programming robots lack a rigorous design methodology, they are costly, time consuming and error prone. In this paper we present an alternative, describing a design method for a mobile robot simulator that accurately models and simulates a physical robot's interaction with its environment, performing a particular task.

The overall aim of this research is the *automatic* generation of control code, without the need for an experienced robot engineer to perform iterative refinement and trial-and-error procedures. As a step towards this goal, in this paper we present our experiments to create accurate robot simulators that make precise predictions of robot sensor measurements based on the robot's position in the environment.

1.1 Motivation

A number of existing robot simulators exist within the robotics community, some are generic such as Player-Stage (Gerkey et al., 2003) and Webots (Michel, 2004), others are platform dependent such as the Kephera simulator (Michel, 1996). The ability of these robot simulators to predict accurately depends on three models: i) the robot model, ii) the task model, and iii) the environment model. The environment model provides the robot's sensory perception based on the robot's position and orientation and is of the main interest to this work. This model is dependent on the structure of the environment and the properties of the materials the environment is composed of. However, being generic, they are based on general assumptions of environment properties and fail to model extreme perceptions such as specular reflections. As argued in (Lee et al., 1998, Kyriacou et al., 2008) we believe that faithful simulation is only obtainable when *specific* environmental scenarios are modelled using data obtained from *real* environments.

1.1.1 Possible Approaches

In the simulator developed by Lund and Miglino (Lund and Miglino, 1996), logged data is stored in a lookup table and values are interpolated to predict sensory perception at unvisited locations. In (Lee et al., 1998) a neural network is used to model the input-output relationships between robot location and sensor perception. This approach works well, however, neural networks produce an opaque model that cannot be used to investigate the characteristics of the inputs and outputs further (such as stability analysis, sensitivity of the behaviour to particular sensor (Saltelli et al., 2000) or environmental features, sensor redundancy etc). Additionally, the structure of a neural network does not allow straightforward comparison between different models. Here, we therefore propose a *transparent* modelling method, which is most closely related to the work presented in (Kyriacou et al., 2008). In Kyriacou's paper, a robot simulator is built using mathematical models representing the sensor perception based on the robots location. Our work differs from (Kyriacou et al., 2008), however, in that we create parsimonious sensor models based on the robot's orientation φ, rather than having many separate sensor models for each (x, y) location in the robot's environment.

2. Modelling Method

We propose the use of system identification, as used in (Kyriacou et al., 2008), to develop accurate robot simulation. We determine the relevant input-output parameters of the sensor simulation process — the accurate simulation of laser perception based on a robot's position, and validate the resulting laser perception against the robot's real laser perception along a (separately logged) validation path.

In this work we adopt the use of system identification, rather than alternative modelling approaches such as artificial neural networks or lookup tables, because it offers numerous benefits: the obtained models are compact, as they consist of a single polynomial with a small number of terms; and the obtained models are transparent and can be examined using standard techniques such as sensitivity analysis (Saltelli et al., 2000).

NARMAX (Non-linear Auto-Regressive Moving Average model with eXogenous inputs) is a parameter estimation methodology to identify the important model terms and their associated parameters of a non-linear polynomial model of an unknown non-linear dynamic system. The NARMAX methodology breaks the modelling problem into the following steps: i) Structure detection, ii) parameter estimation, iii) model validation, iv) prediction, and v) analysis. A detailed procedure of these steps is presented in (Chen and Billings, 1989, Billings and Voon, 1986, Korenberg et al., 1988), and details of the application of NARMAX to mobile robot simulation in (Kyriacou et al., 2008).

In this work we obtain a model that relates the robot's positional location to its laser perception. The fact that we obtain the model as a transparent polynomial representation makes the model easily and accurately transferable to any robot platform with the same sensor type and configuration. Another benefit of the compact mathematical form of the model is that it can be directly implemented in any computer language using standard mathematical libraries. This minimises the time to translate programs between robot platforms and languages. Increased execution time and reduced memory requirements are also obtained due to the task code being condensed into a single polynomial with the modelling process.

3. Experiments

The experiments in this paper where carried out in the robotics arena of the Intelligent Systems Research Centre in the University of Ulster. The robotics arena measures 100 m^2 and is equipped with a powered floor, a *Vicon* motion tracking system and a large number of robots. In all the experiments described in this paper we used the Metralabs *SCITOS G5* autonomous robot *Swilly*, shown in Figure 2. The robot is equipped with 24 sonar sensors distributed around its circumference and a *SICK* laser range finder, which scans the front of the robot ([0°,270°]) with a radial resolution of 0.5°. In our experiments the laser range finder was configured to scan the front semi-circle of the robot in the range ([0°,180°]).

3.1 Experimental Set-Up

During the experiments the input from the robot's laser, position, orientation, transitional and rotational velocities were logged every 250 ms. In addition, the robot's actual x, y, z positions obtained from the *Vicon* motion tracking system are logged simultaneously at 50 Hz, at this speed the positional error was less than 1 mm.

The configuration of the robotics arena used in the experiments is illustrated in Figure 3, where the experimental set-up is built using various cardboard cartons.

Figure 2: *Swilly*, THE METRALABS SCITOS G5 MOBILE ROBOT USED IN THE EXPERIMENTS

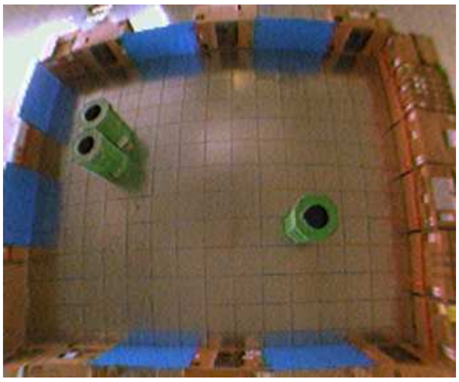

Figure 3: ROBOT ARENA SETUP

3.1.1 The Sensorgraph

In our experiments we first construct what we term a *sensorgraph* — a detailed log of sensor perceptions obtained in the environment — by using the robot to collect data from the environment that is being modelled. The data that is collected during this process are the robot's position, orientation, laser perception and ground truth position (obtained using the *Vicon* motion tracking system). A random walk obstacle avoidance program using the laser sensor is run on *Swilly*, involving obstacle avoidance and random rotations to the robot's movement at predefined intervals. The basis of introducing the random rotations is to try and ensure that the robot explores as much of the environment as possible at different orientations. In Figure 4 we show the actual positions where *Swilly* logged data, over 1400 data samples in these experiments.

It is important that the robot has logged adequate data throughout this data collection stage, as the obtained model's accuracy will depend on the data contained in the sensorgraph. Therefore, to ensure that we have a good distribution of values we computed histograms for the robot's position obtained using the *Vicon* tracking system along the x-axis and y-axis (Figures 5 and 6). Although the histograms show a denser distribution of the *sensorgraph* data at the arena edges, they show that all possibly x and y locations are equally often visited (uniform distribution) — a desirable property for a sensorgraph.

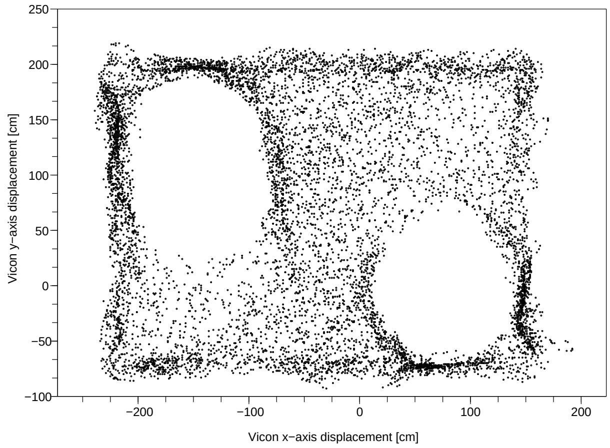

Figure 4: DATA SAMPLING POINTS

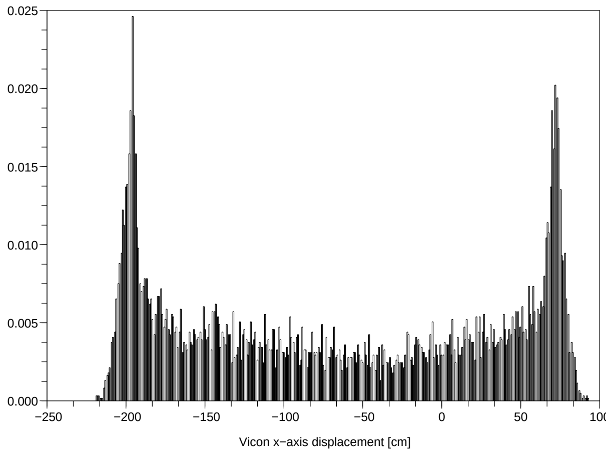

Figure 5: HISTOGRAM OF VISITED x POSITIONS

It is also important to consider the robot's heading, because the modelling process needs to consider all possible orientations of the robot. In Figure 7 we illustrate the histogram representing the rotational angles where the robot has logged data as contained in the sensorgraph, again a good almost-uniform distribution.

After the data was collected we median-filtered the laser data over 30° segments. Thus, rather than 360 laser readings ([0°,180°] × 0.5° resolution), we then used the six median-filtered segments

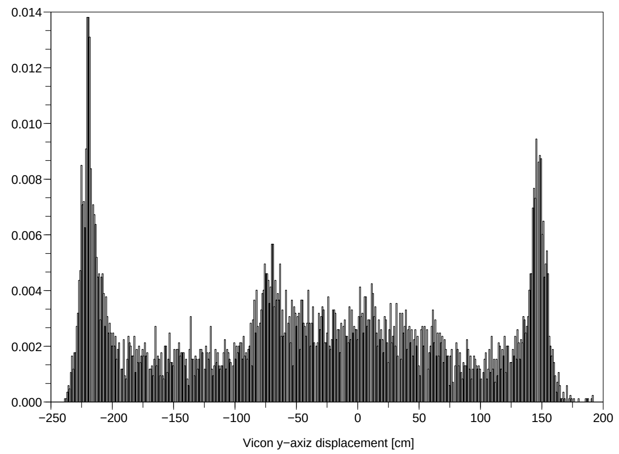

Figure 6: Histogram of visited y positions

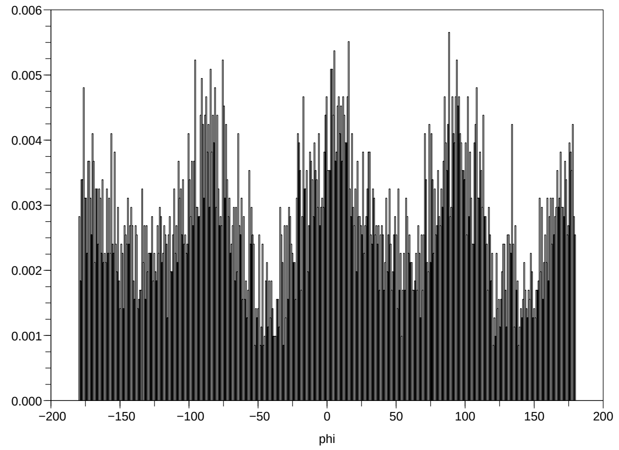

Figure 7: Histogram of headings φ at data-logging points

$$
\begin{aligned}
L_{15} &= median[0°, 30°] \\
L_{45} &= median[30°, 60°] \\
L_{75} &= median[60°, 90°] \\
L_{105} &= median[90°, 120°] \\
L_{135} &= median[120°, 150°] \text{ and} \\
L_{165} &= median[150°, 180°]
\end{aligned}
$$

as input to our modelling process.

3.1.2 Models For Different φ

After logging the sensorgraph data, it is required to compute a model which predicts the robot's sensor perception, given the robot's position and orientation. When computing this model we need to consider all the possible (x, y) positional locations the robot may visit, as well as the robots orientation φ at these positions. The dimensionality of this space is very high, and in order to manage the task of constructing a model we have restricted the number of models to only 12 models, modelling 30° segments, thus covering the entire 360° range of possible robot headings with 12 models. Put differently, we constructed 12 models, one for each 30° segment, of the form $L_k = f(x, y) \forall \varphi = k$, where L_k is a model of the laser reading when the robot assumes a heading φ of k degrees.

3.1.3 Validation Trajectories

To validate the robot's predicted sensor perceptions based on the robot's position (x, y) and orientation φ, a number of independent validation runs were logged by manually driving the robot along a novel path immediately after the logging of the sensorgraph. During these validation runs the robot's position (obtained using the *Vicon* tracking system) and laser perception were logged. In Figure 8 and Figure 9 we show the trajectory of two validation runs, along with the original sampling points used to obtain the sensorgraph.

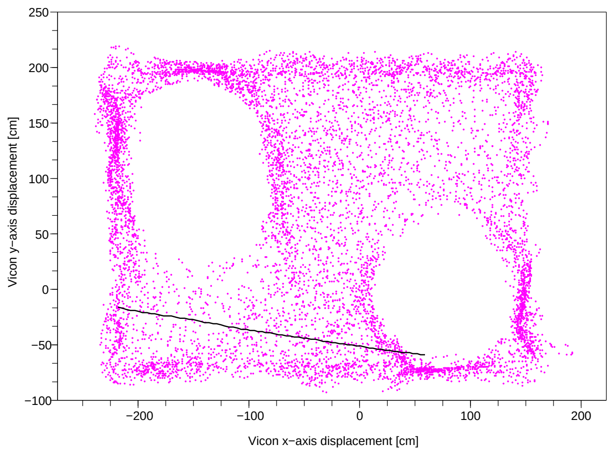

Figure 8: Validation trajectory 1

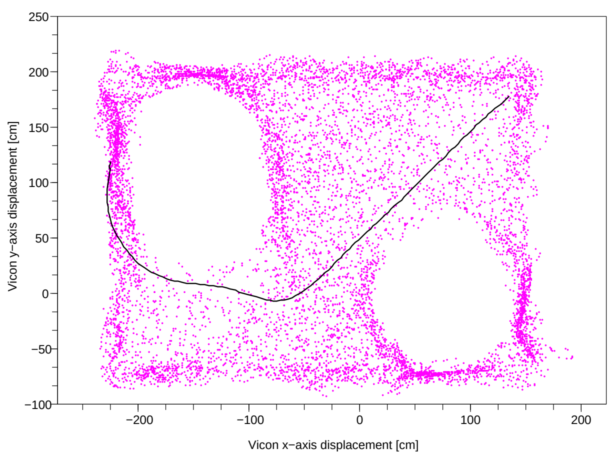

Figure 9: Validation trajectory 2

These validation runs allow us to compare, measure and validate how the actual real sensor perception compares with the model's predicted values along a new trajectory.

3.2 Initial Results

After collection of the data in the sensorgraph we use the NARMAX system identification procedure to estimate the robot's laser perception as a function of the the robot's location, obtained using the *Vicon* tracking system and using the logged training data. As described earlier, 12 polynomials were obtained in total, expressing the laser perception of the robot as a function of the robots location and heading to the nearest 30°. The models were chosen to be of fourth degree and no regression was used in the inputs and outputs.

For example, for $\varphi = -75° \pm 15°$ and laser segment L_{75} this resulted in a NARMAX polynomial structure containing 9 terms

$$\begin{aligned} L_{p75}/cm = &+223.74 \\ &+ 1.47 * x \\ &+ 0.68 * y \\ &- 0.002 * x^2 \\ &- 0.001 * y^2 \\ &- 0.0001 * x^3 \\ &- 0.00000036 * x^4 \\ &+ 0.00000006 * y^4 \\ &+ 0.000012 * y * y^2. \end{aligned}$$

The difference between model prediction and true value is shown in Figure 10.

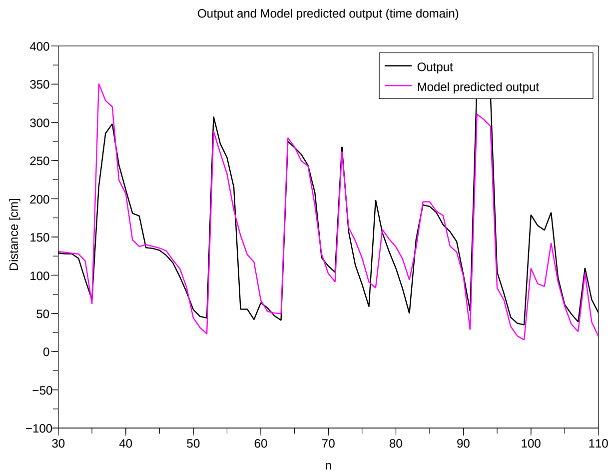

Figure 10: Actual and predicted laser readings along validation trajectory 1

For $\varphi = -165° \pm 15°$ and laser segment L_{75} this resulted in a NARMAX polynomial structure containing 4 terms

$$\begin{aligned} L_{p75}/cm = &+234.84 \\ &+ 0.85 * x \\ &+ 0.30 * y \\ &+ 0.0014 * x * y, \end{aligned}$$

the model validation is illustrated in Figure 11.

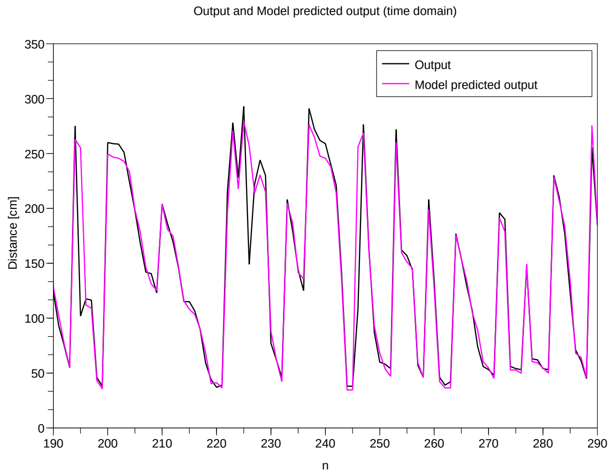

Figure 11: Actual and predicted laser readings along validation trajectory 2

In a similar manner we computed all 12 models for the range of orientations φ and for each laser segment.

3.2.1 Assessment 1

To assess the performance of the obtained polynomial models we used the validation trajectory 1 (Figure 8). In Figure 12 we plot the predicted laser values for the segment $\tilde{L_{75}}$ against the actual laser values L_{75}.

In Figure 13 we plot the predicted laser values for the segment $\tilde{L_{105}}$ against the actual laser values L_{105}.

Mean error and standard error for validation trajectory 1 are given in Table 1.

3.2.2 Assessment 2

We also assessed the performance of the obtained polynomial models for a more complex non-linear trajectory, such as the one given in validation trajectory 2 (Figure 9). Figure 14 shows the predicted laser values for the segment $\tilde{L_{75}}$ against the actually logged laser values L_{75}.

Figure 15 shows the predicted laser values for the segment $\tilde{L_{105}}$ against the actual laser values L_{105}.

The mean error and standard error for validation trajectory 2 are given in Table 2.

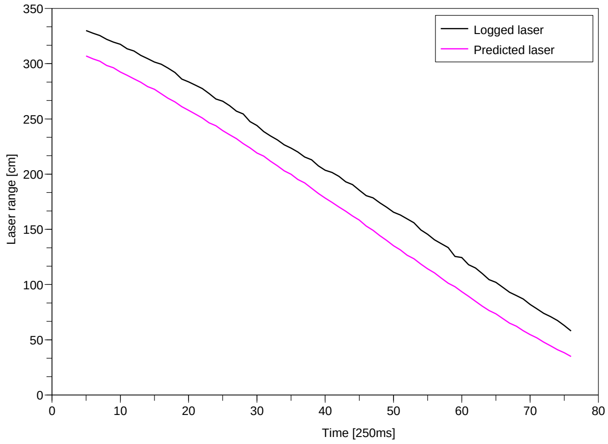

Figure 12: PREDICTED LASER $\tilde{L}_{75}$ VALUES AGAINST ACTUAL LOGGED LASER L_{75} VALUES FOR VALIDATION TRAJECTORY 1

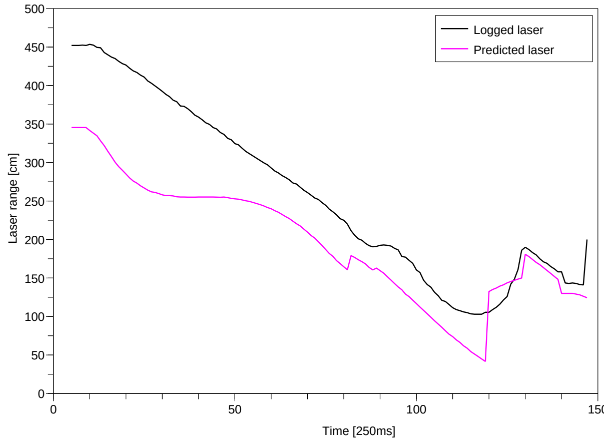

Figure 14: PREDICTED LASER $\tilde{L}_{75}$ VALUES AGAINST ACTUAL LOGGED LASER L_{75} VALUES FOR VALIDATION TRAJECTORY 2

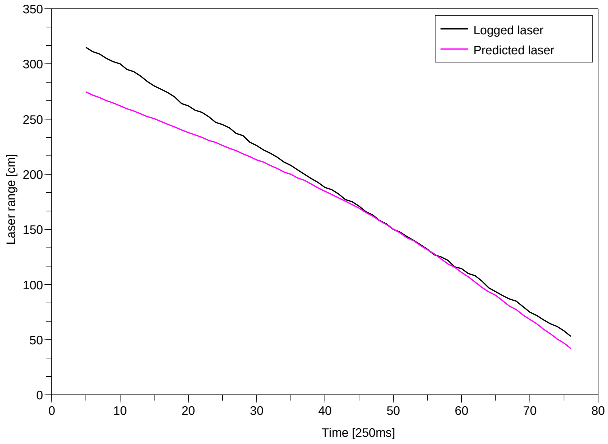

Figure 13: PREDICTED LASER $\tilde{L}_{105}$ VALUES AGAINST ACTUAL LOGGED LASER L_{105} VALUES FOR VALIDATION TRAJECTORY 1

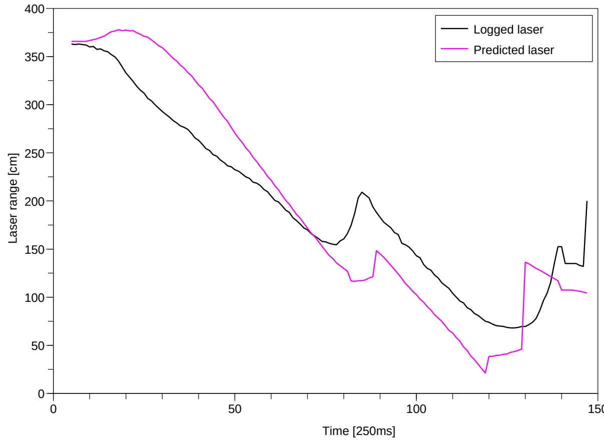

Figure 15: PREDICTED LASER $\tilde{L}_{105}$ VALUES AGAINST ACTUAL LOGGED LASER L_{105} VALUES FOR VALIDATION TRAJECTORY 2

Table 1: ABSOLUTE MEAN ERROR AND STANDARD ERROR IN CENTIMETRES FOR VALIDATION TRAJECTORY 1

Laser	Mean and Standard error [cm]
$\tilde{L}_{15}$	60 ± 5
$\tilde{L}_{45}$	53 ± 5
$\tilde{L}_{75}$	31 ± 4
$\tilde{L}_{105}$	17 ± 4
$\tilde{L}_{135}$	21 ± 2
$\tilde{L}_{165}$	20 ± 2

Table 2: ABSOLUTE MEAN ERROR AND STANDARD ERROR IN CENTIMETRES FOR VALIDATION TRAJECTORY 2

Laser	Mean and Standard error [cm]
$\tilde{L}_{15}$	62 ± 3
$\tilde{L}_{45}$	46 ± 4
$\tilde{L}_{75}$	66 ± 4
$\tilde{L}_{105}$	39 ± 3
$\tilde{L}_{135}$	29 ± 4
$\tilde{L}_{165}$	27 ± 3

Further Statistical Evaluations We also performed numerical tests to determine the performance of the simulator for this second validation trajectory, and compared the error of the simulator model against the error of a "random guess" simulator that generates random predictions from the range of valid laser measurements. The Mann-Whitney U-test (Snedecor and Cochran, 1989, Barnard et al., 1993) was performed on both pairs of distributions in order to check if they are significantly different or not. Both tests indicated that the distributions are different at the 5% significance level as illustrated in Figure 16 for the laser segment $\tilde{L}_{75}$ and in Figure 17 for the laser segment $\tilde{L}_{105}$. In other words, the obtained model performs significantly better than a random guess.

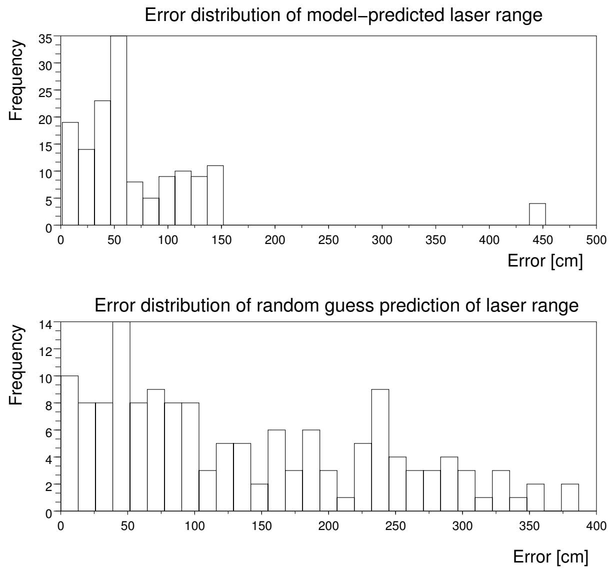

Figure 16: U-STATISTIC ILLUSTRATING COMPARISON OF PREDICTED LASER ERROR WITH RANDOM GUESS ERROR FOR LASER L_{75}

The Spearman rank correlation coefficients between actual laser perception and predicted laser perception are given in Table 3, they are all significant ($p < 0.05$).

Table 3: SPEARMAN RANK CORRELATION COEFFICIENTS BETWEEN PREDICTED AND ACTUAL LASER READINGS FOR VALIDATION TRAJECTORY 2. ALL CORRELATIONS ARE STATISTICALLY SIGNIFICANT ($p < 0.05$).

Laser segment	SR
1	0.7264930
2	0.3045180
3	0.9775737
4	0.9275737
5	0.7720901
6	0.7360612

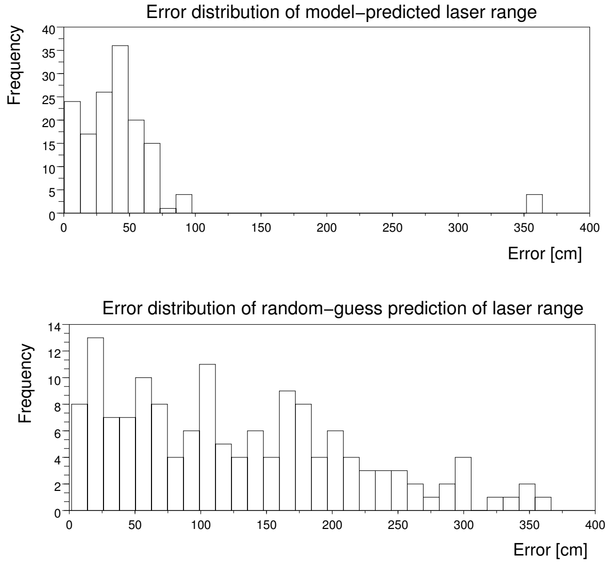

Figure 17: U-STATISTIC ILLUSTRATING COMPARISON OF PREDICTED LASER ERROR WITH RANDOM GUESS ERROR FOR LASER L_{105}

4. Summary And Conclusions

4.1 Summary

In this paper we have presented an approach to model a robot's laser perception based on position and orientation, using the NARMAX model estimation methodology. The mobile robot is used to explore the environment whilst logging sensor perception and its actual position. This data is then used to construct models that predict the robot's perception based on its position and orientation. To evaluate the performance accuracy of the obtained models we compared the robot's real sensor perception along two independent validation runs with the sensor perception predicted by our model.

4.2 Conclusions

We have shown how to model a robot's laser perception as a function of its position, using compact polynomial models, and how the NARMAX modelling approach can be used to produce transparent mathematical functions that can be related directly to the modelling task.

This method of simulating the robot's perception has a number of important benefits. Simulator development is fast and the obtained model is very compact. The model is transparent, i.e. represented as a polynomial function that can be analysed mathematically.

4.3 Future Work

The experiments presented in this paper are first steps towards our goal of automatic generation of robot control code. The goal of this research is to have a robot observe a human performing a task, and for the robot to generate control code automatically from this observation to perform the same task. The experiments presented here illustrate that it is possible to create a model of sensor perception as a function of the robot's position within the environment. The next stage in this research is to model a human task, using the *VICON* tracking system. Using the demonstrator's trajectory we will estimate the robot's perception at these positions, using the method described in this paper, to create a new model that mimics the task performed by the human demonstrator.

5. Acknowledgements

The authors gratefully acknowledge the support of the Leverhulme trust under grant number F00430F.

References

Barnard, C., Gilbert, F., and McGregor, P. (1993). *Asking questions in biology.* Longman, Harlow, UK.

Billings, S. and Voon, W. S. F. (1986). Correlation based model validity tests for non-linear models. *International Journal of Control*, 44:235–244.

Chen, S. and Billings, S. A. (1989). Representations of non-linear systems: The narmax model. *Int. J. Control*, 49:1013–1032.

Gerkey, B., R.Vaughan, and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics.*

Korenberg, M., Billings, S., Liu, Y., and McIlroy, P. (1988). Orthogonal paramter estimation for non-linear stochastic systems. *Int. J. Control*, 48:193–210.

Kyriacou, T., Nehmzow, U., Iglesias, R., and Billings, S. A. (2008). Accurate robot simulation through system identification. *Robotics and Autonomous Systems*, 56:1082–1093.

Lee, T., Nehmzow, U., and Hubbold, R. J. (1998). Mobile robot simulation by means of acquired neural network models. In *ESM*, pages 465–469.

Lund, H. H. and Miglino, O. (1996). From simulated to real robots. In *Proceedings IEEE International Conference on Evolutionary Computation.*

Michel, O. (1996). Kephera simulator version 2.0 user manual.

Michel, O. (2004). Webots: Professional mobile robot simulation. *Advanced Robotic Systems*, 1:40–43.

Saltelli, A., Chan, K., and Scott, E. M. (2000). *Sensitivity analysis.* Wiley.

Snedecor, G. W. and Cochran, W. G. (1989). *Statistical Methods.* Iowa State Press, 8th edition.