Towards Automated Code Generation for Autonomous Mobile Robots

D. Kerr
Intelligent Systems
Research Centre
University of Ulster
Londonderry, BT48 7JL, UK

U. Nehmzow
Intelligent Systems
Research Centre
University of Ulster
Londonderry, BT48 7JL, UK

S.A. Billings
Department of Automatic Control
and Systems Engineering
University of Sheffield
Sheffield, S1 3JD, UK

Abstract

With the expected growth in mobile robotics the demand for expertise to develop robot control code will also increase. As end-users cannot be expected to develop this control code themselves, a more elegant solution would be to allow the end-users to *teach* the robot by demonstrating the task.

In this paper we show how route learning tasks may be "translated" directly into robot control code simply by observing the task. We show how automated code generation may be facilitated through system identification — which algorithmically and automatically transfers human behaviour into control code, using transparent mathematical functions. We provide two route learning examples where a mobile robot automatically obtains control code simply by observing human behaviour, identifying it using system identification, and copying the behaviour.

Introduction

Motivation

Mobile robotics will play an ever more important role in the future. We expect one growth area to be service robotics, especially home care robots for the elderly and infirm. Other important growth areas will be entertainment robotics and games, as well as security applications.

All of these applications require some high-level, sophisticated programming, but the bulk of the programming work will be "standard" components of robot control, that will consume a lot of programmer resources — resources that could be better used.

In this paper we show that it is possible for trajectory learning to obtain robot control code *automatically*, through system identification (Akanyeti et al., 2007): Control code was obtained by observing a human demonstrator following the desired route, and by translating his behaviour *directly* into code, *without programming.*

Learning by demonstration is by now a widely used technique in robotics (see, for instance, (Demiris, 2009) and (Demiris and Dearden, 2005)). In terms of application (route learning), (Coates et al., 2008) are perhaps the most interesting here to mention: an expert was used to control a model helicopter, the desired optimal behaviour was obtained and modelled from a number sub-optimal demonstrations performed by the expert. This model was then used to control the helicopter. In contrast to our work, (Coates et al., 2008) use a specialist to provide the demonstration, and their control model incorporates *a priori* knowledge such as that the helicopter has to remain stationary for certain manoeuvres. In our experiments the trainer is not an expert in the task, and only needs the ability to demonstrate the behaviour to the robot.

The experiments in this paper develop our approach further, their purpose is to show that even more complex behaviour behaviour of an agent — route learning in this case — can the "translated" directly into robot control code, namely by observing it, identifying it, using system identification, and using the identified model of the observed behaviour control the robot (Figure 1).

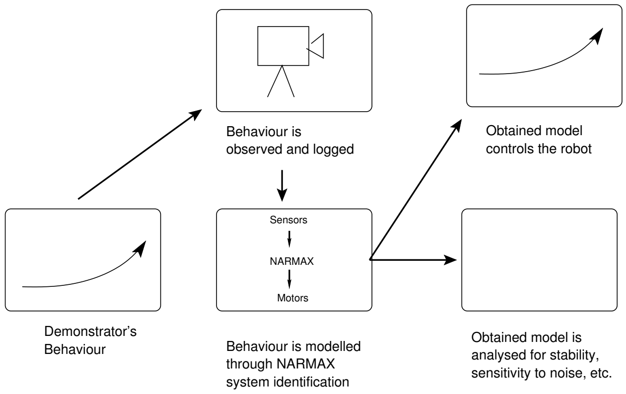

Figure 1: The "behaviour copier": a behaviour is observed and subsequently identified using system identification. The obtained model is then used to control the robot directly, no human intervention is required at any point (other than that the human demonstrates the desired behaviour to the robot).

In essence the experiments reported here form a "behaviour copier", which produces a canonical "carbon copy" of an observed behaviour that can be used to control an autonomous mobile robot.

Approach

We have used a NARMAX approach (Billings and Chen, 1998) to obtain the models we need for automated programming of a robot, because

- The Narmax model itself provides the executable code straight away,
- The model is analysable, and gives us valuable information regarding
  - How the robot achieves the task,
  - Whether the model is stable or not,
  - How the model will behave under certain operating conditions, and
  - How sensitive the model is to certain inputs, i.e. how "important" certain input are.

NARMAX system identification The NARMAX modelling approach is a parameter estimation methodology for identifying both the important model terms and the parameters of unknown non-linear dynamic systems. For multiple input, single output noiseless systems this model takes the form given in equation 1:

$$\begin{aligned} y(n) = \quad & f[\vec{u}(n-N_u)^l, y(n-N_y)], \\ \forall \quad & N_u = 0 \ldots N_u^{max}, \\ l \quad & = 1 \ldots l_{max}, N_y = 1 \ldots N_y^{max}. \end{aligned} \tag{1}$$

were $y(n)$ and $\vec{u}$(n) are the sampled output and input signals at time n respectively, N_y and N_u are the regression orders of the output and input respectively. The input vector $\vec{u}$ is d-dimensional, the output y is a scalar. $f()$ is a non-linear function and it is typically taken to be a polynomial or wavelet multi-resolution expansion of the arguments. The degree l_{max} of the polynomial is the highest sum of powers in any of its terms.

The NARMAX methodology breaks the modelling problem into the following steps: i) Structure detection (i.e. determining the form of the non-linear polynomial), ii) parameter estimation (i.e. obtaining the model coefficients), iii) model validation, iv) prediction, and v) analysis. A detailed description of how these steps are done is presented in (Billings and Chen, 1998; Korenberg et al., 1988; Billings and Voon, 1986).

The calculation of the NARMAX model parameters is an iterative process. Each iteration involves three steps: i) estimation of model parameters, ii) model validation and iii) removal of non-contributing terms.

Using System Identification to Obtain Robot-Executable Narmax models It is difficult for a programmer to teach a particular task to a robot as humans and robots perceive and act in the world differently; humans and robots have different sensor and actuator modalities (Alissandrakis et al., 2005). We have adopted a similar approach to (Nehmzow et al., 2007) where the mobile robot's trajectory of the desired behaviour is used as a suitable communication channel between the human and the robot.

In this paper we present an approach to show how route learning can be translated directly into control code using system identification. The trajectory of the human is used as reference and this is translated algorithmically and automatically into robot control code. An outline this method is illustrated in Figure 2.

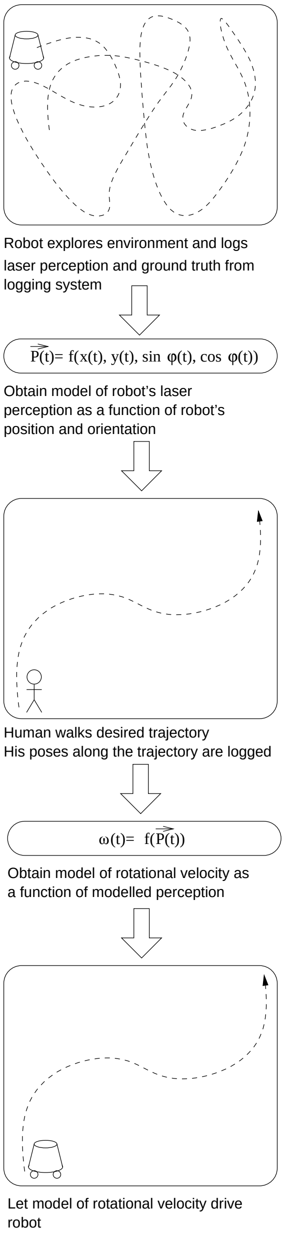
Robot explores environment and logs laser perception and ground truth from logging system
$\vec{P}(t)$= f(x(t), y(t), sin φ(t), cos φ(t))
Obtain model of robot's laser perception as a function of robot's position and orientation
Human walks desired trajectory
His poses along the trajectory are logged
ω(t)= f($\vec{P}(t)$)
Obtain model of rotational velocity as a function of modelled perception
Let model of rotational velocity drive robot

Figure 2: System Identification Process used to obtain Robot-Executable Narmax models

1. *Obtaining the sensorgraph*: The robot explores the environment and obtains a detailed log of sensor perceptions throughout the working environment. This detailed log, *the sensorgraph*, contains information such as laser readings and the robot's pose $< x, y, \varphi >$, i.e. its position $< x, y >$ and heading φ within the environ-

ment.

2. Obtain environment models: The system identification method is used to obtain a number of polynomial models that model the robots laser perception as a function of the robot's pose within the environment.

These models allow us to estimate the robot's laser perceptions along new novel trajectories that have been demonstrated by a human in the next stage.

3. Obtaining a human-demonstrated trajectory: The human user demonstrates the desired trajectory by performing the task in the target environment. During this demonstration period the demonstrator's pose is continuously observed and logged by a tracking system. These poses are then used to compute the translational and rotational velocities of the human by using consecutive poses along the trajectory.

4. Obtain the final, environment-model-based, human demonstrated controller: The final controller is obtained by using the system identification technique to obtain a sensor-based controller. The human demonstrator's location is used with the environment models (from stage 2) to obtain the robot's laser perception at that position. The modelled laser perceptions are then used as inputs to the system identification process with the computed velocities of the human demonstrator (from stage 3) as outputs.

5. Robot copies behaviour: The controller (from stage 4) can then be used to drive the robot along the human-demonstrated trajectory within the target environment, copying the behaviour of the human.

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 *Vicon* motion tracking system that delivers highly accurate position data (x, y, z) for targets using reflective markers and high speed high resolution cameras. In the experiments presented here we use the Metralabs *SCITOS G5* autonomous robot *Swilly*, shown in Figure 3.

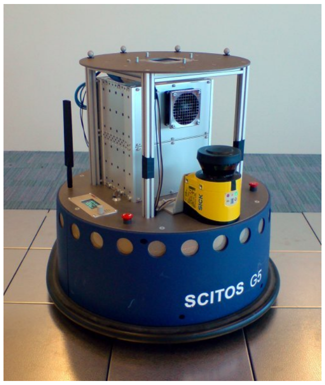

Figure 3: *Swilly*, the Metralabs SCITOS G5 mobile robot used in the experiments

The robot is equipped with 24 sonar sensors distributed around the 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 $([0°, 180°])$.

Experimental Setup The robotics arena is configured with artificial walls to consist of a working environment measuring $4m \times 3m$, as illustrated in Figure 4.

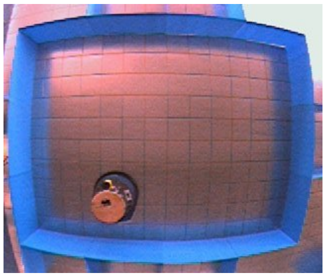

Figure 4: Overhead image of the Robot Arena Setup with the robot visible in the lower left of the test area

The robot explores the test area using a random walk obstacle avoidance behaviour whilst simultaneously logging its laser perceptions and the robot's actual x, y, x positions, obtained from the *Vicon* motion tracking system.

We ensure that adequate data to model the environment has been logged by computing histograms for the robot's actual position using the *Vicon* tracking system along the x-axis and y-axis. It is equally important to consider the robot's heading whilst exploring the environment as the modelling process needs to consider all possible orientations of the robot. Thus, we also construct a histogram of the robot's headings where the robot has logged sensor data. By obtaining an almost uniform distribution with the histograms we can ensure that adequate data has been logged.

The laser data is then median-filtered over 30° segments. So rather than 360 laser readings we have six median filtered segments that are used as input to the first modelling process.

Using the Narmax system identification method we obtain a number of polynomial models that model the robots laser perception as a function of the robot's $< x, y >$ position and heading φ (here we use $sin(\varphi)$ and $cos(\varphi)$) to form the function

$$\vec{P}(t) = f(x(t), y(t), \sin\varphi(t), \cos\varphi(t)) \qquad (2)$$

.

The Narmax method used in this work has multiple inputs and a single output. Thus, we require at least one model per laser segment (6 in this case). 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 laser models to only 4 models, modelling 90° heading segments, thus covering the entire 360° range of possible robot headings with 4 models. Put differently, we constructed four models, one for each 30° laser segment, of the form $P_k = f < x,y > \forall\varphi = k$, where P_k is a model of the predicted laser reading when the robot assumes a heading φ of k degrees.

The accuracy of the obtained environment models have been assessed by driving the robot along a novel trajectory within the test environment (see Figure 5) and logging the real laser perceptions along with the robot's position.

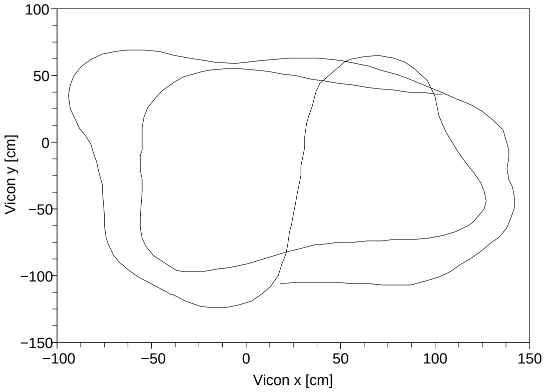

Figure 5: Novel trajectory driven by the robot in the test area during which real laser data was logged and compared with model predicted laser data

The robot's position was used as input into the obtained environment models and the logged laser reading and modelled laser reading compared. In Figure 6 we have plotted the real and modelled laser values for laser segment 1, and plotted the absolute error between the values. The standard error for all the lasers is shown in Table 1.

Table 1: Absolute Mean error and Standard error in centimetres for all lasers over validation trajectory (Figure 5)

Laser	Mean and Standard error [cm]
$\tilde{L}_1$	14.7 ± 1.2
$\tilde{L}_2$	16.9 ± 1.0
$\tilde{L}_3$	16.9 ± 1.1
$\tilde{L}_4$	19.9 ± 1.6
$\tilde{L}_5$	16.9 ± 1.5
$\tilde{L}_6$	14.7 ± 1.1

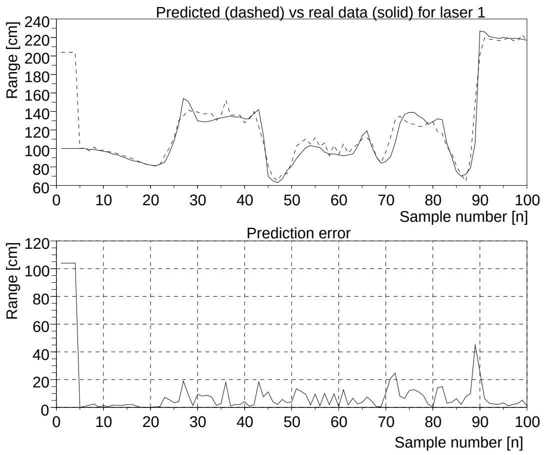

Figure 6: Logged trace from laser 1 compared with modelled trace from laser 1 whilst being driven along the trajectory in Figure 5 and absolute error between logged and modelled lasers.

Having obtained environment models with satisfactory accuracy we conduct experiments where a human follows two different trajectories, S-shaped and U-shaped.

Experiment 1 - S-shaped trajectory

The human demonstrated to the robot how to move within the test area in a S-shaped trajectory. The demonstrator started in the lower left side of the environment and walked in a S-shape finishing in the upper right side. The demonstrator's x and y position was again obtained from the *Vicon* system and logged every 250ms as the human demonstrator moved along the desired path 5-times in total. The logged trajectories are shown in Figure 7.

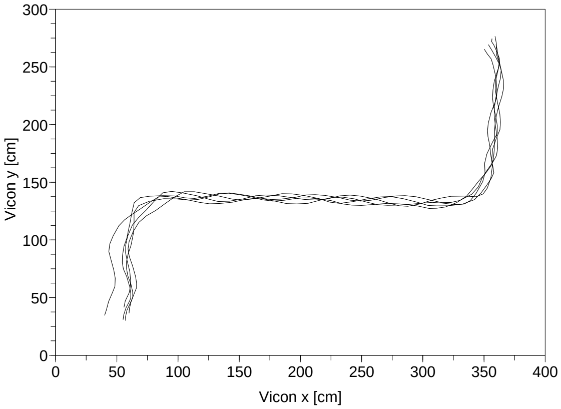

Figure 7: Five trajectories of the desired S-shape behaviour demonstrated by the human in the test environment

The human demonstrator's $< x,y >$ positions are then used to compute the translational and rotational velocities of the human along the trajectory by using consecutive $< x, y >$ samples. Next, we obtain a series of expected robot perceptions along this trajectory of logged $< x, y >$ positions using the environment models.

The final controller is obtained by using the Narmax system identification technique to obtain an environment-model sensor-based controller. The computed human demonstrator's rotational velocity is used with the series of expected robot perceptions forming the function

$$\omega(t) = f(\vec{P}(t)). \tag{3}$$

The modelled laser perceptions are used as inputs to the Narmax system with the computed rotational velocities of the human demonstrator as outputs. We used parameters with an input regression of 10 and polynomial degree 2 to obtain the following final Narmax model that had 20 terms, shown in equation 4.

$$\begin{aligned}
\omega(t) \quad = \quad & -0.311 \\
& +0.001267 * u(n,1) \\
& -0.007369 * u(n,2) \\
& -0.001245 * u(n,3) \\
& +0.00374 * u(n,4) \\
& +0.00787 * u(n,5) \\
& +0.00384 * u(n,6) \\
& -0.0000078 * u(n,1)^2 \\
& -0.000014 * u(n,2)^2 \\
& -0.000011 * u(n,3)^2 \\
& -0.0000053 * u(n,4)^2 \\
& -0.0000034 * u(n,6)^2 \\
& +0.000038 * u(n,1) * u(n,2) \\
& -0.000034 * u(n,1) * u(n,5) \\
& -0.0000059 * u(n,1) * u(n,6) \\
& -0.00000399 * u(n,2) * u(n,5) \\
& +0.0000163 * u(n,3) * u(n,4) \\
& +0.0000129 * u(n,3) * u(n,6) \\
& -0.00000166 * u(n,4) * u(n,5) \\
& -0.0000178 * u(n,5) * u(n,6)
\end{aligned} \tag{4}$$

Results In the final stage of the experiment the obtained final controller is used to drive the robot whilst the robot's actual positions during this stage of the experiment are logged as illustrated in Figure 8.

A visual inspection of Figure 8 shows that, although not perfect, the obtained controller produces a behaviour that resembles the humans demonstrator's initial trajectory.

Experiment 2 - *U*-shaped trajectory

The human demonstrated to the robot how to move within the test area in a *U*-shaped trajectory. The demonstrator started in the lower left side of the environment and walked in a *U*-shape finishing in the upper left side. The demonstrator's x and y position was again obtained from the *Vicon* system and logged every 250ms as the human demonstrator moved along the desired path 5-times in total. The logged trajectories are shown in Figure 9.

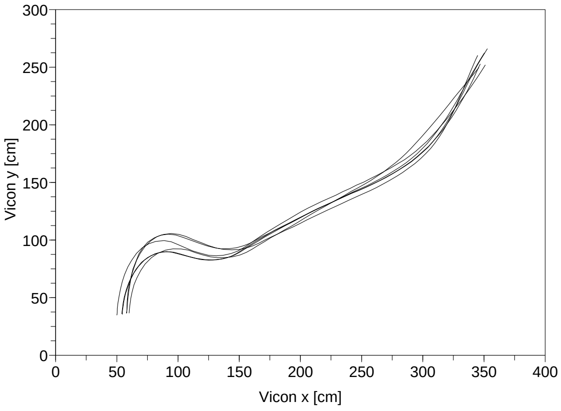

Figure 8: Six trajectories of the robot under control of the *S*-shape sensor based controller in the test environment (compare with figure 7).

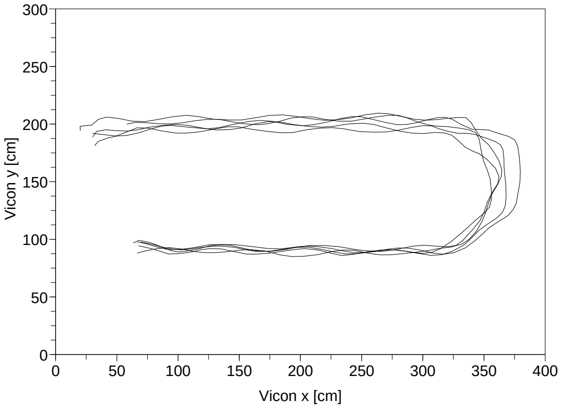

Figure 9: Five trajectories of the desired *U*-shape behaviour demonstrated by the human in the test environment.

The human demonstrator's $< x, y >$ positions are then used to compute the translational and rotational velocities of the human along the trajectory by using consecutive $< x, y >$ samples. Next, we obtain a series of expected robot perceptions along this trajectory of logged $< x, y >$ positions using the environment models.

The final controller is obtained by using the Narmax system identification technique to obtain an environment-model sensor-based controller. The computed human demonstrator's rotational velocity is used with the series of expected robot perceptions. The modelled laser perceptions are used as inputs to the Narmax system with the computed rotational velocities of the human demonstrator as outputs. We used parameters with an input regression N_u of 10 and polynomial degree l of 2 to obtain the final Narmax model that had 96 terms.

Results In the final stage of the experiment the obtained final controller is used to drive the robot whilst the robot's actual positions during this stage of the experiment are logged as illustrated in Figure 10.

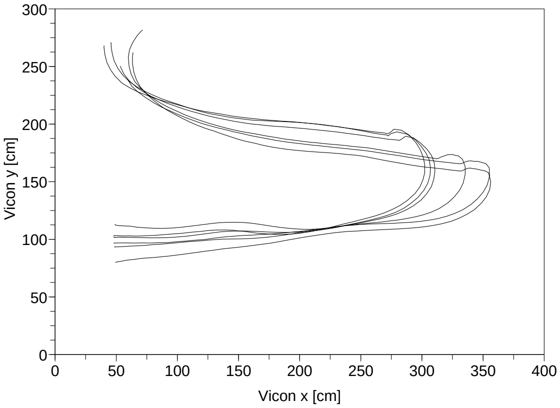

Figure 10: SIX TRAJECTORIES OF THE ROBOT UNDER CONTROL OF THE U-SHAPE SENSOR BASED CONTROLLER IN THE TEST ENVIRONMENT.

A visual inspection of Figure 10 shows that, although not perfect, the obtained controller produces a behaviour that resembles the human demonstrator's initial trajectory.

Summary and Conclusion

Summary

In this paper we present experiments that show that an agent's behaviour — in this case a human demonstrator — can be translated into directly-executable robot control code, meaning that no programming is at all necessary. The method of "translating" behaviour into code is shown in Figure 2.

Conclusions

There are a number of decisive advantages to the presented method of automatically generating robot-controlling code in this way:

1. Code generation is very quick,
2. The generated code is canonical, which allows the development of analysis tools such as sensitivity analysis methods,
3. The generated code is parsimonious, which is relevant when the code is to be used on robots with little on-board computing resources.

Future Work Our next experiments at the Intelligent Systems Research Centre here in Londonderry will address the following weaknesses of our current implementation:

1. The external-camera-based tracking system we used here is very precise, but also very expensive. In future we plan to use a camera-based tracking system, where the camera used is the one mounted on *Swilly.*
2. Developing models of higher accuracy, and
3. Developing smaller models.

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

References

Akanyeti, O., Nehmzow, U., Weinrich, C., Kyriacou, T., and Billings, S. (2007). Programming mobile robots by demonstration through system identification. In *European Conference on Mobile Robotics (ECMR)*, pages 162–167.

Alissandrakis, A., Nehaniv, C., Dautenhahn, K., and Saunders, J. (2005). An approach for programming robots by demonstration: Generalization across different initial configurations of manipulated objects. In *Proceedings of the IEEE international symposium on computational intelligence in robotics and automation*, page 61.

Billings, S. and Chen, S. (1998). The determination of multivariable nonlinear models for dynamical systems. In Leonides, C., (Ed.), *Neural Network Systems, Techniques and Applications*, pages 231–278. Academic press.

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

Coates, A., Abbeel, P., and Ng, A. (2008). Learning for control from multiple demonstrations. In *Proceedings of the 25th international conference on Machine learning*, pages 144–151. ACM.

Demiris, Y. (2009). Knowing when to assist: developmental issues in lifelong assistive robotics. In *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2009)*, pages 3357–3360, Minneapolis, Minnesota.

Demiris, Y. and Dearden, A. (2005). From motor babbling to hierarchical learning by imitation: a robot developmental pathway. In Berthouze, L., Kaplan, F., Kozima, H., Yano, H., Konczak, J., Metta, G., Nadel, J., Sandini, G., Stojanov, G., and Balkenius, C., (Eds.), *Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, pages 87–93.

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

Nehmzow, U., Akanyeti, O., Weinrich, C., Kyriacou, T., and Billings, S. (2007). Robot programming by demonstration through system identification. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007. IROS 2007*, pages 801–806.