6.4 Robot Actuator Modelling

The robot application described in this section is based on data sampled from a physical system, supplied by Tom Kavli, SINTEF, Oslo. The application and the data sets have been described for modelling work in (Kavli, 1992) and in (Johansen and Foss, 1994b). A brief overview is given here:

Many industrial robot applications now demand high dynamic accuracy. Model based control schemes have the potential to improve performance, but the use of hydraulic manipulators has suffered due to the lack of good nonlinear models for the hydraulic components. Model based control schemes have been more successful on electric direct drive arms with low friction and linear actuators. The goal of the learning task is to form a model of the servo valve/actuator system of a hydraulic robot. The robot is an ABB Trallfa TR4000 Robot, specially designed for spray painting, where tracking accuracy over a desired trajectory is extremely important. The control signal u is described as a function of the joint position q, velocity $\dot{q}$ and acceleration $\ddot{q}$:

$$u = f\left(q, \dot{q}, \ddot{q}\right). \tag{6.7}$$

The nonlinearities are due to:

- the changing moment arm of the cylinder over the operating range,
- the nonlinear damping coefficient due to the quadratic flow/pressure relation for turbulent flow and,
- the changing pressure gain characteristics for the servo valve at different flow rates.

The data was sampled by logging the data at 100Hz, while the manipulator moved along a randomly generated path. The velocity and acceleration signals were calculated by low pass filtering the data and differentiating the joint positions. The linear effects in the system were subtracted from the data to emphasise the nonlinearity of the system. The training data consisted of 8000 training points and the test set had 1000 points.

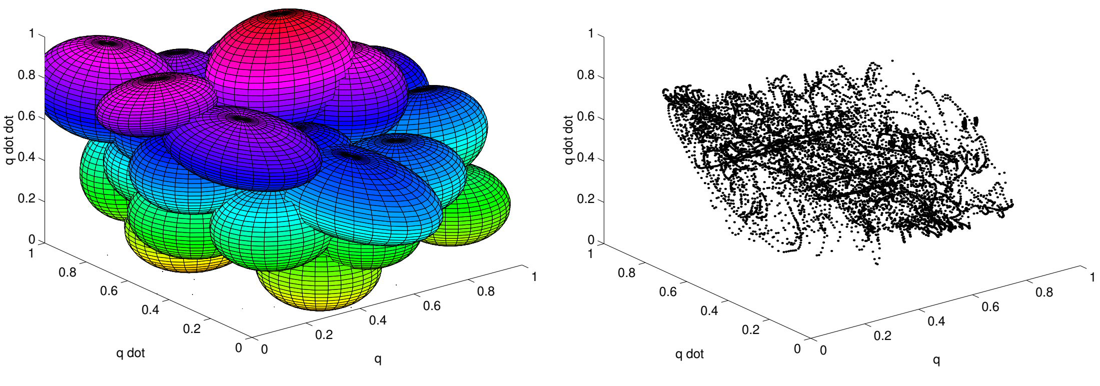
6.4.1 Experimental results

The ASMOD results * are taken from (Kavli, 1992). LSA (Local Search Algorithm) results + are from (Johansen and Foss, 1994b). The LMN results were obtained by using the algorithms described in this thesis, as well as the active selection algorithm to reduce the number of points in the training set (the training set was reduced to a maximum of 1200 points).

The MARS results are shown in Figure 6.30(a). The cross-validation results for the local model net with local learning are shown in Figure 6.29(a) is compared with the global learning case in Figure 6.29(b). Note the poor performance of local learning in the cross-validation results.

Model Type	Test Error (NRMS)
LSA+	17
MLP (3-20-1)	23
ASMOD*(Quadrat.)	15
ASMOD*(Linear)	17
RBF*	23
MARS	17
LMN (local learning)	18
LMN (global learning)	17
LHM (axis oblique)	18
LMN (local learning)	18
LMN (global learning)	17

Table 6.2: Robot modelling results on the test set. The only method to provide better results than the LMN was the ASMOD model with quadratic splines (and a large number of parameters at 512 basis functions). Local Modelling proved to lead to slightly worse results, and required a larger number of parameters, but produced 'safer' extrapolation to areas with little training data.



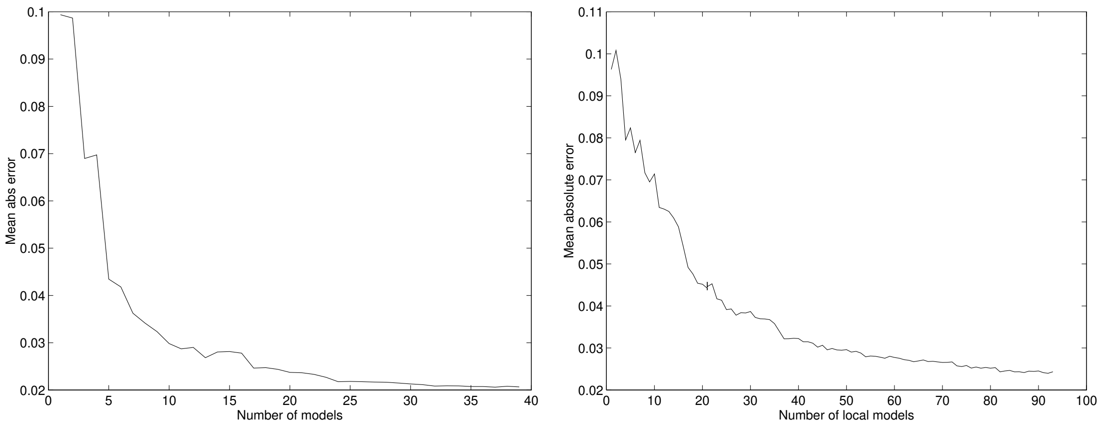
(a) LMN Basis Functions for the robot model

(b) Robot training data

Figure 6.27: Distribution of robot training data and local models' basis functions.

Only one of the runs produces a mean error similar to that of the global optimisation. This is, however, due to a weakness in the structure identification algorithm, which stops growing when the error stops decreasing. In global learning, the greater degree of freedom for the optimisation means that the additional local models have a more pronounced effect, and construction continues. By altering n_{res} to 5 for local learning produced the results shown in Figure 6.28(b), where a larger number of local models were built, but which achieved an improvement in accuracy to that close to the globally trained model.

The problems with ill-conditioning for global learning are less prevalent in this application, due to the low dimensionality, accompanied by large amounts of training data, and the apparent smoothness of the underlying process nonlinearity.



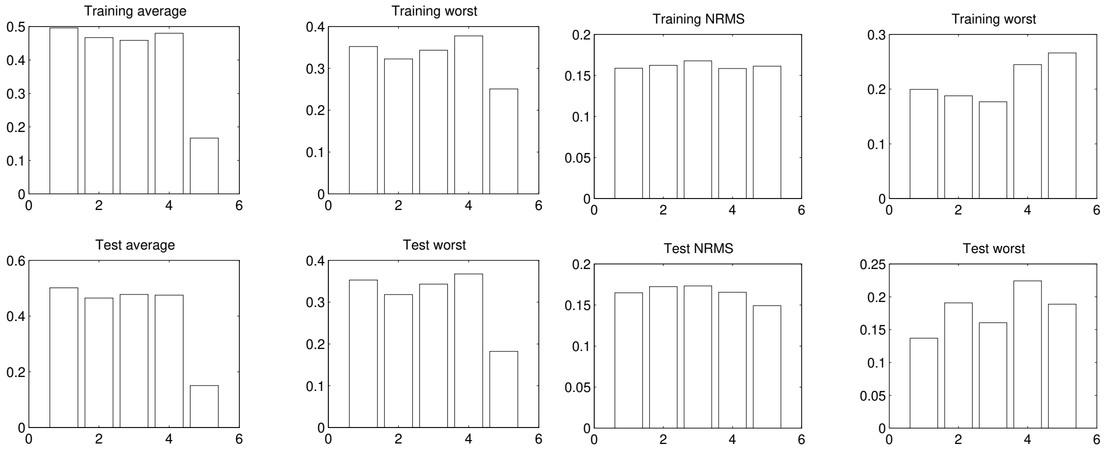
(a) LMN error curve with global learning

(b) LMN error curve with local learning

Figure 6.28: The progress of the average absolute error as new models are added to the robot modelling example. Note that local learning finishes earlier than global learning.

Visualisation of the system

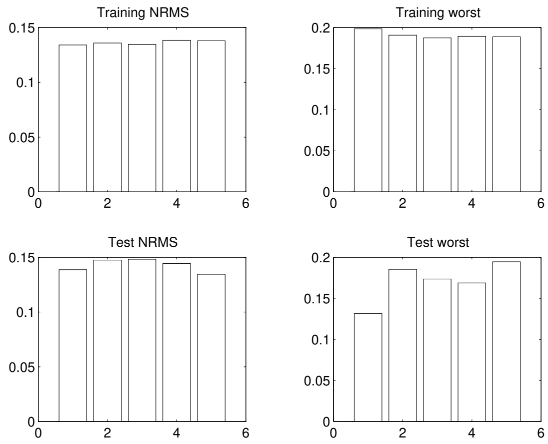
The following surfaces are representations of the input space as seen through different slices through the space. The corners of the plots are areas where the system had no training data, and are therefore unreliable.



(a) LMN results with local learning

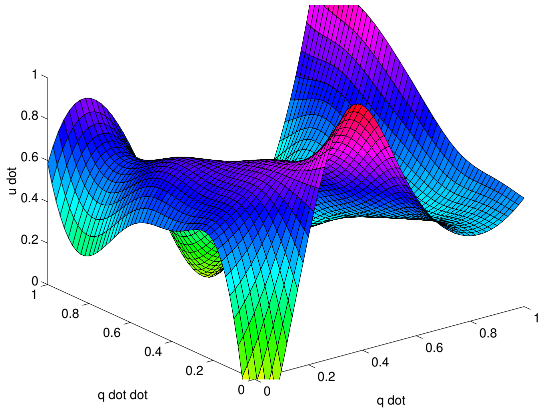
(b) LMN results with global learning

Figure 6.29: Cross-validation results for the Trallfa robot with Local Model Nets. Note the poor performance of local learning in all but one example. By increasing n_{res} to 5, the local learning could perform better, as shown in Figure 6.28(b)

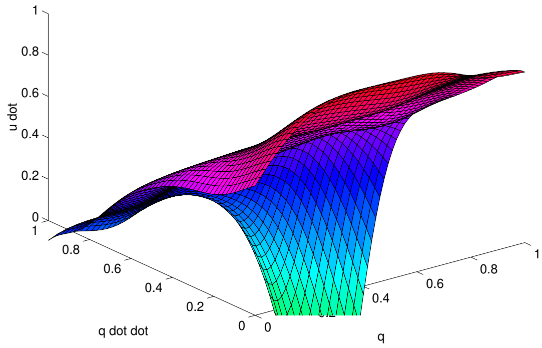


(a) MARS results with maximum number of basis functions set to 200

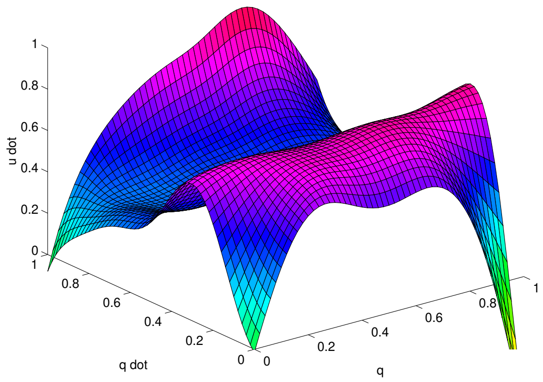
Figure 6.30: Modelling results for the Trallfa robot with MARS.



(a) Speed and acceleration, with position at 0.5.

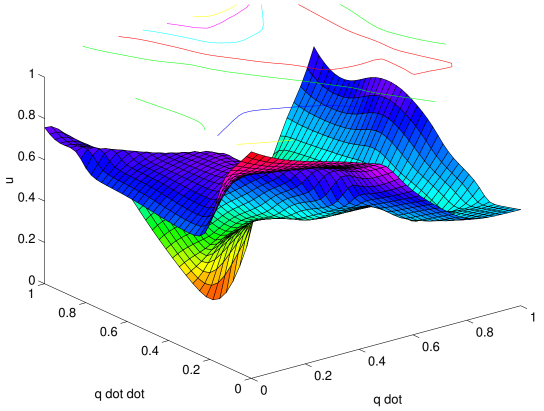


(b) Position and acceleration with speed at 0.5.

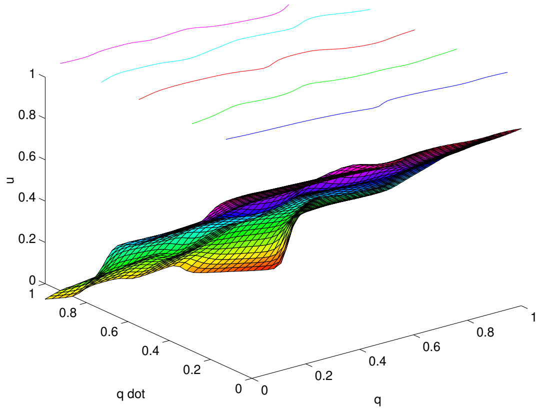


(c) Position and speed with acceleration at 0.5.

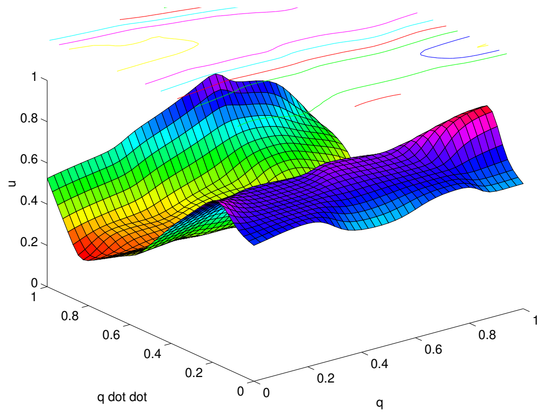
Figure 6.31: Output responses for slices through the robot actuator model. Note the way the model response surface flies off in the corners where data does not exist. This is one disadvantage of normalised basis functions, where the basis functions are supported through the whole input space, combined with global learning, which produces less robust models. Compare the plots here to those achieved with local learning in Figure 6.32.



(a) Speed and acceleration, with position at 0.5.



(b) Position and acceleration with speed at 0.5.



(c) Position and speed with acceleration at 0.5.

Figure 6.32: Output responses for slices through the robot actuator model using local learning. Compare the plots here to those achieved with local learning in Figure 6.31. The locally trained models provide a less dramatic form of extrapolation outside the populated area of the input space.

6.5 Conclusions

6.5.1 Rolling mill results

The rolling mill application provided an interesting environment for testing the local model network methodology and the constructive algorithms proposed in this thesis. To solve the problem meant dealing with the practical problems of data acquisition, pre-processing, and variable selection. In some ways though it is also not an ideal choice to show off the structure, due to the poor available understanding of the system, and the difficulty in performing experiments to acquire data, or to validate the usefulness of models produced. Despite the relative deficit of formalised *a priori* knowledge about the system, the local model network methods proved to be the most accurate, and most interpretable representations of the rolling mill process. The MARS algorithm was relatively fast, but did not have the desired accuracy, and the conventional MLP neural network was very slow during training, and produced low-accuracy models.

The validation phase used cross-validation techniques to test the robustness of the learning algorithms, as well as using validation sets from completely different runs. This showed up some interesting features in the experimental setup. The local learning techniques led to significantly more accurate models for new validation data than globally trained ones, even though they were found to be slightly less accurate on 5-fold cross-validation runs based on partitions of the training set. This seems to indicate that the local learning produced a more robust model of the mill, at the expense of accuracy on the training data.

The state-dependent error estimates proved, despite their simplicity, that they could capture the areas of the input space where the model is least accurate – the acceleration and deceleration phases. Other tools were used which helped interpret the model, including the visualisation of model residuals, accompanied by the position of the data in the operating space of the local model net's basis functions. This can also be combined with slices through the basis functions, in the form of either contour plots or ellipsoidal plots of basis functions in three dimensions.

6.5.2 Robot actuator results

The results obtained by the Local Model Net are comparable with the results quoted in the original work (Kavli, 1992) and also with those in (Johansen and Foss, 1994b). The problem is low dimensional, the nonlinearity seems to be fairly smooth, and there is an abundance of training data. This means that other model structures can also cope with the problem, and that while the local model nets perform well at the modelling task, the potential advantages of local model nets are not as evident as in other applications. A further interesting result from the experiment pointed out weaknesses in the structure identification algorithm working with local learning, where it often stops construction too early. Using a larger n_{res} lead to better results, but also needed a larger number of parameters.