2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

# Multi-Step-Ahead Optimal Learning Strategy for Local Model Networks with Higher Degree Polynomials

Oliver Bänfer, Geritt Kampmann and Oliver Nelles

***Abstract*— The idea of a learning strategy extension for nonlinear system identification with local polynomial model networks is presented in this paper. Usually the polynomial model tree (POLYMOT) algorithm utilizes a one-step-ahead optimal learning strategy. A demonstration example shows that this greedy behavior is not the best choice to reach a satisfying global model. Thus this strategy should be enlarged to a multi-step-ahead optimal learning. Therefore, it is possible to find the optimal global model in a special case.**

## I. Introduction

The output $\hat{y}$ of a local model network, also known as local neuro-fuzzy network, with $p$ inputs $\underline{u} = [u_1\ u_2\ \cdots\ u_p]^T$ can be calculated as the interpolation of $M$ local model outputs $\hat{y}_i$, $i = 1, \ldots, M$

$$\hat{y} = \sum_{i=1}^{M} \hat{y}_i(\underline{u})\Phi_i(\underline{u}), \tag{1}$$

where the $\Phi_i(\cdot)$ are called interpolation or validity or weighting functions [4]. These validity functions describe the regions where the local models are valid; they describe the contribution of each local model to the output. Because a smooth transition (no switching) between the local models is desired here, the validity functions are smooth functions between 0 and 1. For a reasonable interpretation of local model networks it is furthermore necessary that the validity functions form a *partition of unity*:

$$\sum_{i=1}^{M} \Phi_i(\underline{u}) = 1\,. \tag{2}$$

Thus, everywhere in the input space the contributions of all local models sum up to 100%.

In principle, the local models can be chosen of arbitrary type. If their parameters are to be estimated from data, however, it is extremely beneficial to choose a linearly parameterized model class. The most common choice are polynomials. Polynomials of degree 0 (constants) yield a neuro-fuzzy system with singletons or a normalized radial basis function (NRBF) network. Polynomials of degree 1 (linear) generate local linear model structures, which is by far the most popular choice. As the degree of the polynomials increases, the number of local models required for a certain accuracy decreases. Thus, by increasing the local models' complexity, at some point a polynomial of high degree with just one local model ($M = 1$) is obtained, which is in fact equivalent with a global polynomial model ($\Phi_1(\cdot) = 1$):

$$\begin{aligned}\hat{y}_i(\underline{u}) =& w_{i,0} + w_{i,1}u_1 + w_{i,2}u_1^2 + w_{i,3}u_2 + \ldots \\ & w_{i,4}u_1u_2 + w_{i,5}u_2^2 + \ldots + w_{i,nx}u_p^l.\end{aligned} \tag{3}$$

The number of parameters/inputs of the consequents $nx$ of this local model is

$$nx = \frac{(p+l)!}{p!l!}, \tag{4}$$

where $p$ is the number of inputs and $l$ is the polynomial degree.

For many applications the complexity steps to local models with higher degree polynomials [2], [5] do pay off. The higher degree dependency of the number of local polynomial parameters on input space dimensionality $p$ is compensated by the smaller number of required neurons.

This article is organized as follows. Section II gives an overview on the basics of the POLYMOT algorithm. The disadvantage of a one-step-ahead optimal learning strategy is demonstrated with an example in Sect. III. The idea and the motivation for a multi-step-ahead optimal strategy for learning local model networks are proposed in Sect. IV. This paper ends by summarizing the important conclusion.

## II. Polynomial Model Tree Algorithm

The partitioning strategy of the polynomial model tree (POLYMOT) algorithm [1] is strongly motivated by the local linear model tree (LOLIMOT) algorithm and depends on an incremental tree-construction that divides the input space by orthogonal splits. In each iteration a new local model is added or the number of parameters of the local worst one is increased. Thus, POLYMOT belongs to the class of incremental or growing algorithms and can be seen as a one-step-ahead optimal strategy, i.e., only the improvement from one iteration to the next iteration is considered. In each iteration of the algorithm the validity functions which correspond to the actual partitioning of the input space are computed and the corresponding rule consequents are optimized by a local weighted subset selection technique.

## III. Demonstration Example

The disadvantage of an one-step-ahead optimal learning strategy for the POLYMOT algorithm becomes apparent with a polynomial approximation problem. Therefore, in the following the below function is modeled with a local model

O. Bänfer, G. Kampmann and O. Nelles are with the Department of Mechanical Engineering, University of Siegen, D-57068 Siegen, Germany oliver.baenfer@uni-siegen.de

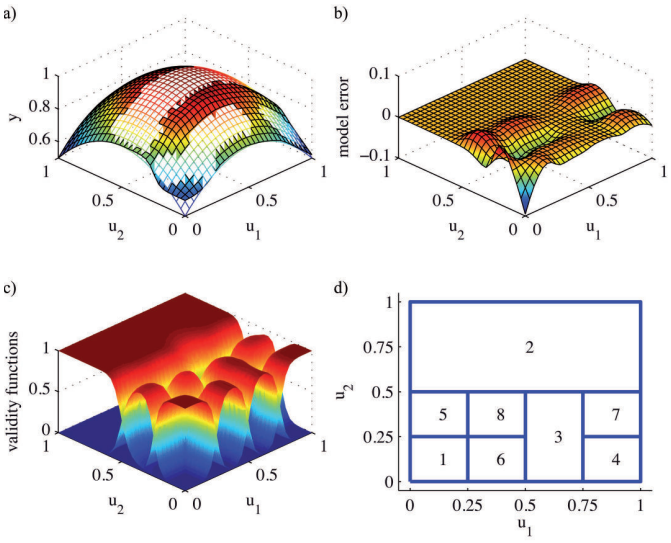


Fig. 1. Approximation with a local model network and a penalty factor $C = 0.85$. Figure a) shows the function to be approximated (light) and their approximation (solid). In Fig. b) the absolute model error $e = y - \hat{y}$ is illustrated. Figure c) shows the validity functions which demonstrate the partitioning. In Fig. d) the partitioning in 8 local models is presented. This figure demonstrates also the sequence of generating new local models beginning with local model number 1 up to local model number 8.

network, see Fig. 1:

$$y = 1 - (u_1 - 0.5)^2 - (u_2 - 0.5)^2. \tag{5}$$

For the approximation 900 equally distributed, noise-free data samples are generated. This function shall be approximated with a normalized root mean squared error (NRMSE) of less than 10%. The algorithm can generate local models at most of 3rd polynomial degree. In each iteration 2 new nominal parameters $n_{\text{nom}}$ (1 for every input dimension) were added for the stepwise regression. The parameter for the local model offset is always estimated. The setting of the POLYMOT algorithm and the results for this approximation problem are summarized in table I and Fig. 2.

TABLE I
RESULTS OF THE DEMONSTRATION EXAMPLE USING DIFFERENT PENALTY FACTORS

| Penalty factor | Local models | Parameters overall | NRMSE |
|---|---|---|---|
| 0.85 | 8 | 35 | 0.084 |
| 0.90 | 1 | 7 | 0.000 |

## IV. Multi-Step-Ahead Optimal Learning Strategy

The polynomial model tree algorithm utilizes a one-step-ahead optimal learning strategy. This means that the algorithm decides between splitting or using a higher degree polynomial which tends in each iteration to the lowest global model error. Hence, the POLYMOT algorithm belongs also to the class of *greedy algorithms*. Genetic programming offers an alternative since it performs a global search for the optimal partition tree and is therefore able to backtrack

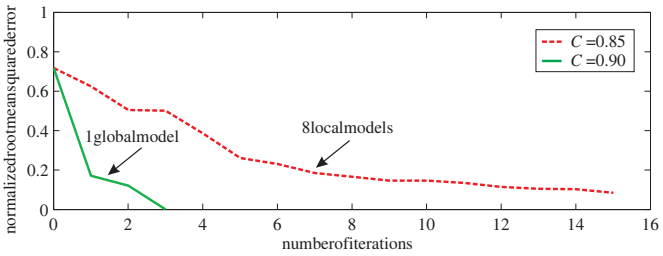


Fig. 2. Convergence behavior (global model error) for a 2-dimensional approximation problem with penalty factor $C = 0.85$ (8 local models) and penalty factor $C = 0.90$ (1 global model).

in case of sub-optimal intermediate split decisions [3]. The demonstration example shows that a splitting is not a good choice to model a polynomial function. After the first split one half of the input space is modeled perfectly with only one local model (number 2), see Fig. 1. Since the other half of the input space is splitted perpetually, the combination of many local models reduces the global model error only slowly and a lot of local model parameters are necessary. At this time the only way to achieve different model architectures is a modification of the penalty factor, but the algorithm is still a greedy algorithm. On that account the POLYMOT algorithm should be enlarged which a multi-step-ahead optimal learning strategy. This means that the algorithm does not only check all possible splits and higher degree polynomials for one-step ($k = 1$), but also for k-steps.

## V. Conclusion

The idea and the motivation of an extension from an one-step-ahead to a multi-step-ahead optimal learning strategy for the POLYMOT algorithm has been presented. The computational demand will grow approximately exponentially (*curse of dimensionality*), because the global model error of all combinations of splits and higher degree polynomials for k-steps must be calculated. Since the training of medium sized problems with an one-step-ahead strategy on modern computers still is a matter of seconds or a couple of minutes, the multi-step-ahead strategy for POLYMOT will be a feasible solution to reduce the global model error. Furthermore it will be possible to find the optimal global model, if the number of steps of the multi-step-ahead optimal strategy is equal to the number of iterations during the learning phase.

## References

[1] O. Bänfer and O. Nelles. Polynomial Model Tree (POLYMOT) - A New Training Algorithm for Local Model Networks with Higher Degree Polynomials. In *IEEE International Conference on Control and Automation (ICCA)*, pages 1571–1576, Christchurch, New Zealand, December 2009.

[2] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman & Hall, London, UK, 1996.

[3] F. Hoffmann and O. Nelles. Genetic programming for model selection of TSK-fuzzy systems. *Information Sciences*, 136:7–28, 2001.

[4] O. Nelles. *Nonlinear System Identification*. Springer, Berlin, Germany, 2001.

[5] H. Sequenz, A. Schreiber, and R. Isermann. Identification of nonlinear static processes with local polynomial regression and subset selection. In *IFAC Symposium on System Identification (SYSID)*, pages 138–143, Saint-Malo, France, July 2009.