Modelling of Parameterized Processes via Regression in the Model Space

Witali Aswolinskiy, René Felix Reinhart and Jochen Jakob Steil *

Research Institute for Cognition and Robotics - CoR-Lab
Universitätsstraße 25, 33615 Bielefeld, Germany

Abstract. We consider the modelling of parameterized processes, where the goal is to model the process for new parameter value combinations. We compare the classical regression approach to a modular approach based on regression in the model space: First, for each process parametrization a model is learned. Second, a mapping from process parameters to model parameters is learned. We evaluate both approaches on a real and a synthetic dataset and show the advantages of the regression in the model space.

1 Introduction

Many processes in nature and technology depend on the environment or the context. For example, a chemical process may dependent on the temperature and pressure and a mechanical process may depend on the applied force and material properties. Models of such processes are important in many applications, e.g. for optimization of production processes.

We distinguish between process parameters, which do not change during the process, e.g. temperature and pressure in the before mentioned examples, and the process inputs, e.g. time. The constancy of the process parameters during the process separates our definition from the contextual features defined in [4], where contextual, primary and irrelevant features were differentiated.

The classical, data-driven modelling approach for parameterized processes is to train a regressor using the combination of process parameters and process inputs. This leads to an increased training data demand to cover the high-dimensional input space spanned by the combination of process parameters and process inputs as independent variables for the regression. In contrast to this monolithic approach, we propose a novel modular approach, where we separate the learning of process parameters and process inputs utilizing learning in the model space [5]. First, for each process parameter combination, we learn a model for the process given the process input, which yields learned model parameters. Second, we learn a mapping from process parameters to model parameters - a map from the process parameter space to the space of process models. By decoupling the learning of process parameters from the process inputs, the dimensionality of the input data is smaller and thus better generalization from fewer samples can be achieved.

*This work is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition "it's OWL" (intelligent technical systems OstWestfalenLippe). The authors thank Hesse GmbH [1] and Infineon Technologies AG [2] for permission to use the data from the project 'Intelligent Copper Bonding' (InCuB) [3].

Learning in the model space was previously mainly applied to time series classification [5, 6]. Our contribution is the extension of the approach to regression in the model space of process models. A similar approach was used in robotics for learning of parameterizable skills based on dynamic motion primitives [7, 8]. Another related approach are context-dependent neural nets [9].

2 Model Space Regression (MSR)

For regression we use Extreme Learning Machines (ELM, [10]). ELMs are feed-forward neural networks with three layers: Input layer $\boldsymbol{x} \in \mathbb{R}^I$, a layer $\boldsymbol{h} \in \mathbb{R}^N$ with N hidden neurons, and output layer $\boldsymbol{y} \in \mathbb{R}^O$. The output is computed by $\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{W}^{out}\boldsymbol{h}(\boldsymbol{W}^{in}\boldsymbol{x} + \boldsymbol{b})$, where $\boldsymbol{W}^{in} \in \mathbb{R}^{N \times I}$ is the random input weight matrix, $h(a) = (1 + e^{-a})^{-1}$ the logistic function applied elementwise to the N neuron inputs, $\boldsymbol{b} \in \mathbb{R}^N$ the neuron biases, and $\boldsymbol{W}^{out} \in \mathbb{R}^{O \times N}$ the readout weight matrix. The readout weights are learned with ridge regression: $\boldsymbol{W}^{out} = \arg\min_{\boldsymbol{W}}(\|\boldsymbol{H}(\boldsymbol{X})\boldsymbol{W}^T - \boldsymbol{T}\|^2 + \alpha\|\boldsymbol{W}\|^2)$, where $\boldsymbol{X}$ are the collected inputs, $\boldsymbol{H}(\boldsymbol{X})$ the collected neuron activations for $\boldsymbol{X}$, $\boldsymbol{T}$ the target values and α the regularization strength.

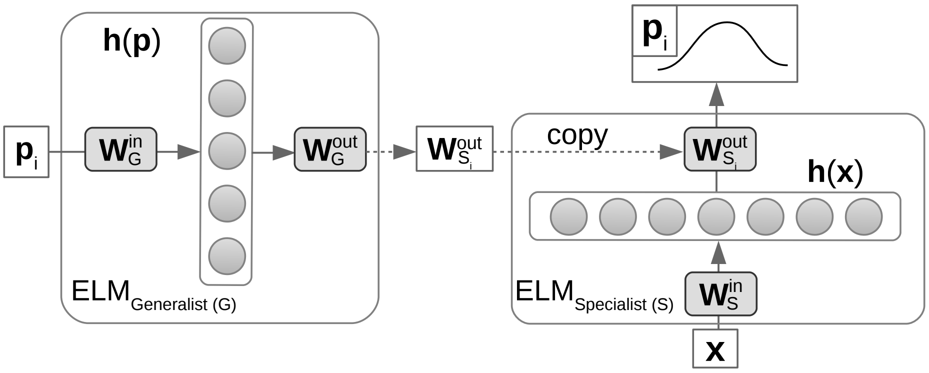


Fig. 1: Model Space Regression (MSR) with ELMs.

Fig. 1 sketches the MSR approach during exploitation. The specialist model ELM_S is responsible for modelling the process for a given process parameter value combination $\boldsymbol{p}_i$. The generalist model ELM_G parameterizes the specialist model by computing readout weights $\boldsymbol{W}_{S_i}^{out}$ for a given $\boldsymbol{p}_i$. Training of the MSR approach consists of two phases (cf. Algorithm MSR-ELM): In the first phase, for each process parameter value combination $\boldsymbol{p}_i$, we train a specialist model ELM_{S_i} using only the process input and process output, but not the process parameters $\boldsymbol{p}_i$. In the second phase, we train the generalist model ELM_G to predict the readout weights $\boldsymbol{W}_S^{out}(\boldsymbol{p})$ of the specialist model ELM_S from the process parameters $\boldsymbol{p}$. The readout weights $\boldsymbol{W}_S^{out}$ form the model space of process models, where the generalization over the process parameters takes place.

ELMs are especially suited as specialist process models, because only their readout weights are learned and their solution via linear regression is unique for given input weights. Interestingly, the resulting model space is 'flat': The difference between models is approximately equal to the Euclidean distance between the model parameters [11]. This 'flatness' of the model space allows for effective generalization.

Algorithm MSR-ELM

```
Inputs:
    Set$\{\boldsymbol{p}_i\}_{i=1,\dots,M}$ of process parameter combinations
    Matrices $\boldsymbol{X}_i \in \mathbb{R}^{T_i \times I}$, $\boldsymbol{Y}_i \in \mathbb{R}^{T_i \times O}$ of process input and output samples
Initialize:
    Random input weights and biases of specialist $ELM_S$ and generalist
    $ELM_G$ models with $N_S$ and $N_G$ hidden neurons
for all process parameter combinations $i$ do
    Train $ELM_{S_i}$ by $\boldsymbol{W}^{out}_{S_i} = \arg\min_{\boldsymbol{W}}(\|\boldsymbol{H}(\boldsymbol{X}_i)\boldsymbol{W}^T - \boldsymbol{Y}_i\|^2 + \alpha_S\|\boldsymbol{W}\|^2)$
    Store specialist readout matrix $\boldsymbol{W}^{out}_{S_i}$ in vectorized format $\boldsymbol{\omega}_i \in \mathbb{R}^{N_S \cdot O}$
end for
Concatenate all $(\boldsymbol{p}_i, \boldsymbol{\omega}_i)$ to $\boldsymbol{P}$ and $\boldsymbol{\Omega}$
Train $ELM_G$ by $\boldsymbol{W}^{out}_G = \arg\min_{\boldsymbol{W}}(\|\boldsymbol{H}(\boldsymbol{P})\boldsymbol{W}^T - \boldsymbol{\Omega}\|^2 + \alpha_G\|\boldsymbol{W}\|^2)$
```

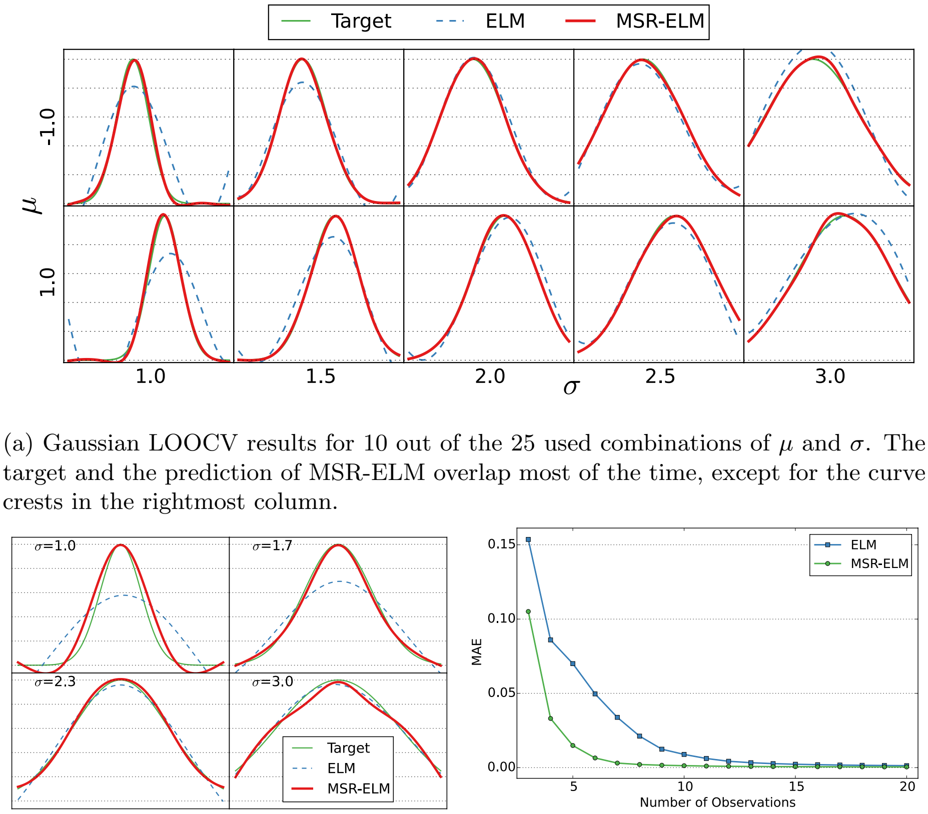
3 Experiments

We evaluate MSR and a monolithic ELM on a synthetic and a real-world dataset. The monolithic ELM is trained with process parameters and process inputs fused together. We use leave-one-out-cross-validation (LOOCV), where in each fold a process parameter combination is left out. We report mean average errors (MAE) $E = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{T_i}\sum_{t=1}^{T_i}|y_{m,t} - \hat{y}_{m,t}|$, where M is the number of process parameter value combinations and T_i the number of samples available per process parameter combination i. Since ELMs are initialized randomly, we chose the best results out of ten ELM initializations. The ELM parameters and the training time are listed in the Appendix.

3.1 Synthetic Example

As a synthetic toy-example for a parameterized process, we model the Gaussian function $f(x, \mu, \sigma) = e^{\frac{-(x-\mu)^2}{2\sigma^2}}$ with x as the process input, and μ and σ as the process parameters. Fig. 2a shows the results of learning f in the range $[-5, 5]$ with step size 0.1. μ was varied from -1 to 1 and σ from 1 to 3 each in 5 steps, which results in 25 parameter combinations. The test error $E_{MSR-ELM} = 0.00575$ with model space regression is almost ten times lower than the error $E_{ELM} = 0.05086$ using a single ELM to learn $f(x, \mu, \sigma)$.

Next, we vary the amount of training data. We compare the error rate of the monolithic ELM to MSR-ELM on the Gaussian function with fixed $\mu = 0$ and a varying number of equidistant observations of σ in the range $[1, 3]$. Fig. 2b shows



(a) Gaussian LOOCV results for 10 out of the 25 used combinations of μ and σ. The target and the prediction of MSR-ELM overlap most of the time, except for the curve crests in the rightmost column.

(b) Results with only four σ observations in the range $[1, 3]$.

(c) Test Error rate depending on the number of σ observations in the range $[1, 3]$.

Fig. 2: Gaussian LOOCV results. Each plot in (a) and (b) shows test results for novel process parameter combinations.

the results for four σ-observations, which means that the learners could only use three observations to predict the curve for the fourth process parametrization. Fig. 2c shows that MSR-ELM is by far better than the monolithic ELM when only few observations are present. We attribute this superiority to the modular character of MSR, which results in lower-dimensional input spaces for the generalist and specialist models.

3.2 Real-World Example: Modelling the Copper Bonding Process

Ultrasonic wire bonding is a cold welding technique to connect the electrodes of electrical devices. The copper bonding process depends strongly on several parameters, e.g. the applied normal force and ultrasound amplitude. In order to improve the quality of the copper bonds, a model of the bonding process was

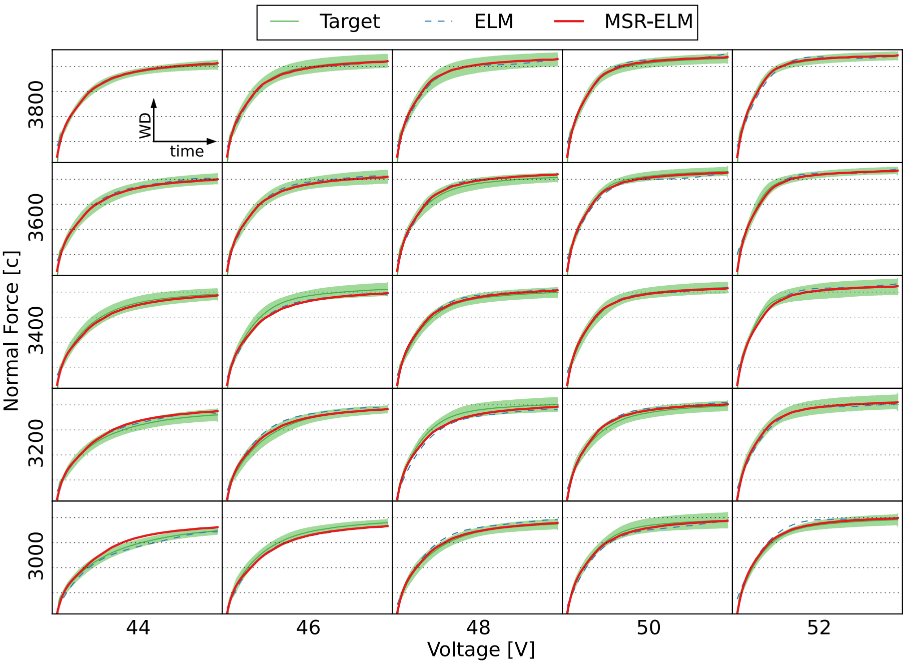


Fig. 3: InCuB LOOCV test results for the 25 parameter combinations. Each cell depicts the recorded (target) and predicted wire deformation (WD) over time for the process parameters normal force and voltage. The test prediction for each parameter combination is made using the other 24 combinations for training. The shaded area corresponds to the twofold standard deviation of the target values.

developed in the project 'Intelligent Copper Bonding' (InCuB, [12]). We compare the monolithic and the MSR approach on the dataset from [3] for modelling the ultrasonic softening effect of the bonding process.

The dataset consists of 25 curves, which describe the copper wire deformation over 250 time steps in dependence to the process parameters normal force and voltage of an ultrasound module. Fig. 3 shows the prediction results for the wire deformation by the monolithic ELM and the MSR-ELM depending on the process parameters. The average LOOCV-error $E_{MSR-ELM} = 0.00363$ for the wire deformation is significantly lower then the error $E_{ELM} = 0.00543$ of the monolithic approach (Wilcoxon paired signed-rank test $p = 0.00119$). The 'flatness' of the model space in this example is confirmed by the insignificant performance degradation if a linear model is trained as generalist ($E_{Linear-MSR-ELM} = 0.00384, p = 0.242$).

4 Conclusion

We presented a modular approach with regression in the model space of ELMs to model parametrized processes. The comparison to a monolithic ELM showed the superiority of MSR. This is due to the modularity of MSR, which reduces the dimensionality of the input space for the process model and exploits the flatness of the model space to achieve excellent generalization from few observations.

5 Appendix

All input variables were scaled to the range $[-1/I, 1/I]$, where I is the number input neurons. ELM parameters were determined via repeated, manually driven grid parameter search. N: Number of reservoir neurons, Scaling a^{in} and a^{bias}: Scaling of the uniform distribution from which the input weights $\boldsymbol{W}^{in}$ and biases $\boldsymbol{b}$ are drawn; α: Ridge regression coefficient, T: Training time with a single 3.5GHz CPU on a standard workstation.

- Gaussian fct.: ELM: $N=300$, $a^{in}=5$, $a^{bias}=1$, $\alpha=10^{-4}$, $T=6.30s$
- Gaussian fct.: MSR-ELM: ELM_S: $N_S=50$, $a^{in}_S=5$, $a^{bias}_S=1$, $\alpha_S=10^{-6}$, $T_S=0.14s$
ELM_G: $N_G=100$, $a^{in}_G=5$, $a^{bias}_G=1$, $\alpha_G=10^{-6}$, $T_G=0.30s$
- InCuB ELM: $N=100$, $a^{in}=1$, $a^{bias}=0.8$, $\alpha=10^{-8}$, $T=4.4s$
- InCuB MSR-ELM: ELM_S: $N_S=50$, $a^{in}_S=20$, $a^{bias}_S=1$, $\alpha_S=10^{-10}$, $T_S=0.22s$
ELM_G: $N_G=20$, $a^{in}_G=1$, $a^{bias}_G=1$, $\alpha_G=10^{-2}$, $T_G=0.26s$

References

[1] Hesse GmbH, Paderborn, Germany. `http://www.hesse-mechatronics.com`.

[2] Infineon Technologies AG, Warstein, Germany. `https://www.infineon.com/`.

[3] A. Unger, W. Sextro, S. Althoff, T. Meyer, K. Neumann, R.F. Reinhart, M. Broekelmann, K. Guth, and D. Bolowski. Data-driven modeling of the ultrasonic softening effect for robust copper wire bonding. In *International Conference on Integrated Power Systems (CIPS)*, pages 1–11, 2014.

[4] P. Turney et al. The identification of context-sensitive features: A formal definition of context for concept learning. In *Proceedings of the Workshop on Learning in Context-Sensitive Domains, at the International Conference on Machine Learning (ICML)*, 1996.

[5] Huanhuan Chen, Fengzhen Tang, Peter Tino, and Xin Yao. Model-based kernel for efficient time series analysis. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 392–400, 2013.

[6] W. Aswolinskiy, R.F. Reinhart, and J.J. Steil. Impact of regularization on the model space for time series classification. In *Machine Learning Reports*, pages 49–56, 2015.

[7] A. Ude, A. Gams, T. Asfour, and J. Morimoto. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Trans. on Robotics*, 26:800–815, 2010.

[8] R.F. Reinhart and J.J. Steil. Efficient policy search in low-dimensional embedding spaces by generalizing motion primitives with a parameterized skill memory. *Autonomous Robots*, 38(4):331–348, 2015.

[9] P. Ciskowski and E. Rafajłowicz. Context-dependent neural nets - Structures and learning. *IEEE Transactions on Neural Networks*, 15(6):1367–1377, 2004.

[10] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *IEEE International Joint Conference on Neural Networks*, volume 2, pages 985–990, 2004.

[11] H. Chen, P. Tino, A. Rodan, and X. Yao. Learning in the model space for cognitive fault diagnosis. *IEEE Trans. on Neural Networks and Learning Systems*, 25(1):124–136, 2014.

[12] Intelligent copper bonding. `http://www.its-owl.com/projects/innovation-projects/details/intelligent-copper-bonding`.