Learning Controllers for Adaptive Spreading of Carbon Fiber Tows

Julia Krützmann[1], Alexander Schiendorfer[1], Sergej Beratz[2], Judith Moosburger-Will[2], Wolfgang Reif[1], and Siegfried Horn[2]

[1] Institute for Software & Systems Engineering, University of Augsburg, Germany
{julia.kruetzmann,alexander.schiendorfer,reif}@informatik.uni-augsburg.de
[2] Institute of Materials Resource Management, University of Augsburg, Germany
{sergej.beratz,judith.moosburger,horn}@mrm.uni-augsburg.de

Abstract. Carbon fiber reinforced polymers (CFRP) are lightweight but strong composite materials designed to reduce the weight of aerospace or automotive components – contributing to reduced greenhouse gas emissions. A common manufacturing process for carbon fiber tapes consists of aligning tows (bundles of carbon fiber filaments) side by side to form tapes via a spreading machine. Tows are pulled across metallic spreading bars that are conventionally kept in a fixed position. That can lead to high variations in quality metrics such as tape width or height. Alternatively, one could try to control the spreading bars based on the incoming tows' profiles. We investigate whether a machine learning approach, consisting of a supervised process model trained on real data and a process control model to choose adequate spreading bar positions, is able to improve the tape quality variations. Our results indicate promising tendencies for adaptive tow spreading.

1 Introduction to Spreading of Carbon Fiber Tows

Carbon fiber reinforced polymers (CFRP) are lightweight composite materials with extraordinary structural integrity. That makes them attractive for the construction of lighter aerospace and automotive parts (conventionally made from steel or aluminum) to reduce fuel consumption and CO2 emissions [6].

These composites are made from a polymer matrix that is reinforced with textiles containing carbon fibers, often in the form of long tapes (e.g., featuring a width between 25 and 50 mm) that are eventually layered to form a textile. Such a carbon fiber textile is composed of carbon fiber filaments that are grouped to form so-called *tows*. To align the tows side-by-side to form tapes, a spreading machine is commonly applied [10]: By pulling the tows across a number of metallic spreading bars at a constant velocity, overlapping tows are "entangled" and aligned properly, as Figure 1 illustrates.

Conventionally, the positioning of the spreading bars is fixed for all profiles of the tows, leading to varying quality in the width and height of the outgoing tapes [5]. In this paper, *we investigate whether it is beneficial to adapt the spreading bar positions to the initial profiles.* To do so, we first train a process model

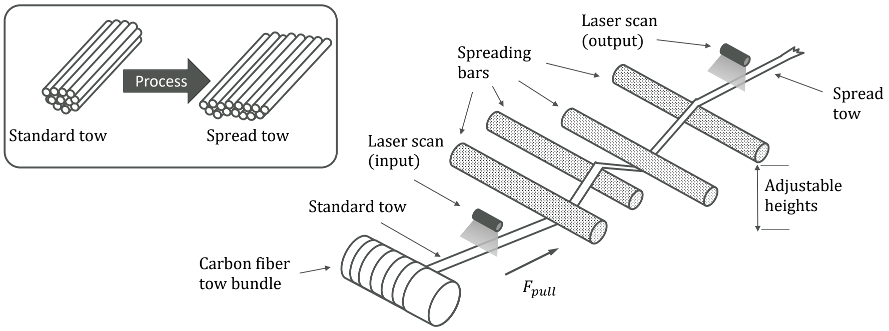

Fig. 1. Overview of tow spreading in the course of composite tape production.

representing the forward behavior of the real system in a supervised fashion (cf. Figure 2). It is based on experimental data obtained from a real spreading machine and thus alleviates the need to define an abstract mathematical model of the process, including numerically determining the process parameters. This model takes tow height profiles from a laser scanner as well as the spreading bar positions as inputs and predicts the resulting tow height profiles as output. We then apply Neuroevolution [4] to optimize for a process control model predicting the most suitable bar positions, given the scanner reading of the input tows and target tape widths (see Figure 3). Our results show that, compared to the best (but fixed) spreading bar positioning, on average adaptive spreading can halve the deviations in the desired tape width.

Following a discussion of related work, we present the process model including the data acquisition in Section 2, present the Neuroevolution based process control model in Section 3, and conclude with experimental results in Section 4.

1.1 Related Work

Research related to our approach is divided into two major domains: one, the spreading process and advances in its automation and, two, machine learning techniques in similar industrial settings.

Several publications address different types of the spreading process in general [9,11] as well as the underlying acting forces [7]. Gizik et al. [5] compared the effects of spreading on various types of heavy tows and showed that deviations in width of the resulting tow can vary greatly. Appel et al. [1] aim to develop an active control of the spreading process and, as a preliminary result, have identified numerically the process parameters that have an immediate influence on the quality of the tow. They also generated regression models for different quality parameters that should build the base for a control unit. In contrast to our work, they focus on the mathematical relations of various process parameters – mainly spreading velocity, pre-tension force of the initial tow, and wrapping angle which is directly dependent on the vertical bar position – with the tow quality but do not consider variations in the input tows.

When applying techniques that learn from online experiences, such as Reinforcement Learning (RL), to real-world problems (e.g. industrial processes or in a medical context), sufficiently exploring the action space is often problematic. One common approach to reduce this issue is to pre-train RL models on simulations, before adapting them carefully to the real environment requiring less real data [12]. However, the development of an adequate simulation can be infeasible, e.g., if the effects of behavior or forces in physical processes are not fully known. Several variations of off-policy evaluation were suggested as a way to gain confidence in RL policies by evaluating them on pre-collected databases before applying them online [8, 16]. Model-based reinforcement learning [2, 15] reduces real-world interactions by using an internal world model to generate samples for planning ahead and evaluating the next steps before applying them. This model is built while performing actions in the environment and continuously updated.

Due to the lack of a feedback loop to perform actions on the real system, our approach can be understood as a combination of some of the core ideas of the RL techniques described. We use pre-collected data to develop an internal model that is used to evaluate the suitability of adaptive control in general and Neuroevolution as optimization strategy in particular. As we consider no sequences of actions but only evaluate single steps, it is more viable to sufficiently cover the action space with experiments in advance than for more complex reinforcement learning tasks.

2 The Process Model – A Supervised System Predictor

Since it is not feasible (not to mention highly inefficient) to randomly adjust spreading bar positions in the real world process and learn which of these lead to high quality tows (i.e., a process control policy), we first aspire to develop the process model that approximates a representation of the input-output behavior based on real data, before using it to train a second model that proposes suitable bar positions (the "process control model"). Figure 2 shows the process model which is trained, in a supervised fashion, to map input tow height profiles to output tow height profiles based on two laser scanners.

2.1 Data Acquisition

In order to learn how tows are affected by different bar positions, a sufficiently large database is required. The data[1] we use to train the process model was recorded using a system slightly different to the one shown in Figure 1. It is composed of five metallic, non-rotating spreading bars through which a tow is pulled at a constant velocity. The spreading bars are placed horizontally and arranged alternately above and below the tow. A light section sensor is mounted above the first and last spreading bar to measure the height profile of the tape

[1] Data available under: https://doi.org/10.6084/m9.figshare.c.4958948.v1 Code is available under: https://github.com/isse-augsburg/adaptive-spreading

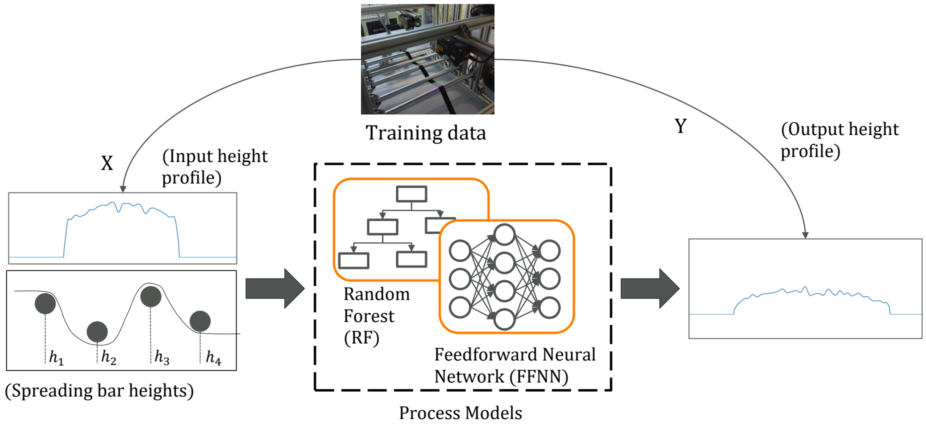

Fig. 2. The process model maps input height profiles and spreading bar heights to output height profiles. The model is trained from a real world tow spreading machine.

before and after the spreading process, respectively. To obtain a varied database, experiments were conducted with a total of fifteen different bar positions and twelve velocity settings for each bar setup. The height of the bars was altered within a range of 29 centimeters, while the pulling velocity was set to a value between 2.7 m/min and 15 m/min and the sensors took measurements at a frequency of 500 Hz and a resolution of 800 pixels where one pixel covers about 0.1mm. Experiments were recorded over about 60 seconds each with a fixed bar position and velocity which results in raw data consisting of 687 files (one per 60 seconds of recording) with 15 million measuring points in total.

Due to the high measurement frequency and the absorption behavior of carbon fiber tows, the data contains substantial noise as well as artifacts. Additionally, the whole system was often adjusted manually to conduct various experiments leading to inaccuracies in the form of considerably differing value ranges of the measurements for different recordings. To facilitate the learning task, we therefore unify the data while maintaining relevant information such as irregularities in the profile. First, the measured profiles are smoothed by applying a Savitzky-Golay-Filter [13] and averaging sequential data points. Second, to reduce further inaccuracies, they are shifted and rotated in such a way that the underlying bar is always aligned horizontally at zero height - serving as a form of supplementary sensor calibration. Moreover, since the two sensors are only synchronized to measure at the exact same time, the information which measurements belong to the same part of the tape before and after spreading is not immediately available in the data. That mapping is, however, crucial in order to learn the effect of bar positions on the tow, i.e., to obtain temporally aligned target values for the supervised training. Hence, in the last step, we calculate the time offset of the sensors for each experiment taking into account the velocity and the distance covered by the tow which depends directly on the bar positions.

Table 1. Data set sizes – train/val/test-split

Data set	# Files	# Samples	% (Samples)
Training	416	511,727	63.9
Validation	104	138,500	17.3
Test	130	150,662	18.8
All	650	800,889	100

These pre-processing steps lead to a reduced data set size of 650 files holding 800,899 samples. Table 1 shows the split of the data for training our models. When dividing the data, we ensure that sequential measurements of a single experiment are fully contained in the same set, avoiding the occurrence of highly similar data points in training and test set.

Analysis of the preprocessed data indicates that the average width of the tows before spreading is 212.9 pixels and 293.7 pixels after spreading with a standard deviation of 19.6 and 31.7 pixels respectively. Furthermore, there is no significant effect of the velocity on the spreading factor. Therefore, we focus on the measured height profiles and the bar positions as the main characteristics of the setup.

2.2 Models

The goal of the process model is to predict the tow height profile after spreading as accurately as possible based on the initial status, while also being robust against light noise (cf. Figure 2). The input of the models, therefore, consists of 805 values (800 pixels as read from the laser scanner for tow profiles before spreading and current positions of the five bars), while the output has a dimension of 800. Besides these specifics, it is crucial that the model yields output that can be deemed realistic even for unknown inputs, i.e. bar setups previously not seen and new combinations of profiles and bar settings. For example, the tape predicted is expected to become wider when raising the middle bar. To reach a good solution, as part of this work, we compare two candidates that both are generally suitable for such supervised multivariate regression tasks: feedforward neural networks (FFNN) trained with gradient descent and random forests (RF) aggregating multiple decision trees.

3 A Process Control Model

Based on the process model, it is possible to train a control model that, by experimenting with the bar positions, learns which changes lead to a tow profile closer to the one desired (see Figure 3). The process model fulfills a function partially comparable to a “world model” which is more commonly used in reinforcement learning. In detail, the workflow is such that the process control model receives a height profile and the current bar setting as input and yields a new setup which

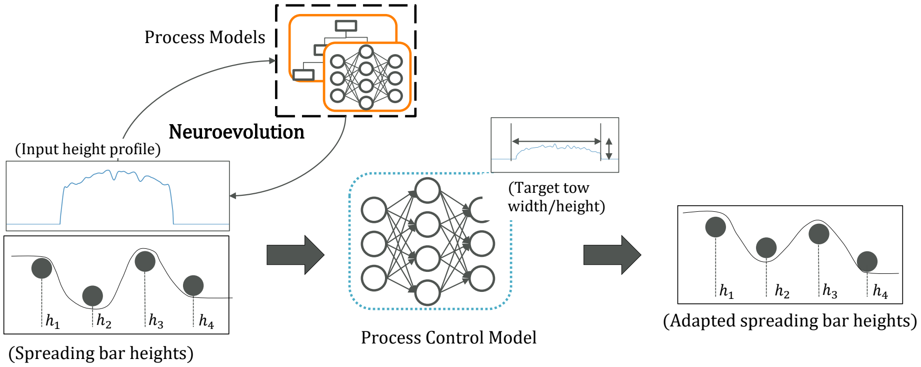

Fig. 3. The process control model that adaptively selects bar heights based on the input height profile, the current bar positions, and a fixed target width and height.

is optimized for a fixed target tow width. Based on this prediction of the process control model and the height profile, the process model then generates the resulting tow which, in turn, is used to determine the suitability of the suggested bar setting.

To optimize the process control model, we choose a neuroevolutional approach based on a rather simplistic genetic algorithm (partially based on the genetic algorithm described in [14]). This is necessary since supervised training is not applicable: for a given input tow, current bar positions, and target tow width and height, we do not know “target bar positions” from data which is why we strive for an RL-approach.

Neuroevolution approach For the implementation of the process control model, first, a population of neural networks is initialized with weights sampled from a scaled normal distribution. Second, the fitness of these “genotypes” is determined based on the training set (cf. Table 1) and the subset of the best performing neural networks is selected as parents for the next generation. Subsequently, a neural network is randomly selected from the parents and mutated by adding normally distributed noise to its weights. These mutations and the parents that performed best form the population of the next generation. To perform the training of the controller, the second and third steps are repeated until the performance of the fittest neural network does not improve on the validation set for multiple epochs (comparable to early stopping in gradient-based training). Finally, the fittest neural network of the last generation is chosen to be the process controller.

To determine the fitness (or reward) r of a single genotype, for each suggested action we generate the resulting tow profile p using the process model and compare its width and height to a designated target t. Additionally, we consider the distance of the height h of each of the b bars before and after the action, in order to favor solutions closer to the original position and, thus, reducing the required bar movement. Formally, r is defined as follows:

$$\begin{aligned} r = \ - (k_{height} \cdot |height_t - height_p| + k_{width} \cdot |width_t - width_p| \quad + \\ k_{distance} \cdot \textstyle\sum_{i=1}^{b=5} |h_{\{current,i\}} - h_{\{p,i\}}|) \end{aligned} \tag{1}$$

where the hyperparameters k_{height}, k_{width} and $k_{distance}$ control the relative importance of each criterion. Each subterm should be minimized which is why r is negated, following the conventions of fitness-maximization in neuroevolution. For the following experiments, we set k_{height} = 1.0, k_{width} = 4.0 and $k_{distance}$ = 0.5, mainly emphasizing the tow width. The accumulated fitness of a neural network is given by averaging the calculated rewards.

To obtain values in $[0, 1]$, indicating the relative bar positions, the output of the neural network is activated by the sigmoid function. Each of these values is transformed into its unique range specified by the data, meaning that the controller never suggests moving a bar below or above its lowest or highest position in the experiments with the real system.

Baseline: Fixed bar setup To assess the benefits of an adaptive process controller in general, we also determine the fixed bar positions that perform best to achieve a given target width. The fixed setup is picked from a set of all the settings used to record the real data.

4 Evaluation

Since the mere error can be hard to interpret when evaluating solutions of regression tasks, we focus on the width of the predicted tows as the main criterion. For this purpose, we need, firstly, to clarify how the profile width is determined and, secondly, how we can derive a meaningful metric for the models. As described in Section 2.1, we pre-processed the data such that pixels beside the tape are set to zero. Accordingly, the most naive approach to calculate the tow width is to consider all positive values in a prediction as ”tow” and to infer the width. Due to uncertainty in the developed models we see, however, softer edges in the predictions than in the real data, leading to overestimation of the width. As a dynamic threshold, we take the mean value of the predicted profile and consider the first and last pixels above this value as the tow edges. The derived width is set in relation to the width of the corresponding target. In order to factor in overestimations as well as underestimations, we calculate the quality metric m of a prediction p as follows:

$$m(p) = 1 + |1 - \frac{width_p}{width_{target}}| \tag{2}$$

Thus, the optimum score is 1, while it increases accordingly in case of deviations. For comparison, we take the geometric mean and the geometric standard deviation (due to m being a relative quantity) of the quality of all predictions on the same test set for each model.

Before applying this metric to evaluate the performance of the process control model, we put to test whether the underlying process model can be considered a sufficiently accurate representation of the real spreading process.

4.1 Process Models

As expected, keeping the dimension of the input data at 805 values is not feasible when training the random forest. Besides the well-known training runtime complexity of $O(n_{samples} \cdot n_{features} \cdot \log(n_{samples}))$, the size and, thus, the memory usage of the underlying decision trees increases immensely. To overcome this, using random subsets of the features, as described by [3], is a very applicable approach. Results from our experiments suggest that a subset size of 200 features is sufficient to achieve good scores, with no improvement in the quality of predictions when further increasing the input dimension. This seems reasonable, given the fact that the average tow width is slightly above 200 pixels, as mentioned in Section 2.1. Thus, about 600 values of a profile are mostly set to zero and can easily be dropped - with the caveat that such a model can presumably not be adapted to tows that are significantly wider.

By contrast, since the feedforward neural network does not suffer from runtime or space problems due to high dimensionality, we trained that model with the full input dimension to keep it as versatile as possible.

After optimizing the architectures of the feedforward neural network and the random forest, both approaches achieve similar m-scores (see Figure 4) – with a slight advantage for the random forest. The average width of tows predicted on the test set matches the ground truth of the measured profiles to within two pixels, with the standard deviation of the predicted width being lower than the real one (cf. Figure 4b) which can be accounted for by the fact that the models are a generalisation of the real process. In accordance with the geometric mean and standard deviation in Figure 4a, the quality m_{rf} is in $[1.002, 1.081]$, while m_{ffnn} is in $[1.001, 1.082]$ which means the worst predictions are about 2.37 mm and 2.40 mm off while the mean prediction deviates about 1.18 mm and 1.19 mm for average tow width, respectively for the random forest and the neural network. Both models are equally likely to over- and underestimate the width. It is an open question if the predictions are sufficiently accurate to use the models in the real process, e.g. for quality monitoring , but we presume that they are a sufficiently approximate representation of the spreading behavior to help evaluate whether a control model as described in Section 3 can be beneficial.

Not only are the scores of both approaches highly similar but the predicted profiles are remarkably alike for the recorded bar positions (cf. Figure 5). However, to decide which one of the two solutions is the more suitable backend for the controller, it is necessary to consider the behavior of the models for different bar settings as well. As there is no target profile available for random settings, the qualitative evaluation which prediction is closer to the real process is difficult. We, thus, examined empirically which model is more likely to show the

	FFNN	RF
Geo. Mean	1.0406	**1.0403**
Geo. Std.	1.0396	**1.0387**

(a) Geometric mean and geometric standard deviation of m.

	Ground Truth	FFNN	RF
Avg. Tow width	288.9	286.3	**287.3**
Std. Tow width	32.1	**27.5**	27.1

(b) Mean and standard deviation of the real and predicted tow width.

Fig. 4. Scores of the trained process models: feedforward neural (FFNN) and random forest (RF).

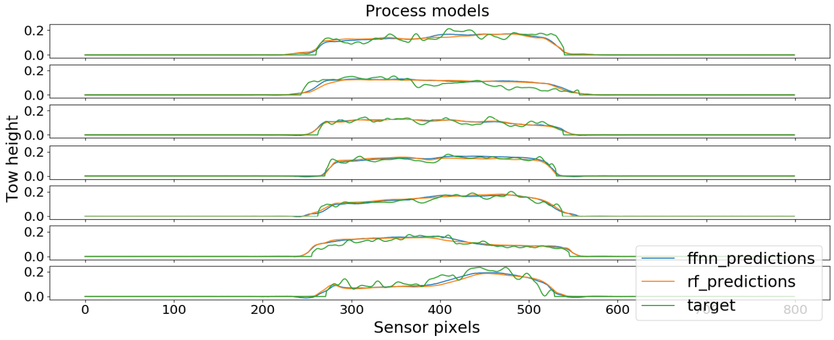

Fig. 5. Tow profiles after spreading as recorded during experiments and the according predictions by the neural network and random forest process model. Samples are taken at random.

expected behavior. Especially when altering the bar positions to more extreme heights (each within the ranges of the real experiments but new combinations thereof), the differences between the two models become apparent. At a certain point, the profiles predicted by the random forest do not get any wider. Figure 6 shows that the random forest predicts narrower tows than the feedforward neural network for the same input data. We presume that, one, wide tows are less common in the original data, also indicated by the similarities in the predictions by the neural network, such as the elevation on the right side. And, two, the random forest tends to be a stronger generalisation of the process in general as its output is robust to unknown bar settings while the profiles of the neural networks become slightly noisier. Since we would expect a wider output for the maximum bar setup, we choose the FFNN as the underlying process model for the controller. To improve its robustness, we train it on additional synthetic data consisting of previously unseen bar positions that would not touch the tape and, therefore, lead to no differences in the profiles before and after spreading (i.e. the expected output is the same as the input profile), serving as a form of regularization.

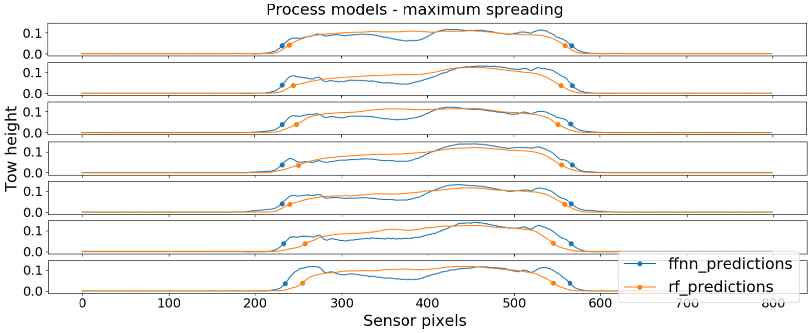

Fig. 6. Tow profiles predicted by the neural network and random forest, respectively, when setting the bars to extreme but realistic positions. This bar setup was not part of the recorded data. Samples are taken at random.

Table 2. Geometric mean and geometric standard deviation of the absolute ratio of achieved and desired tape widths for fixed bar setups and the adaptive approach. Target width in pixels.

	Fixed Bar Setup		Adaptive Control	
Target width	Geo. Mean	Geo. Std.	Geo. Mean	Geo. Std.
280	1.048	1.032	**1.023**	**1.023**
290	1.039	1.033	**1.028**	**1.025**
300	1.041	1.036	**1.023**	**1.022**
310	1.034	1.037	**1.022**	**1.022**
320	1.045	1.032	**1.021**	**1.026**
330	1.052	1.043	**1.028**	**1.030**

4.2 Process Control Model

When optimizing the hyperparameters of the neuroevolution approach, we found that the dimension of the neural networks is of secondary importance, with rather small neural networks with two hidden layers performing as well or slightly better than larger ones, whereas identifying a suitable population size and batch size has more impact. We observed the best performance with a population of 300 neural networks where the 15 fittest genotypes (on a rather small batch size of 2048 samples) would mutate and reproduce each generation.

Overall, the evaluation of the process control model suggests that adaptive control is promising to generate high-quality tows and that neuroevolution may be a suitable approach to realize it. Examining the scores achieved, it is apparent that adjusting the bar positions consistently outperforms fixes settings (Table 2). In detail, developing a controller for the target tow width reduces the mean offset of the desired width by up to 47,9% in comparison to the fixed setup or down to 0.64 mm in general. Additionally, the standard deviation is lower as well. In fact, considering e.g. a target width of 28 mm, even the tow produced with the controller that differs most from the target width achieves a score of 1.046

while the overall geometric mean of the quality is 1.048 for the baseline fixed bar setup. For target widths greater 320 pixels, the quality declines for the two approaches which was to be expected. This can be attributed both to the fact that our process model is limited as less real data was available for this size and that this width might also not be achievable with the given bar settings.

Interestingly, when visualising the bar positions proposed, we see that the control model tends to suggest solutions where mainly one bar (mostly the middle one) is adjusted depending on the input while the others are kept still. This finding is very promising, especially with respect to the time constraints that would apply when adapting the controller to the real process. It, however, remains open for further research to enhance the developed strategy for the real world application taking into account neighbouring measurements and the time it takes to move bars.

5 Conclusion

In this work, we presented a process controller for adaptive spreading of carbon fiber tows with an underlying process model, both implemented with neural networks. While the process model was trained conventionally using gradient descent, the process control model was optimized by applying a genetic algorithm. Regarding the process model, we showed that neural networks as well as random forests can adequately approximate a representation of the process behavior, even though they each carry certain shortcomings. Overall, we found a neural network to be more suitable for the application at hand as it yielded predictions for bar settings previously not seen that were closer to the expected behavior. In regards to the controller, adjusting the bar setup depending on the present tow profile is promising when compared to fixed setups. Our controller was able to reduce the deviations in tow width after spreading by half.

Future work may focus on two main issues: In the short term, the process controller as implemented is static in a sense, since it was trained to provide bar positions that result in tows with one given, fixed target width. Expanding its input to accept a desired tow width would lead to a more versatile solution. The long term goal is to replace the underlying process model with the real setup, which requires some changes on the controller. In particular, it is necessary to include time constraints of real world processes. Contrary to the process model, adjusting bar positions is not instantaneously possible and affects neighbouring parts of the tow, so it is mandatory to consider spatial dependencies. To reduce costly experiments with real carbon fiber tows, a cyclic approach is promising where findings from the real process are fed into the process model. In doing so, the exhaustive tuning of the hyperparameters of the controller is performed virtually, before evaluating the results on the real setup. The first iteration of this cycle, training a process model with real data and designing a suitable controller, was accomplished in this paper.

Acknowledgments This research is partly funded by the Bavarian Ministry of Economic Affairs, Regional Development and Energy in the project LufPro.

References

1. Appel, L., Kerber, A., Abbas, B., Jeschke, S., Gries, T.: Determination of interactions between bar spreading process parameters and spreading quality for the development of an automated quality control of spread high modulus fiber tows. ECCM 2016 - Proceeding of the 17th European Conference on Composite Materials **17**, 26–30 (2016)
2. Atkeson, C.G., Santamaria, J.C.: Comparison of direct and model-based reinforcement learning. Proceedings - IEEE International Conference on Robotics and Automation **4**(April), 3557–3564 (1997)
3. Breiman, L.: Random Forests. Machine Learning **45**, 5–32 (2001)
4. Floreano, D., Dürr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. Evolutionary intelligence **1**(1), 47–62 (2008)
5. Gizik, D., Metzner, C., Weimer, C., Middendorf, P.: Spreading of heavy tow carbon fibers for the use in aircraft structures. ECCM 2016 - Proceeding of the 17th European Conference on Composite Materials **17** (2016)
6. Heywood Don MacKenzie, J., Bonde Akerlind Parisa Bastani Irene Berry Kandarp Bhatt Alice Chao Eric Chow Valerie Karplus David Keith Michael Khusid Eriko Nishimura Stephen Zoepf, I.: On the Road toward 2050: Report Massachusetts Institute of Technology Potential for Substantial Reductions in Light-Duty Vehicle Energy Use and Greenhouse Gas Emissions. Massachusetts Institute of Technology (2015), http://mitei.mit.edu/publications/
7. Irfan, M.S., MacHavaram, V.R., Mahendran, R.S., Shotton-Gale, N., Wait, C.F., Paget, M.A., Hudson, M., Fernando, G.F.: Lateral spreading of a fiber bundle via mechanical means. Journal of Composite Materials **46**(3), 311–330 (2012)
8. Mandel, T., Liu, Y.E., Brunskill, E., Popovíc, Z.: Offline evaluation of online reinforcement learning algorithms. 30th AAAI Conference on Artificial Intelligence, AAAI 2016 pp. 1926–1933 (2016)
9. Marissen, R., van der Drift, L.T., Sterk, J.: Technology for rapid impregnation of fibre bundles with a molten thermoplastic polymer. Composites science and technology **60**(10), 2029–2034 (2000)
10. Morgan, P.: Carbon fibers and their composites. CRC press (2005)
11. Newell, J.A., Puzianowski, A.A.: Development of a pneumatic spreading system for kevlar-based sic-precursor carbon fibre tows. High Performance Polymers **11**(2), 197–203 (1999)
12. Peng, X.B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. Proceedings - IEEE International Conference on Robotics and Automation pp. 3803–3810 (oct 2018)
13. Savitzky, A., Golay, M.J.E.: Smoothing and differentiation of data by simplified least squares procedures. Analytical chemistry **36**(8), 1627–1639 (1964)
14. Such, F.P., Madhavan, V., Conti, E., Lehman, J., Stanley, K.O., Clune, J.: Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning (2017), http://arxiv.org/abs/1712.06567
15. Sutton, R.S.: Dyna, an integrated architecture for learning, planning, and reacting. ACM SIGART Bulletin **2**(4), 160–163 (1991)
16. Thomas, P.S., Theocharous, G., Ghavamzadeh, M.: High confidence off-policy evaluation. Proceedings of the National Conference on Artificial Intelligence **4**, 3000–3006 (2015)