Cybernetic Data Augmentation for Neural Network Classification of Control Skills

de Jong, Martijn J.L.; Pool, Daan M.; Mulder, Max

DOI
10.1016/j.ifacol.2022.10.252

Publication date
2022

Document Version
Final published version

Published in
IFAC-PapersOnline

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-29 (2022) 178–183

Cybernetic Data Augmentation for Neural Network Classification of Control Skills

Martijn J.L. de Jong, Daan M. Pool, and Max Mulder

Control and Simulation Section, Aerospace Engineering, Delft University of Technology, Delft, The Netherlands
(e-mail: mjl.dejong@outlook.com, {d.m.pool, m.mulder}@tudelft.nl).

Abstract: Mathematical human controller (HC) models are widely used in tuning manual control systems and for understanding human performance. Typically, quasi-linear HC models are used, which can accurately capture the linear portion of HCs' behavior, averaged over a long measurement window. This paper presents a deep learning HC skill-level evaluation method that works on short windows of raw HC time signals, and accounts for both the linear and non-linear portions of HC behavior. This deep learning approach is applied to data from a previous skill training experiment performed in the SIMONA Research Simulator at TU Delft. Additional human control data is generated using cybernetic HC model simulations. The results indicate that the deep learning evaluation method is successful in predicting HC skill level with 85-90% validation accuracy, but that training the classifier solely on *simulated* HC data reduces this accuracy by 15-25%. Inspection of the results especially shows a strong sensitivity of the classifier to the presence of remnant in the simulated training data. In conclusion, these results reveal that current quasi-linear HC model simulations, and in particular the remnant portion, do not adequately capture real time-domain HC behavior to allow effective training-data augmentation.

Keywords: Cybernetics, manual control, classification, neural networks, control skills

1. INTRODUCTION

Quantifying human manual control skill level is difficult. Cybernetics has shown to be a useful tool to assess skill level (Pool and Zaal, 2016; Pool et al., 2016), but most methods suffer from the fundamental limiting assumption that the human controller (HC) is constant over a substantial time interval, ignoring any 'fast' changes in control behavior (Mulder et al., 2018). Although efforts have been made to model time-varying HC behavior (Zaal and Sweet, 2011; Duarte et al., 2017; Rojer et al., 2019), capturing *short*-duration behavioral variations from inherently noisy data remains challenging (Mulder et al., 2018).

Traditionally, cybernetics describes HCs' behavior as a combination of linear and non-linear signals (McRuer and Jex, 1967). The latter, also known as *remnant*, is difficult to model and is often ignored in cybernetic evaluations (Pool et al., 2016). However, this non-linear portion of the control behavior changes as HCs gain more experience (Wijlens et al., 2020) and therefore contains information that is useful for the evaluation of HC skill level. Deep learning methods may be the key to a more complete real-time assessment of HC skill level. The automatic feature extraction capability of deep learning on raw time series could enable the use of the full range of HC behavior – both linear and non-linear – for skill level predictions.

Although recent years have shown numerous investigations of Machine Learning (ML) for identifying HC behavior (Nittala et al., 2018; Saleh et al., 2017; Jain et al., 2016; Tango and Botta, 2013), the assessment of differences in skill level has remained largely untouched. Furthermore, most proposed approaches (Xi et al., 2019) require manual feature engineering and are most accurate when using 60 s of data, falling into the same time range where 'classic' cybernetic methods are also effective (Mulder et al., 2018).

The overarching goal of this project is to develop an HC skill level classification method using deep learning. For this, deep convolutional neural networks (CNNs) are trained to classify raw time series of manual control behavior as either 'skilled' or 'unskilled'. Experimental '*real*' human-in-the-loop data from a previously conducted compensatory pitch tracking experiment by Pool et al. (2016) were used to train and validate our classifier. The main classification results will be presented elsewhere.

In this paper we investigate the effects of applying a classifier trained on *simulated* HC data generated from cybernetic model simulations to *real* HC data. ML methods work better when the number and quality of the data for learning and testing increases. However, performing human-in-the-loop experiments can be expensive and time-consuming. As generating more 'human-like' data using cybernetic models may significantly reduce the need for experiments, in this paper we analyze whether such *artificial* data indeed enables training an effective classifier.

2. CLASSIFICATION RATIONALE

2.1 Classification problem

In our research, we treat HC skill level assessment as a Time Series Classification (TSC) problem. Namely, a time series X is fed to a classifier that predicts the probability P

10.1016/j.ifacol.2022.10.252

that X belongs to class Y. In the context of this research, X is a Multivariate Time Series (MTS) containing raw time signals of HC behavior: i.e., the tracking error e, the controlled element (CE) output x, and the HC control signal u, see Fig. 1. The corresponding class variable Y is a binary label that indicates whether the HC is a skilled ('trained', Y_1) or unskilled ('untrained', Y_2) controller. The classifier labels the data observed in a (small, moving) window, as belonging to a skilled or unskilled HC.

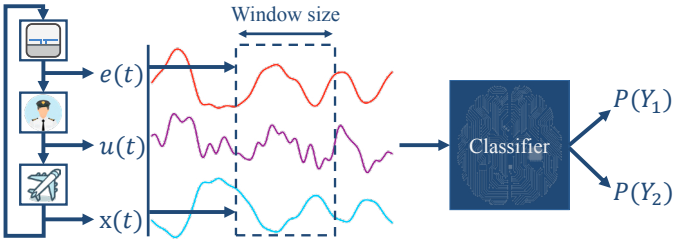

Fig. 1. HC skill level classification scheme.

HCs' skill level is of course not a binary property, but rather a quantity that varies along a continuous scale. It is difficult, however, to exactly define skill level and the definition of ground truth scalar values would heavily depend on assumptions. Hence, in this paper we treat skill level as a binary property, depending on the amount of experience the HC has had in the control task at hand.

2.2 Input preprocessing

Fig. 1 also illustrates the used *window slicing* (Cui et al., 2016) technique, which takes a time series T with N time steps $T = \{t_1, \ldots, t_N\}$ and produces multiple (shorter) 'slices' defined as $S_{i:j} = \{t_i, t_{i+1}, \ldots, t_j\}$, $1 \leq i \leq j \leq N$. The size of each slice is set here to 1.2 s, a value found through empirical optimization. Window slicing is beneficial for two reasons. First, a classifier trained on short time series samples can also predict based on short measured snippets of HC behavior. This allows for near *real-time* classification, enabling applications such as sliding scale autonomy or skill deterioration warnings. Second, with a larger set of smaller snippets of time-series data, effectively more training data become available. Increasing the amount of training data ("data augmentation") helps to avoid overfitting and improves generalizability (Le Guennec et al., 2016). This data augmentation aspect of window slicing is further enhanced by *overlapping* consecutively sliced windows by 90%. For example: a time series of 90 s provides 75 non-overlapping 1.2 s samples, whereas 741 samples of the same size are obtained with 90% overlap.

2.3 Network structure

A deep artificial neural network was used to classify the raw time signals of the HC behavior. Deep learning for TSC has shown to achieve performance comparable to state-of-the-art TSC methods (Wang et al., 2017; Fawaz et al., 2019) with the added benefit of not requiring manual feature engineering. In this paper, a convolutional Residual Network (ResNet) is used. This class of networks, developed by He et al. (2015), contains 'shortcut' residual connections that ease some of the difficulties in training deep networks, whilst maintaining their capacity for enhanced accuracy (Simonyan and Zisserman, 2015). Different architectures – i.e., Long Short-Term Memory, Fully Convolutional Networks, and InceptionTime – were also tested, however, all were found to be 5-10% less accurate.

Wang et al. (2017) implemented ResNet for TSC using 1D convolutional layers. Fig. 2 shows the implementation through *residual blocks* containing three convolutional layers – with an equal number of kernels K per block, but with varying kernel sizes L per layer – each followed by a Batch Normalization (BN) and Rectified Linear Unit (ReLU) activation layer. A shortcut connection combines the input of the first convolutional layer with the output of the third (last) convolutional layer, before passing it through the final BN + ReLU layer. The depth d of the network equals the number of residual blocks. Using grid search optimization it was found that the hyperparameter settings proposed by Wang et al. (2017) are optimal for the HC classification task: network depth $d = 3$, number of filters $K = \{64, 128, 128\}$, kernel sizes $L = \{8, 5, 3\}$.

3. METHOD

3.1 Experimental data

The data used here for the classification of HC skill level are from a previous experiment performed in the

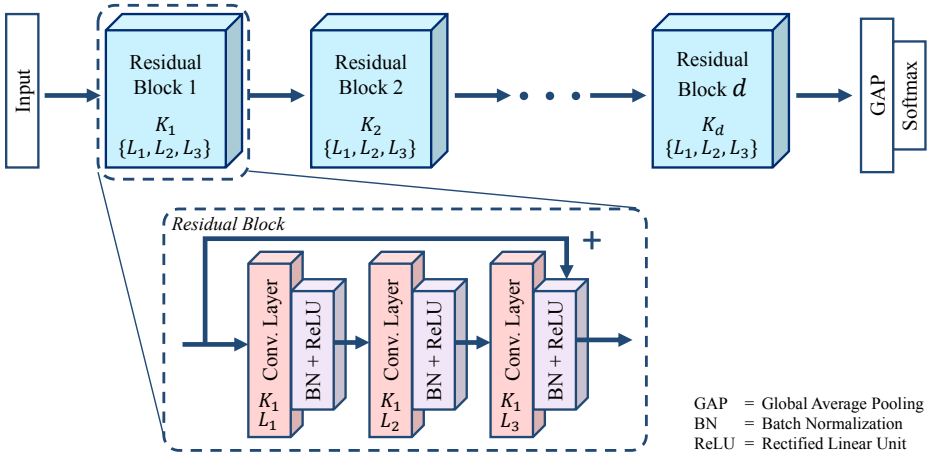

Fig. 2. Network structure of the Residual Network.

SIMONA Research Simulator at TU Delft (Pool et al., 2016). The experiment studied the effects of simulator motion feedback on the training of multimodal skill-based HC behavior, for which 24 fully task-naive participants underwent training in a *compensatory* pitch tracking task designed for multimodal HC behavior identification, see Fig. 3. Each participant completed a total of 100 training runs and 75 evaluation runs, with each run providing 81.92 s of measurement data recorded at 100 Hz.

Only the *training* phase data of Pool et al. (2016) are used for this project as they include measurements of both *inexperienced* HC behavior (at the start of training) and *experienced* behavior (at the end of training). While Pool et al. (2016) collected data from two 12-participant groups who were trained in either a fixed- or a moving-base simulator, for this paper only the fixed-base experiment data are used. This allows for using a single-channel HC model for the cybernetic data generation, as discussed in Section 3.2. The experimental data set contains time traces from 12 participants, who each performed 100 tracking runs, totaling 98,304 s of HC tracking data. For our classifier training, the first 20 training runs of each participant are labeled as 'skilled' (Y_1) and the last 20 are labeled as 'unskilled' (Y_2). This value of 20 was empirically found to ensure distinct 'skilled' and 'unskilled' samples. The in-between tracking runs (21-80) are not used.

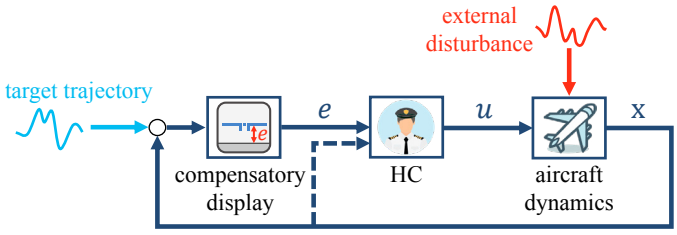

Fig. 3. The pitch angle compensatory tracking task.

Fig. 3 illustrates the pitch tracking task, where e indicates the tracking error, u the HC output, and x the CE output (here the aircraft pitch angle). The dotted line connecting x directly to the HC is only active if motion feedback is provided. The tracking task requires the HC to follow a target pitch angle as accurately as possible, while rejecting an additional disturbance signal placed on the CE. The CE dynamics were the elevator-to-pitch dynamics of a Cessna Citation I, using a reduced-order linearized model:

$$H_{\theta,\delta_e}(s) = 10.62\frac{s+0.99}{s\left(s^2+2.58s+7.61\right)} \tag{1}$$

The target- and disturbance signals, f_t and f_d, are both a sums of 10 sines with different excitation frequencies, ω_t and ω_d, respectively. In the experiment, five different realizations of the f_t and f_d signals – with identical sinusoid frequencies and amplitudes, but with different phase settings – are used in a balanced and randomized order to ensure participants do not learn the signals. For the complete details of the used f_t and f_d settings, please refer to (Pool et al., 2016). The combined target-following and disturbance-rejection task enables reliable separation and identification of the HC response to visual and motion cues and was based on a number of previous studies that successfully identified multi-channel HC behavior (Zaal et al., 2009). An example of the time traces recorded during a tracking run are shown in Fig. 4. Throughout this paper, only e, $\dot{e}$, u, and $\dot{u}$ are used as input features for the classification model, as these inputs yielded the best classification performance.

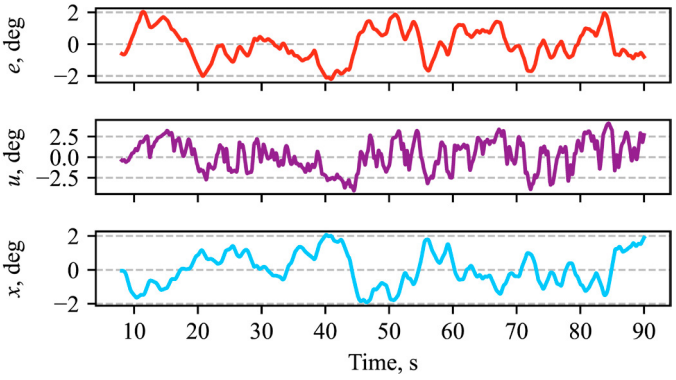

Fig. 4. Example time traces of one tracking run.

3.2 Cybernetic HC simulations

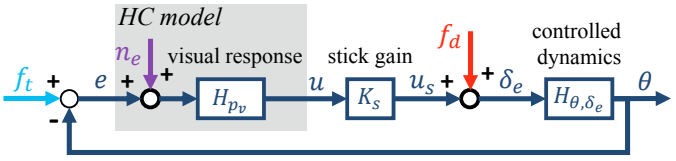

Fig. 5. Quasi-linear HC simulation model; remnant is modeled as colored noise n_e.

Model structure The HC model simulations are performed using a quasi-linear model proposed for this pitch tracking task by Pool et al. (2016), see Fig. 5. Since Pool et al. used this same model to evaluate the experiment data that are used in this paper, the model parameters – except the remnant parameters – of each subject were readily available for simulation. The *visual response* $H_{p_v}(s)$ of the HC model is given by Eq. (2). Because HCs generate lag at frequencies below the CE's short-period mode natural frequency, but exert lead equalization at higher frequencies, a squared lead term is included to capture these HC dynamics (Pool et al., 2011). In total, the visual response model of Eq. (2) contains six model parameters: the (visual) gain K_v, the lead time constant T_{lead}, the lag time constant T_{lag}, the time delay τ_v, and the neuromuscular dynamics modeled as a second-order mass-spring-damper (McRuer and Jex, 1967) with natural frequency ω_{nm} and damping ratio ζ_{nm}.

$$H_{p_v}(s) = \underbrace{K_v\frac{\left(T_{lead}s+1\right)^2}{T_{lag}s+1}}_{\substack{\text{human}\\ \text{equalization}}}\underbrace{e^{-s\tau_v}}_{\substack{\text{response}\\ \text{delay}}}\underbrace{H_{nm}(s)}_{\substack{\text{neuromuscular}\\ \text{dynamics}}} \tag{2}$$

$$H_{nm}(s) = \frac{\omega_{nm}^2}{s^2+2\zeta_{nm}\omega_{nm}s+\omega_{nm}^2} \tag{3}$$

Remnant model HC remnant n_e, see Fig. 5, is modeled as filtered white noise injected at the e signal, see Eq. (4). As proposed by Levison et al. (1969), a first-order low-pass filter (Eq. (5)) is used to generate the colored noise.

$$S_{nn_e}(j\omega) = |H_n(j\omega)|^2 S_{ww}(j\omega) \quad (4)$$

$$H_n(j\omega) = K_n \frac{1}{1 + T_{n,lag} j\omega} \quad (5)$$

The parameters K_n and $T_{n,lag}$ are estimated from the experiment data of Pool et al. (2016) using the methodology described in (Van der El et al., 2019). This involves fitting the model of Eq. (5) to an estimated S_{nn_e} PSD using a least squares cost function, see Fig. 6. To decrease noise in the PSD estimate, S_{nn_e} was averaged over five consecutive tracking runs. This results in K_n and $T_{n,lag}$ estimates for each participant across sets of five consecutive runs.

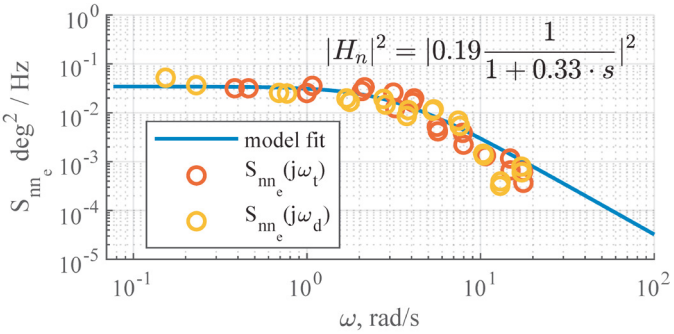

Fig. 6. Example of fitting the low-pass filter remnant model to an estimated remnant spectrum.

Model simulations To facilitate a direct comparison of classification performance on 'real' HC and simulated cybernetic HC model data, the model explained in Section 3.2 is used to generate a matched simulated counterpart of each real HC tracking run from all participants in the data set (i.e., 12×100 simulated tracking runs). This is done using the linear HC model parameters estimated for all experimental tracking runs by Pool et al. (2016) and, for simulated data with remnant, the 5-run average estimated remnant model parameters we obtained as described in Section 3.2.2. With this approach, we ensure that all linear and non-linear HC behavior variations that occur during training in the experiment data are also present in the simulated data, to the extent possible with current cybernetic methods. Furthermore, the variation in f_t and f_d signal realization (see Section 3.1) is also matched perfectly between simulated and experimental tracking runs to avoid artifacts in classifier training. Finally, for simulated data that includes remnant, a different realization of white noise w (fixed random seeds) is used to generate a unique remnant signal for every run according to Eq. (4).

While a matched simulated equivalent of all tracking runs from all participants in the experiment is generated, see also Section 4.1, for classifier training again only the first 20 and last 20 runs of each simulated HC are used to represent 'unskilled' and 'skilled' HC behavior, respectively. As an indication of HC and remnant model parameter differences between 'unskilled' and 'skilled' samples, Table 1 lists the means and standard deviations of all HC simulation parameters across all (simulated) participants. Please note that these values are provided here only for reference: for generating the simulation data the (unaveraged) values for all 12 individual participants and all runs were used.

Table 1. Mean and standard deviation of all cybernetic HC model simulation parameters for 'unskilled' and 'skilled' samples.

Symbol	Unit	Mean		Standard deviation	
		'Unskilled'	'Skilled'	'Unskilled'	'Skilled'
K_v	-	2.54	3.02	0.93	0.94
T_{lead}	s	0.58	0.44	0.22	0.11
T_{lag}	s	2.44	1.39	1.59	0.62
τ_v	s	0.28	0.21	0.15	0.05
ω_{nm}	rad/s	10.86	10.01	5.22	2.63
ζ_{nm}	-	0.51	0.27	0.42	0.13
K_n	-	**0.68**	**0.37**	0.55	0.17
$T_{n,lag}$	s	1.04	0.72	0.86	0.59

3.3 Classifier training and validation

The effectiveness of the cybernetic data augmentation method for classifier training was empirically tested. This is done by training the optimized deep learning model using an 80%/20% training/validation data split, where the 20% validation data are *always* real HC data, see Fig. 7. For training the classifier, three different training cases are considered:

- '*Real HC data*': the baseline case, where the classifier is trained on the remaining 80% real HC data.
- '*Simulated HC data* with *remnant*': where the classifier is trained on a matched 80% simulated HC data set that also includes simulated remnant.
- '*Simulated HC data* without *remnant*': where the classifier is trained on the same 80% simulated HC data, but excluding the simulated remnant.

To ensure a fair comparison, the matched simulated counterparts of the real HC data, i.e., those generated with the HC model and remnant parameters estimated from the same trials of real HC data, were used for the second and third cases. This also ensures there can never be any overlap between simulated HC training data and the 'real' HC validation data. Furthermore, to obtain a reliable indication of classification performance, the three cases were compared for 30 different randomly split 80%/20% distributions of the training and validation data. In all cases, both the training and validation data sets contained 50% 'unskilled' and 50% 'skilled' samples.

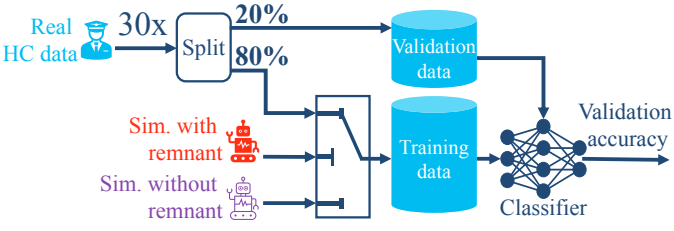

Fig. 7. Overview of classifier training and validation approach, with 30 different random 80%/20% distributions of the training/validation data.

4. RESULTS

4.1 Simulated data verification

To verify whether our simulated HC data accurately represents the measured HC data, the variance of the tracking error σ_e^2 and the control input σ_u^2 of these real and artificial

tracking runs are shown in Fig. 8. Overall, Fig. 8 shows an accurate match between simulation and experiment data, with equivalent absolute values, trends, and spread in the real and artificial data set with remnant. Furthermore, the simulated data with remnant (red markers in Fig. 8) matches the 'real' HC data (blue markers) at very high accuracy, while for the simulations without remnant (purple markers) both σ_e^2 and σ_u^2 are underestimated, as expected.

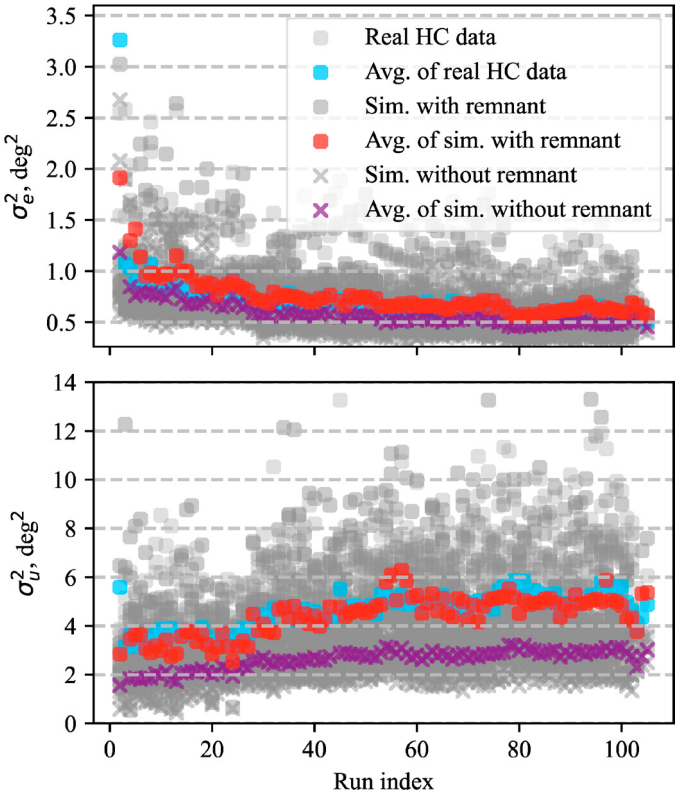

Fig. 8. Comparison of e and u variances for simulated and real HC data.

4.2 Classification performance

The results of the analysis described in Section 3.3 are shown in Fig. 9, where each marker represents one of the 30 random 80%/20% train/validation splits. The vertical axis shows the validation accuracy when training with the respective data sets and validating on real HC data. The horizontal axis indicates the bias of the trained model, expressed as the percentage of samples classified as 'skilled'. This bias should be 50% for an unbiased classifier, as we ensured that the validation data are evenly distributed between both classes.

Fig. 9 shows that when training *and* validating on real HC data (blue markers), a consistent validation accuracy between 85-90% is achieved. Furthermore, around 50% of the samples is classified as 'skilled'/'unskilled', as expected due to the ensured class-balance in the validation data. Overall, these results thus indicate that accurate and unbiased classification of HC skill level can be achieved with our approach. However, when training the same classifier on *simulated* data and validating on real HC data, a consistent drop in accuracy and increased spread is observed from Fig. 9. Training on simulated HC data

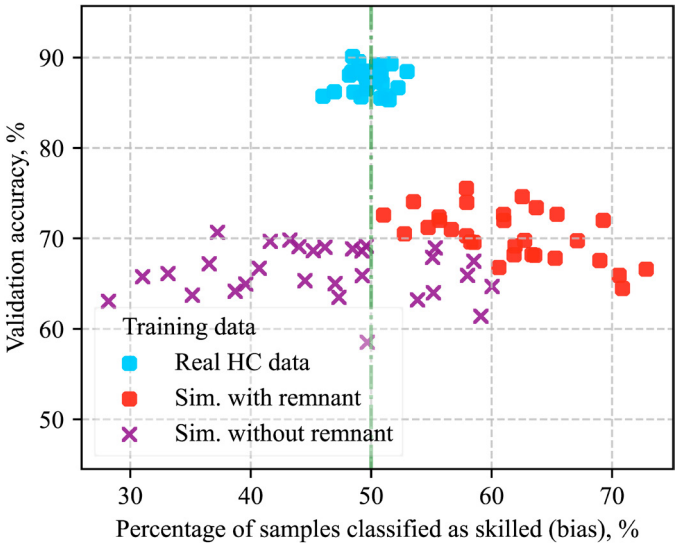

Fig. 9. Testing the effectiveness of cybernetic data augmentation by training the model on simulated data—with- and without remnant—and validating it on real HC data. Each marker represents a random 80%/20% distribution of train/validation data.

without remnant (purple markers) results in a validation accuracy of approximately 65%, while the inclusion of simulated remnant (red markers) increases this to around 70%. While this is a notable drop in accuracy compared to training on real HC data, these results do confirm that the non-linear remnant in HC data is also 'learned' by the classifier and directly influences its predictions. In addition to the reduced accuracy, Fig. 9 also shows that when training on simulated HC data the classifier's outcome is less consistent across the 30 times the classifier trained, especially in terms of the percentage of samples classified as 'skilled'. This stronger sensitivity to the (random) selection of the validation data suggests that the classifiers trained on simulation data are less robust to the inherent between-subject differences in HC data than when real HC data is used for training.

When using simulated training data *without* remnant, the classifier does not learn to incorporate remnant in its skill-level predictions. This implies that any remnant present in the real HC validation data may result in classification uncertainty, which is consistent with the lower accuracy and increased spread seen in the purple markers in Fig. 9. The wide-spread outcomes for the percentage of identified 'skilled'/'unskilled' samples seem to indicate an interaction with the inherently different overall skill levels present in experimental data sets obtained from multiple HCs; Fig. 8 shows a clear overlap in σ_e^2 and σ_u^2 at the beginning and end of training. Without remnant also 'knowing' of accompanying variations in remnant, this overlap may increase classifier outcome variability and its dependency on the selected validation samples. While Fig. 9 shows that HC simulation data without remnant does not result in an effective classifier, further investigation of the slight bias towards 'unskilled' classification may be of interest to guide cybernetic data augmentation improvements.

The bias towards 'skilled' predictions seen for the classifier trained on simulated data *with* remnant may be

explained by the method used for the remnant simulation, see Section 3.2.2. Using time-invariant colored noise as remnant (Levison et al., 1969) likely more accurately matches the remnant of 'skilled' HCs than that of 'unskilled' HCs; trained HCs show more consistent overall behavior, whereas untrained HCs will have more erratic and time-varying contributions to their remnant (McRuer and Jex, 1967). This can explain why the trained classifier can effectively identify 'skilled' HC data, but is less successful in recognizing 'unskilled' data, resulting in the observed bias towards 'skilled' predictions. This result suggests that the state-of-the-art remnant model implemented in this paper does not fully cover the complex non-linear behavior of real HCs in the time domain. More research is needed to enhance the effectiveness of this cybernetic data augmentation method through improved remnant modeling. For example, a key improvement may lie in the use of 'generative neural networks' to simulate a realistic and more non-linear remnant contribution that can be combined with classical linear cybernetic HC simulations.

5. CONCLUSIONS

This paper introduces a deep learning approach for binary classification of human controller (HC) skill level based on short (1.2 s) time signals of control behavior. Specifically, the extent to which additional, artificial, training data generated from state-of-the-art cybernetic HC model simulations can be used for classifier training is investigated. The results highlight a strong sensitivity of the classifier to the presence of remnant in the training data. While classifiers trained fully on experiment data attained an accuracy of 90%, training the same classifier on simulated HC data resulted in a 15%-25% accuracy reduction and a consistent bias towards one of the classes. Overall, these results indicate that current quasi-linear HC model simulations, and especially the state-of-the-art in remnant modeling, do not sufficiently capture the intricacies of real time-domain HC data to enable truly effective simulated data augmentation for classifier training.

REFERENCES

Cui, Z., Chen, W., and Chen, Y. (2016). Multi-scale convolutional neural networks for time series classification.

Duarte, R.F.M., Pool, D.M., van Paassen, M.M., and Mulder, M. (2017). Experimental scheduling functions for global LPV human controller modeling. *IFAC-PapersOnLine*, 50(1), 15853–15858.

Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4), 917–963.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. ArXiv: 1512.03385.

Jain, A., Singh, A., Koppula, H.S., Soh, S., and Saxena, A. (2016). Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *Int. Conf. on Robotics and Automation (ICRA)*, 3118–3125.

Le Guennec, A., Malinowski, S., and Tavenard, R. (2016). Data Augmentation for Time Series Classification using Convolutional Neural Networks. In *ECML/PKDD Workshop on Advanced Analytics, Riva Del Garda, Italy.*

Levison, W.H., Baron, S., and Kleinman, D.L. (1969). A model for human controller remnant. *IEEE Transactions on Man-Machine Systems*, 10(4), 101–108.

McRuer, D.T. and Jex, H.R. (1967). A Review of Quasi-Linear Pilot Models. *IEEE Trans. on Human Factors in Electronics*, 8(3), 231–249.

Mulder, M., Pool, D.M., Abbink, D.A., Boer, E.R., Zaal, P.M.T., Drop, F.M., van der El, K., and van Paassen, M.M. (2018). Manual control cybernetics: State-of-the-art and current trends. *IEEE Transactions on Human-Machine Systems*, 48(5), 468–485.

Nittala, S.K.R., Elkin, C.P., Kiker, J.M., Meyer, R., Curro, J., Reiter, A.K., Xu, K.S., and Devabhaktuni, V.K. (2018). Pilot skill level and workload prediction for sliding-scale autonomy. In *IEEE Int. Conf. on ML & Applications (ICMLA)*, 1166–1173.

Pool, D.M., Harder, G.A., and van Paassen, M.M. (2016). Effects of simulator motion feedback on training of skill-based control behavior. *Journal of Guidance, Control, and Dynamics*, 39(4), 889–902.

Pool, D.M. and Zaal, P.M.T. (2016). A cybernetic approach to assess the training of manual control skills. *IFAC-PapersOnLine*, 49(19), 343–348.

Pool, D.M., Zaal, P.M.T., Damveld, H., van Paassen, M.M., Vaart, J., and Mulder, M. (2011). Modeling wide-frequency-range pilot equalization for control of aircraft pitch dynamics. *Journal of Guidance, Control, and Dynamics*, 34(5), 1529–1542.

Rojer, J., Pool, D.M., van Paassen, M.M., and Mulder, M. (2019). UKF-based identification of time-varying manual control behaviour. *IFAC-PapersOnLine*, 52(19).

Saleh, K., Hossny, M., and Nahavandi, S. (2017). Driving behavior classification based on sensor data fusion using LSTM recurrent neural networks. In *IEEE 20th Int. Conf. on Intelligent Transportation Systems*, 1–6.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. ArXiv: 1409.1556.

Tango, F. and Botta, M. (2013). Real-time detection system of driver distraction using machine learning. *IEEE Trans. on Int. Transp. Systems*, 14(2), 894–905.

Van der El, K., Pool, D.M., and Mulder, M. (2019). Analysis of human remnant in pursuit and preview tracking tasks. *IFAC-PapersOnLine*, 52(19), 145–150.

Wang, Z., Yan, W., and Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. In *Int. J. Conf. on Neural Networks.*

Wijlens, R., Zaal, P.M.T., and Pool, D.M. (2020). Retention of manual control skills in multi-axis tracking tasks. In *AIAA Scitech 2020 Forum.*

Xi, P., Law, A., Goubran, R., and Shu, C. (2019). Pilot workload prediction from ECG using deep convolutional neural networks. In *IEEE Int. Symp. on Medical Measurements & Appl.*, 1–6.

Zaal, P.M.T., Pool, D.M., de Bruin, J., Mulder, M., and van Paassen, M.M. (2009). Use of pitch and heave motion cues in a pitch control task. *Journal of Guidance, Control, and Dynamics*, 32(2), 366–377.

Zaal, P.M.T. and Sweet, B. (2011). Estimation of time-varying pilot model parameters. In *AIAA Modeling and Simulation Technologies Conference.*