Deep Neural Networks for Classifying the Response of Human Controllers by Display Types[1]

Alexander Kiselev
Supervised by: Daan Pool, Max Mulder, Kasper van der El

Human adaptation remains an elusive trait for the field of cybernetics. The current modeling methods fail to fully capture the time-varying adaptable nature of human controllers. A potential solution to further investigate this unique character may be the application of deep neural networks - mathematical functions which could match the complexity of the brain. This paper aims to collect an additional data-set of human response data to control tasks utilising a single integrator controlled element dynamics. It also aims to investigate the data by classifying the display type of the task, based on 1.5-second samples, using a deep learning classifier for time series - InceptionTime. The data will be collected and verified via analysing the variances of the error and input signal. Next, the classifier will be trained on the data and the difference in performance between the single and double integrator data will be studied. Thirdly, the compensatory and pursuit displays will be grouped into one category, and classified against the preview display, leading to an impressive 99.9% accuracy. In addition, the same classifier only utilising the input signal and its derivative will show an arguably even more impressive accuracy of 89.0%. Finally, this will lead to the proposal for further investigation into the potential of applying such a classifier as a trigger for safety systems in human-controlled vehicles, and general recommendations to continue applying deep learning within the field of cybernetics.

Nomenclature

2 CC = 2 Class Classifier
3 CC = 3 Class Classifier
C = Compensatory
CED = Controlled Element Dynamics
DI = Double Integrator
H = High
HC = Human Controller
L = Low
LTI = Linear Time-Invariant
M = Medium
P = Pursuit
PR = Preview
SI = Single Integrator
VPR = Variable Preview

[1]Title generated by a neural network

I. Introduction

What sets humans apart is adaptability. Whatever the task faced, humans are able to learn and improve at, with unmatched speed and efficiency. But due to its nature, this trait remains hard to capture and quantify. Human adaptability, and in particular its time varying nature, also plays a key unknown when attempting to predict the behaviour of a Human Controller (HC) responding to a control task. This has proven to be a tremendous challenge in the field of cybernetics, where most traditional methods make use of (quasi-)linear time-invariant models to describe the responses of an HC[1], whereas HCs have been observed adopting time-varying control strategies[2] and in general adapting to the task on hand over time.

A common research technique for studying adaptation has been measuring the responses of an HC tracking a target signal, while varying the bandwidth of the signal (perceived as difficulty by the HCs), and the information presented to the HC, viz. the type of display. The display types typically include: a Control (C) display, where only the error signal is available: $e(t) = f_t(t) - x(t)$, viz. the difference between the target and output signal, a Pursuit (P) display, where both the target and the output signal are visible, and a Preview (PR) display showing the same two signals, but also displaying the future trajectory of the target for a certain time frame known as the preview time τ_p [3]. The displays are chosen to mimic the conditions that an HC may encounter when performing a control task in the real world. When comparing to driving a car the PR display could be thought of as the nominal condition where the driver is looking ahead at the road, the P display as the situation where the car enters a tight turn, and the road ahead is not visible, and the C display as the situation where the car is starting to skid and the driver is fighting to not go off the road.

A variety of such experiments has led to the creation of HC quasi-linear time-invariant (LTI) feedback models, for both the C[4], and subsequently for the P and PR[5] displays. But these models lack the ability to account for neither the inherently time-varying nature of human adaptation, nor for the observed phenomenon of HCs switching control strategies during the control task. Hence there is the need for further investigation into HC responses in these conditions.

As the current approach of LTI models appears insufficient, a promising approach is the application of machine learning to this investigation. Throughout the past decade machine learning methods utilising Deep Learning, viz. artificial neural networks consisting of many layers, have shown great promise in almost every field of research and industry. They excel at their ability to extract highly non-linear relationships from large data sets, and at finding patterns that even human experts are not able to notice, in a broad range of fields such as radiology [6], or material sciences[7]. But a fundamental requirement for applying deep learning techniques is a large reservoir of labelled data.

A number of such data-sets have been collected at the Human-Machine Interaction Laboratory in Delft, a particular data-set of note is vdEL3. A data-set of 9 subjects completing 5 runs per each of the 9 conditions [8]. The 9 conditions consisted of a Cartesian product of the 3 aforementioned display types, and 3 target signal bandwidths. The mentioned experiment was conducted using a Double Integrator (DI) Controlled Element Dynamics (CED). The extensiveness of the data-set makes it well suited for application of neural networks. In fact a previous experiment attempting to classify the type of display using this technique managed to achieve a classification accuracy of 95%[9]. It made use of a Convolutional Neural Network (CNN) architecture, known as InceptionTime [10]. In addition, a preceding experiment demonstrated a 96% accuracy at classifying the CED (Single Integrator vs DI) used by the HC [11]. It utilised a Recurrent Neural Network (RNN). These experiments have demonstrated the effectiveness of applying neural network based machine learning methods to investigating HC response, but similar success has not yet been replicated on a SI CED data-set. Although such a data-set may be a better reflection of the conditions encountered by HCs in real life situations, It is hypothesised to be more tricky to classify due to the inherent similarity between the responses to C and P displays with this CED.

To investigate the aforementioned classification of displays based on HC data in response to SI CED, an SI data-set analogous to vdEL3 must be collected. This will be performed under the supervision of D. M. Pool, the supervisor of the original experiment, in order to ensure a close match in experimental conditions.

Subsequently the quality of the data-set will be investigated via examining the variances of the error and input signals.
With the data-set collected an investigation into the HC responses may begin. The investigation will be centered around classifying the display type. In order to classify it, a CNN using the same architecture as presented by Verkerk will be developed [9], and validated by applying it to classification of the vdEL3 DI CED data-set. Next the model will be used to classify the new SI data-set in order to compare the performance of the method on SI data. This will be further validated using nine fold cross validation, and random labelling of the classification categories. The performance of the method analysed through confusion matrices, and by comparing the classification accuracy with different combinations of signals and their derivatives serving as input to the model. A further investigation will be performed into a 2 Class Classifier (CC), where the C and P displays will become one category. Finally the 2 CC and 3 CC models will be applied to classification of a number of variable preview time experimental runs, where during the run, the preview window reduces from a preset time to 0, in order to investigate the adaptation of an HC, and the ability of the model to respond to a run in mixed conditions.
This paper shall describe the methodology followed for conducting the new SI CED experiment in section II, and subsequently the methodology of conducting the experiments with the machine learning classifiers in section III. Then the results will be presented and discussed, for the Control Task experiment in section IV, and subsequently for the classification experiments in section V. Finally the paper will be concluded in section VI with the summary of the most significant findings and recommendations for further research and application of such classifiers.

II. Control Task - Methodology

The manual tracking experiment was performed in order to expand on the VdEL3 data-set created previously [8]. The VdEL3 data-set made use of 9 subjects each completing 5 experimental runs per 9 conditions [8]. Furthermore, it made use of a DI CED. In order to further investigate the behaviour of HCs in manual tracking tasks, a matching experiment was performed, replicating all the conditions of the previous experiment, but making use of a SI CED, instead. The experiment was performed under the supervision of D. M. Pool, who also supervised the aforementioned experiment, in order to match the experimental setup as close as possible.

A. Manual Tracking

During the experiment, the HCs were instructed to minimize the tracking error between the target signal $f_t(t)$, and the output signal $x(t)$ which they controlled, while the output signal was subject to a disturbance signal $f_d(t)$. Viz. $x(t) = u(t) + f_d(t)$.
The type of display available to the HCs was varied among 3 types: Compensatory (C), Pursuit (P) and Preview (PR). In a C display the HC may only see the error between the target signal and the output signal - $e(t) = f_t(t) - x(t)$. In a P display the HCs are able to separately see the target signal and the output signal, leading to HCs having more information in a P task as compared to a C task. Finally the PR display shows both the target signal and output signal separately, akin to the P display, but the future trajectory of the target signal is shown for the preview time $\tau_p = 2s$

B. Controlled Element Dynamics (CED)

Attention should also be drawn to the Controlled Element Dynamics (CED). These dynamics describe how the control input of the HC affects the output signal, viz. the controlled element. Generally two kinds of CEDs are applied in research, the Single Integrator (SI) where the HC controls the velocity of the controlled element, and the Double Integrator (DI) where the HC controls the acceleration element. Neither the Gain nor higher order integral CEDs are often applied, as neither are observed in typical applications. For the

experiment described in this paper the SI CED is used.

C. Target Signal Characteristics

The characteristics of the target signal were taken to match the signals used in the previous DI experiment [8], in order to provide for the most equal comparison between the two CEDs. They were constructed out of 10 sinusoids with amplitude and phase shift specified per sinusoid. The phase shifts were varied among 5 different realisations, being switched every run in order to avoid the HC memorising the signal. The amplitude and phase shifts including the 5 realisations were taken from the previous experiment [8].

The power and bandwidth of the signals is typically reported in order to simplify replication of experiments. In this case 3 different bandwidths were used. 1.5, 2.5 and 4.0 rad/s, as a higher bandwidth signal is more difficult for an HC to follow, the 3 bandwidths were labelled as Low (L), Medium (M) and Hard (H) respectively. The power was 1.61 cm^2 for all signals. The exact power and amplitude for the 3 signals can be seen in Fig. 1, and are labelled as "MR" followed by the bandwidth. The fourth signal labelled "VDE" may be ignored.

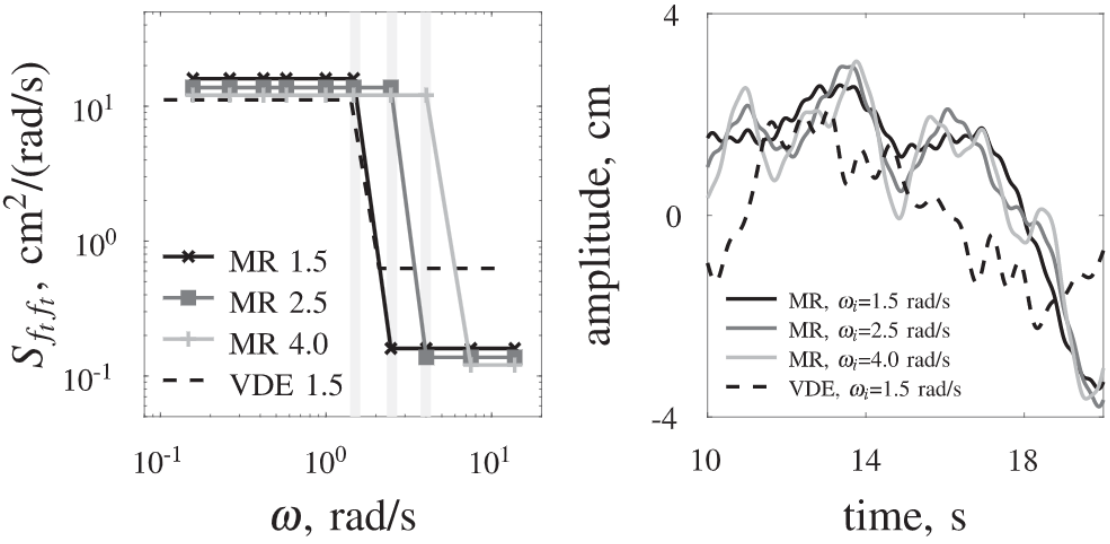


Figure 1. Target signal spectra (a) and time traces (b) [8]

D. Experimental Procedures

The experiment was conducted with 9 subjects, all TU Delft students and staff members. Each completed 5 runs per 9 experimental conditions, and 3 of the variable preview task as described in subsection II.F. The 9 experimental conditions were the Cartesian product of the of the 3 display types described insubsection II.A, and 3 levels of target signal bandwidth. The order of the conditions was randomised per subject according to a balanced Latin-square. The last 3 VPR runs were one of each bandwidth. Each run lasted 128 seconds, with only the last 120 seconds being recorded and subsequently used. The first 8 seconds was the run in time for the subject where they could tune into the control element dynamics and the task at hand.

The experiment began with exposing the subjects to the 3 different display types, while making sure they were comfortable in the simulator and had fully understood all instructions. Display types were repeated during this phase if requested by the participant. Next the measurements began. The subject had a warm-up phase of at least 2 runs after which the performance of the subject was measured via recording the RMS of the error

signal. This performance metric was reported to the subject after every run, and they were informed that a lower score is better. After 5 runs (excluding the 2 mandatory warm up ones), if the RMS of the error signal did not show a decreasing trend, the condition was concluded and the next one would begin. The subjects would be made to take a break after completing 3 conditions. The experiment was completed in half a day per subject, with each subject taking around 4 hours to complete the experiment.

E. Data Collection

The data collected during each experiment run included time traces of all the signals involved, sampled at the rate of 100 Hz. The signals recorded were: time - t, target and disturbance signals $f_t(t)$, $f_d(t)$, error signal e_t, the HC input signal $u(t)$, the output signal $x(t)$, along with the stick inputs that the HC performed and others. Only the error, input and output signals were used to generate the final data-set. In addition to this the performance metrics of each run were recorded, the performance metrics were the RMSs of the error signal and the output signal, used to judge the performance of the subjects during the experiment.

F. Variable Preview (VPR) Condition

In addition to the 9 conditions described in subsection II.D, 3 more runs were performed by 5 of the subjects at the end of the regular experiment. The runs were utilising a Variable Preview (VPR) condition. A condition that begins analogous to a PR condition but half way through the run, the preview time quickly decreases following a sigmoid curve from 2 seconds to 0 seconds at a rate of 25 seconds per second. Towards the end the run becomes analogous to a P condition. No warm up runs, and no initial training was performed for the VPR condition, in addition only 1 run was recorded per bandwidth of this condition. This was due to the fact that already the experiment took 4 hours exhausting the subjects, and this condition deviated from the scope of the original research. In addition there was an experiment planned already to further investigate this condition hence a complete and robust data-set was not needed. This condition was performed in order to have an early glimpse at the potential of applying the machine learning classifier to it.

III. Classification - Methodology

A machine learning algorithm was employed in order to predict the display type that the HC was utilizing based on a 1.5 second time trace of a variety of signals. This was a supervised learning task where the 1.5 second time traces of the experimental signals labelled with the display type were used to train a neural network[1].

A. Data Processing

The data from the experiments was converted into a data-set in the MATLAB .mat format, containing the error, input, and output signal. This was performed to ease the sharing of the data with other researchers, as several data-sets have already been shared in an identical format. Next the data had to be sampled and converted to a format which a neural network training script could accept. This was performed in several steps. First the data was extracted from the .mat file into Python, where the derivative of all the 3 signals per run was computed using `numpy.gradient`. Next the data had to be normalised per run per signal. This was performed using the `StandardScaler` from the `sklearn` Python library. It was necessary to normalise the data, as for example the error signal is almost always greater in a C or P condition than in a PR condition, and it was desirable to avoid having the classifier learn to simply consider the magnitude of the error, this normalisation method was previously shown to be robust and reliable [11]. Then the data was split into individual samples. Samples of 150 time steps, equivalent to 1.5 seconds at 100Hz sample rate of the originally record data, were taken from

[1]All the code used for data processing, creation of the model, training and evaluation can be found at: github

each run. The first sample was taken at the start of the recording, after 8 seconds of run in time, and the next sample was taken 75 time steps later, leading to a 50% overlap between samples. This was repeated until the end of each recording. Half of the data points from each sample were dropped leading to a final sample rate of 50Hz, as previous research did not demonstrate a benefit of a higher sample rate, while compute time was increased. Finally the data was labelled with its display type, receiving either a 0, 1 or 2 signifying C, P or PR displays respectively. For validation these labels were given randomly for the two experiments described in subsection V.D. It was also labelled for either the training or the testing data-set randomly with a 20% chance to be labelled testing. This whole data processing pipeline was repeated for each classifier trained. Hence each classifier had a different testing data-set, leading to more robust results after aggregating 10 trained classifiers per condition.

B. Neural Network Architecture

A neural network classifier is nothing more than a complex mathematical function mapping one set onto another, but what makes neural networks stand out is the ability to efficiently but stochastically determine an optimal mapping using the backpropagation algorithm[12]. This algorithm improves the mapping for every training example it is exposed to.

The selected neural network architecture was Inception-Time [10]. It accepts a multivariate time series, optionally applying a "bottleneck" layer where the multivariate time series are combined into one, and then applying a set of convolution layers with varying filter lengths to either the output of the bottle neck or the input time series itself leading to a new multivariate time series. This composes a single InceptionModule, a number of which are combined to make the network, culminating in 2 fully connected layers which are responsible for classifying the input, based on all the patterns detected and amplified by the earlier layers. This architecture was selected as it was the best performing architecture when attempting to perform a similar classification task but on DI CED data-set - the vdEl3[9]. A detailed discussion of the neural network architecture, and deep learning as a whole, is outside the scope of this paper, and is best left to dedicated deep learning papers. The curious reader is referred to the original Inception-Time paper for further information regarding this architecture in particular, or an introductory deep learning text book for a broader overview of the subject matter which served as a guide for the deep learning theory throughout this paper [13].

C. Software setup

The experiment was performed completely in Python due to the flexibility and the extensive collection of scientific libraries. The library used for to setup the neural network architecture and perform the training and evaluation of the model was TimeSeriesAI [14] chosen for its ease of use and due to the fact that it contained an already existing implementation of the Inception-Time architecture [15]. The library is built on top of PyTorch and FastAI libraries, aiming to expand their capabilities for use with time series data. To track the experiments and to conduct hyper parameter sweeps, the software Weights and Biases was used [16]. It should also be noted that all the models were trained for 25 epochs using `train_one_cycle` function from FastAI.

D. Hyper-parameter Sweep

The default configuration of the classifier did not lead to a satisfactory classification accuracy due to the tendency to over-fit the training data, hence leading to a worse performance on the test data - the reason for the data split. The term parameters refers to the variables that determine the output of the classifier and are updated via the backpropagation algorithm[12]. Whereas the term hyper-parameters refers to additional variables which while still determining the output of the classifier are not updated with the backpropagation algorithm, and must be set before the training begins. Yet their influence cannot be understated, but there

exists no algorithm for determining the optimal hyper-parameters. Hence the need for a hyper-parameter sweep, a process where the classifier is repeatedly trained and its performance evaluated for a variety of hyper-parameter values.

For this experiment a random sweep method was chosen, the other common method being a grid sweep which is less efficient since the model will be trained several times with the same hyper-parameter value. The default model showed a tendency to overfit the data, observed by the fact that the training loss kept decreasing to a very low value, whereas the validation loss stayed constant and even increased towards the end of the training. Here loss refers to a performance metric used to evaluate the quality of the classifier with the training loss being used as input to the backpropagation algorithm. A loss function is used instead of accuracy as the mapping of parameters to the accuracy is not a smooth function making it less suitable and efficient for backpropagation. Due to the observed overfitting the main hyper-parameters of focus were weight decay, and convolution drop out rate, the parameters typically used to regularise the model, viz. reduce overfitting. These parameters make the model be evaluated more poorly for high values of parameters in order to make the mapping function more smooth, and disable a certain percentage of connections in the convolution layers of the network in order to decrease the reliance on a specific connection respectively. Along with those two, learning rate was also varied during the sweep, this hyper-parameter, being another key one, indicates the degree to which all the parameters are changed at each step of the training loop. A too low value leads to slower training progress and the possibility that the model will not learn, where as a too high value can lead the classifier to miss the optimal point.

An initial 3 dimensional sweep space was determined, and after training a few classifiers on the DI data-set, the approximate location of the optimal hyper-parameter choices was determined. Subsequently the sweep space was reduced. This process was repeated until the classifier was producing accuracies within the range seen in a previous paper in which the same classifier architecture was trained on the same data-set[9]. After that point the choice of hyper-parameters was frozen and was not altered during any further experiments. The final choices of the hyper-parameters are summaries Table 1.

Table 1. Hyper-parameters used for the final model

Name	**Name in code**	**Value**
Batch Size	bs	64
Training Epochs	epochs	25
Bottleneck	bottleneck	No
Convolutional Dropout	conv_dropout	0.05
Kernel Size	ks	64
Number of Convolution Filters	nf	24
Maximum Learning Rate	lr_max	0.00275
Residual Connection	residual	No
Weight Decay	wd	0.05
Batch Normalization	bn	No

E. Experiments

A variety of experiments was conducted, first to validate that the classifier is up to par with classifiers demonstrated in previous research on classifying the display type of a DI data-set[9]. Subsequently new experiments on the new SI data-set were performed. The process for each experiment was consistent. Each experiment involved training 10 classifiers for 25 epochs each, with epoch here referring to showing the

training set once, fully to the classifier. The highest accuracy per classifier at any epoch was selected. These accuracies were averaged over the 10 classifiers. The accuracy was evaluated by getting the classifier to predict the categories of the testing data set at the end of each epoch. The best highest accuracy at any epoch was selected as the model would typically oscillate about the same accuracy towards the end of the 25 epochs, and due to the stochastic nature of the process sometimes the final accuracy was not the highest accuracy. The classification distribution used to construct the confusion matrices presented in subsection V.B was obtained at the last epoch of the training process, hence sometimes the accuracy there may be slightly lower than the aggregate of the best accuracies.

F. Variable Preview

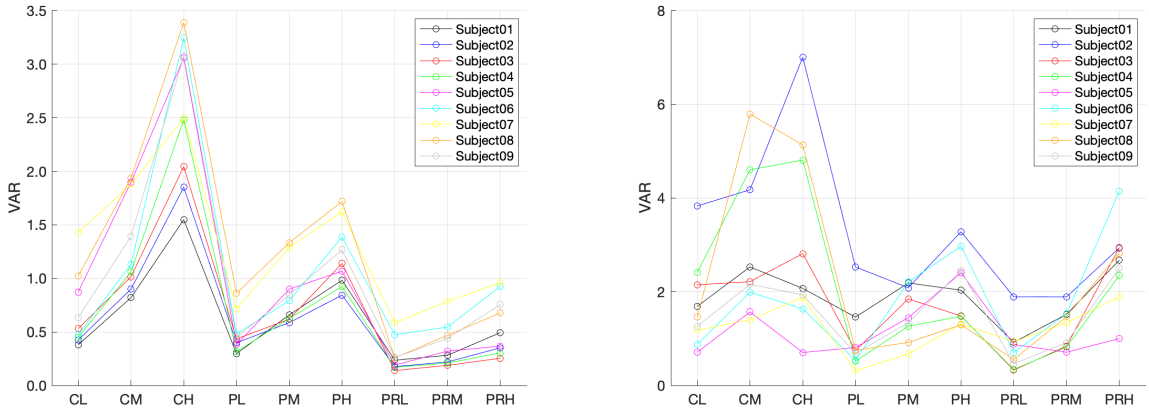
In addition to the regular experiments used to construct accuracy aggregates, an additional experiment was performed investigating the behaviour of the classifier on data with a varying preview time as described in subsection II.F. For this the data processing was changed to take samples at every time step leading to an overlap of 99.3% between consecutive samples. The overlap was increased in order to be able to obtain more samples from the data, and due to the fact that the data was not used for training, hence prior exposure to a testing run sample during training cannot occur. For analysis the distribution of the predictions of each sample was obtained per bins of 1.5 seconds each. The classifier used for this was a classifier previously trained on the SI data-set utilising all the signals and their derivatives in order to ensure the best performance. The investigation was also conducted using a 2 CC described in subsection V.B.

G. Validation

Along with performing the experiments, special attention should be paid to the validation of such a classifier, on account of a neural network acting as a black box, with its behaviour too complex to understand. This experiment was verified in 3 ways. First it was applied to the DI data-set in the same conditions as it was applied previously, in order to verify it is performing as expected. Next an experiment was conducted where all the category labels were randomised to validate that there is no mistake in the code allowing the classifier to know more than is expected. Lastly a leave one out cross validation was performed, training the classifier on 8 subjects and evaluating its performance on the remaining subject, repeated for all subjects. This was performed in order to spot potential problems with some specific subject, and to test the robustness of the classifier.

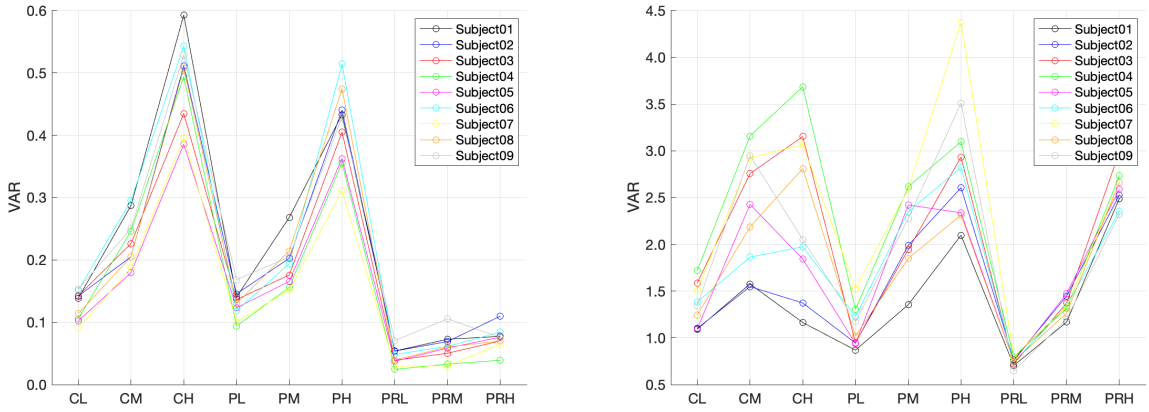
IV. Control Task - Results and Discussion

As part of the validation of the experimental data an analysis of the average variance of the error and input signals was performed. It was performed on a per subject per condition basis. The variances for the SI data set is presented in Fig. 3. The variances can be compared with the same variances computed for the DI data set presented in Fig. 2. It should be noted that the two experiments made use of different subject groups, but there was some overlap. It should also be noted that the last letter in the axis labels represents the bandwidth of the target signal, with 'L' meaning low - 1.5 rad/s, 'M' meaning medium - 2.5 rad/s, and 'H' meaning high - 4.0 rad/s.



(a) Error Signal **(b) Input Signal**

Figure 2. Variance of signals, per subject, per condition, DI CED[8]



(a) Error Signal **(b) Input Signal**

Figure 3. Variance of signal, per subject, per condition, SI CED

Several observations can be made about these graphs. Firstly, it can be noted that the average magnitude of the variances was lower significantly for the SI experiment. This is expected as the SI dynamics make for an easier task, which allows the HCs to put less effort while achieving a lower error. The variance of the input signal is taken as a proxy for effort in this case, as moving the control stick more frequently over larger displacements leads to a higher variance of the input signal.

Secondly, the trend in the error variance across conditions is the same for both the SI CED and matches the expectations. PR display leads to the smallest errors, C display leads to the highest errors, matching the reality where the HC has the most information with a PR display and the least with a C display. Increasing the bandwidth of the target signal increases the error, which again matches the expectations as a higher frequency signal requires a higher frequency input to match it, which increases the difficulty of the task leading to larger errors. But it should also be noted that the difference in the variances of the two signals between the P and C displays for the SI CED is very small, while being significant for the DI case. This is believed to be due to the differenent control approaches that must be taken for the C and P displays for the two control dynamics.

It appears that the extra information provided by the P display as compared to the C display is not too helpful when the CED is SI. Whereas for DI CED the extra information is beneficial to time the deceleration needed to avoid overshooting the target, and allows the subjects to "keep the Controlled Element symbol at the inner side of the target"[2]. Furthermore another cause for the difference in the P display between the SI and DI CEDs is the fact that some trained HCs are able to perceive the derivative of the signal shown and possibly even a double derivative. Hence when the output signal is directly shown, a trained HC is able to perceive its velocity, information much more useful when controlling with a DI CED.
On the other hand the differences in the C display between the two CEDs can also be compared, the relative increase in difficulty of the C display as compared to others is much greater for the DI CED. This is due to the fact that the HC is controlling the acceleration, and hence lacking accurate information about the position of the output, faces a significant challenge in avoiding overshoot.
Lastly, it can be observed that for some subjects, there is a decrease in the variance of the SI signal when transition from a medium bandwidth to a high bandwidth, using the C display. This can most clearly be seen in Fig. 2b for Subject 08, but can also be seen for Subject 05 in Fig. 3b. This is due to the fact that with the small amount of information available to the subject, and the highest bandwidth, some subjects will stop attempting to match all the oscillations of the target signal. They will instead attempt to follow the general trend, hence will exert less effort compared to attempting to follow all the oscillations of the lower bandwidth signal. This is a previously report observation known as *crossover regression*[8].
Overall the analysis of the variance of the error and the input signals indicate that the SI data was collected successfully and is valid to be used in the machine learning portion of the experiment.

V. Classification - Results and Discussion

In this section the results of the machine learning classification part of the experiment will be presented. They are typically presented as a box plot of the accuracies obtained after training the classifier on that specific task 10 times, and selecting the highest accuracy obtained at any epoch by each classifier.

A. Accuracies with different signal combinations

To investigate the performance of the classifier on SI data, models were trained to classify based on a variety of signal combinations and their derivatives, presented in Fig. 4. These results can then be compared with the results from the DI data in order to evaluate the classifier, plotted in the same figure. Finally the DI results can be compared with previously achieved results, presented in Fig. 5, which applied the same neural network architecture, but with a different set of hyper-parameters, to the same task on the same DI data-set in order to further validate the classifier.

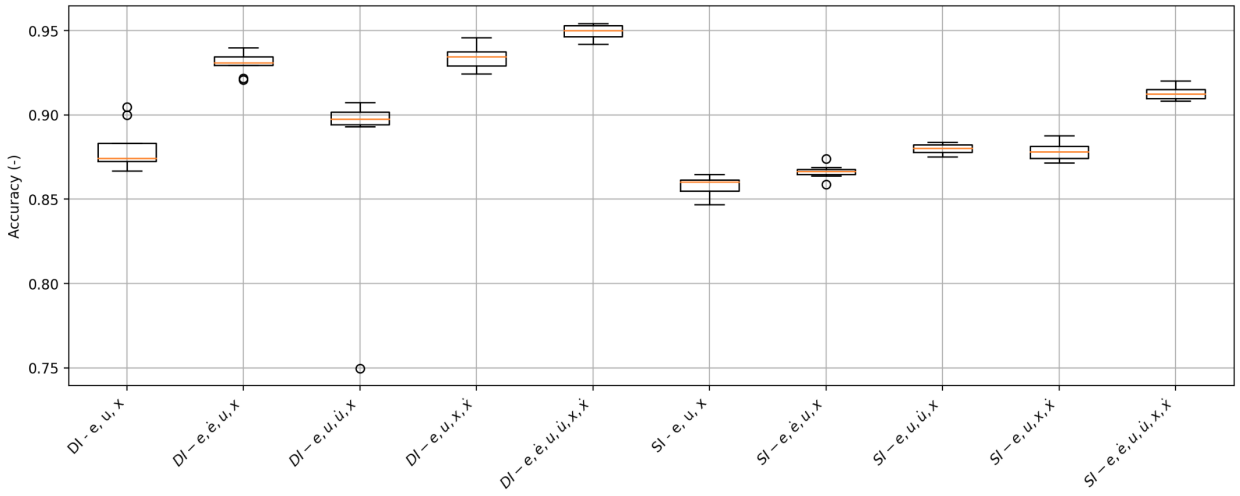


Figure 4. Accuracies of classification, varying the signals used

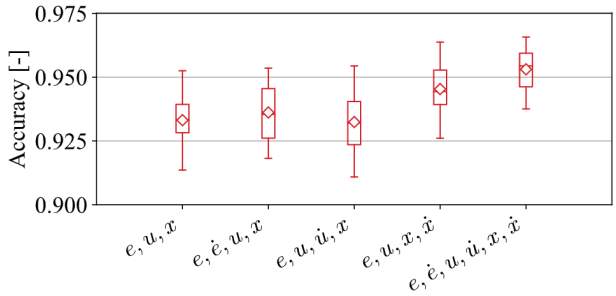


Figure 5. Accuracies of classification, varying the signals used, previous research[9]

Firstly, it should be noted that the classifier is successful at classifying the display type that the HC was responding to, achieving a mean best accuracy of 91% on the test part of the SI data-set. While this is significant, it was an expected result as similar classifiers have proven successful before achieving a 96% accuracy in classifying the CED[11] and 95% accuracy at classifying the display type with a DI CED data-set[9]. The result of the 95% accuracy on the DI CED is also replicated here.

The overall accuracies for the SI data set are lower than for the DI data-set. It is proposed that this difference is due to the fact that there is a much greater overlap between the C and P displays for the SI data-set, due to the HC not being able to benefit from knowing the absolute position of the controlled element with an SI CED as described in section IV. This hypothesis will be further investigated in subsection V.B.

It can also be seen that the variance of the accuracies per signal combination is slightly lower using the SI data-set, and that there are less outliers which are closer to the mean accuracy. This could be explained by the fact that the data for the SI data-set is less noisy than for the DI data-set. The SI task is easier for HCs to perform, which can be seen by the lower magnitude in the variance of the error signal in Fig. 3a and Fig. 2a. Hence it is hypothesised that this translates into a more consistent behaviour from the HC, especially considering that they do not have to account for decelerating the control element in order to avoid overshooting the target. As the classifier is primarily "seeing" the HC response, as the target signal is the same across both the CEDs and the key differentiator are the actions performed by the HC.

B. Confusion Matrices

To further investigate the reasons behind the difference in accuracies obtained in subsection V.A and the performance of the classifier on SI data, and the differences compared to the DI data, confusion matrices were constructed for both the DI and SI results. These confusion matrices were obtained from the 10 classifiers trained on all of the 3 signals and all of their derivatives. They are presented in Fig. 6. The numbers represent the number of samples in that category across the test sets of all the 10 classifiers. The percentages represent the distribution of the number of samples for that category across all the predicted categories.

	Total Accuracy	Predicted		
	95.42%	C	P	PR
Actual	C	99.89%	0.09%	0.02%
		42919	39	10
	P	7.88%	89.45%	2.67%
		3330	37803	1127
	PR	1.54%	2.23%	96.23%
		658	953	41169

(a) DI

	Total Accuracy	Predicted		
	91.66%	C	P	PR
Actual	C	93.17%	6.81%	0.02%
		39910	2917	10
	P	19.95%	79.99%	0.05%
		8509	34111	22
	PR	0.18%	0.21%	99.60%
		79	91	42598

(b) SI

Figure 6. Confusion Matrices

These matrices even better show significant difference between the DI and SI data-sets. For both the true predictions of the P display have the lowest accuracy, due to the fact that the P display lies in the middle of the Successive Organisation of Perception[17]. Viz. it can be seen as the mixture of both PR and C displays hence having a valid category either side, it is expected to be the most confused.

For the DI data-set the C display appears to be the easiest to classify, where as for the SI data-set the PR display is the easiest to classify. This difference is due to the nature of how the C task changes between the two dynamics. For the DI data set it is by far the hardest task which can be seen by the C task having the highest variances for the error signal in Fig. 2a, whereas for SI it is much easier as can be seen Fig. 3. This difference in relative difficulties is due to the differences in nature of the C display between the two dynamics as described previously in section IV.

The similarity between the C and P displays in the SI case leads to the HCs responding in a very similar manner which manifests as the high level of confusion between the C and P displays, significantly higher than for the DI data-set. But this similarity between the C and P responses for the SI data-set also implies that the PR responses are more distinct, which can be seen in the confusion matrix, where the classifier hardly ever misclassifies PR samples.

To further investigate this hypothesis a 2 Class Classifier (CC) was setup, where the C and P displays become one category, and PR the other. This investigation is particularly curious due to its applicability to real world situations. For example when driving a car, which is a SI CED system, it is expected that the HC is following the PR strategy, viz. looking ahead at the road. It could be assumed that when either a C or P strategy is being followed that an abnormal condition has occurred, such as skidding or being in a turn at too high of a speed. Hence a classifier which could accurately predict if a PR strategy is being followed, could be used to trigger safety systems, as it would detect when the HC is struggling and may lose control. The confusion matrices from applying such a 2 CC to both the SI and DI data-sets is presented in Fig. 7 along with applying the 2 CC on the SI data-set using only the input signal and its derivative as the input to the classifier.

	Total Accuracy	Predicted	
	97.26%	C/P	PR
Actual	C/P	98.78%	1.22%
		84366	1046
	PR	5.94%	94.06%
		2534	40145

(a) DI

	Total Accuracy	Predicted	
	99.86%	C/P	PR
Actual	C/P	99.98%	0.02%
		85312	19
	PR	0.37%	99.63%
		158	42232

(b) SI

	Total Accuracy	Predicted	
	89.03%	C/P	PR
Actual	C/P	91.62%	8.38%
		77975	7134
	PR	16.60%	83.40%
		7126	35797

(c) SI, only input signal and its derivative

Figure 7. Confusion Matrices, 2 CC

It can be seen that the 2 CC does achieve almost perfect accuracy on the SI data-set. This is an excellent indication of the applicability of the classifier for a car safety system as described earlier. Curiously the accuracy of correctly predicting a PR display is slightly diminished for the DI data-set due to more PR samples being classified as C/P, the reason for this observation should be further researched.

It is also interesting to note that the accuracy of correctly predicting the PR display is almost the same for the 2 CC as the 3 CC one for the SI data. The big improvement in overall accuracy is due to eliminating the confusion between the C and P displays by grouping them together. This further confirms the hypothesis that the difference in the accuracy levels between applying the classifier on the SI and DI data-sets is due to the similar character of the C and P displays when using an SI CED.

It should also be noted that the increase in total accuracy going from a 3 CC to a 2 CC is much greater for the SI data-set than for the DI one, 8.2% and 1.84% respectively. This is likely explained by the fact that for the SI CED the C and P displays are much more similar than the DI CED C and P displays as described in section IV. It is also likely that for the DI CED the P and PR displays cause a more similar response than SI CED P and PR displays, because for the SI CED the PR display becomes almost trivial, where the HC only needs to deal with the disturbance. Whereas with the DI CED the HC still needs to deal decelerating the control element to avoid target overshoot, an action they have to perform with the P display also, leading to more similarity.

To further investigate the applicability of the SI 2 CC to a car safety system, the same confusion matrix was constructed again, but this time the classifier only had access to the input signal and its derivative. The resulting confusion matrix is presented in Fig. 7c, and surprisingly with just that signal alone, the classifier manages to predict correctly around 89% of the time. This further increases the applicability of the SI 2 CC, as the input signal would be trivial to measure. Especially compared to the error signal which requires the target signal, or the output signal which requires measuring the position of the vehicle. It also increases the confidence in the robustness of such a classifier.

C. Variable Preview Time

To further investigate how the classifier responds to HC adaptability a variable preview time experiment was setup. It was performed in order to investigate the behaviour of the classifier in a situation where the HC changes their response strategy. The results are displayed in Fig. 8 and Fig. 9, where the breakdown of the

total number of predictions per 1.5 second intervals are grouped. The variation of the preview time during the experiment is shown by the red line, note that the preview time is normalised by dividing by the maximum preview time of 2 seconds. The greatest rate of change of the preview time was a reduction of 25 seconds per second. The experiment was performed using both the 2 CC and the 3 CC which were trained on the SI data-set using all the 3 signals and all of their derivatives.

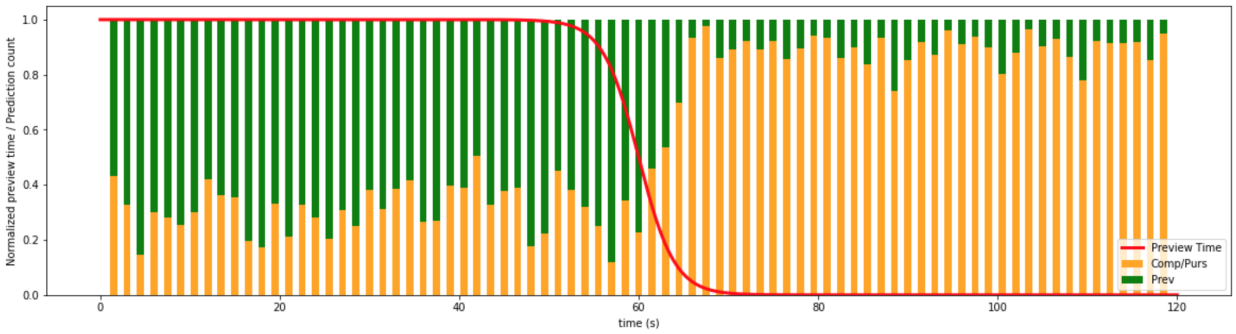


Figure 8. Classifications per time step, varying preview time, 2 CC

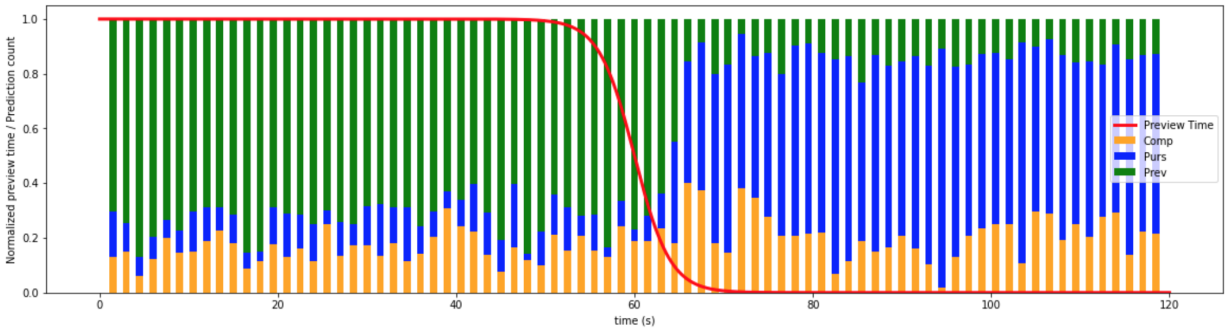


Figure 9. Classifications per time step, varying preview time, 3 CC

The first thing of note is that the classifier does indeed successfully detect a change in the SI strategy approximately from the moment that the preview time decreases to 0.4 seconds. It is curious that the classifier still mostly successfully identifies PR behaviour for a range of preview times, an indication of a robust classifier as it can deal with previously unseen data. It is of course expected that some confusion occurs at very short preview times, as the strategy being followed there is likely to be ambiguous and vary per subject. But the accuracy of the predictions is much lower than for either of the classifiers on the SI data-set, for example the accuracy of correctly predicting PR in the first third of the runs is around 70%. The accuracy of correctly predicting P or C, in the last third of the runs, for the 2 CC is better at around 90% , but still less than the accuracy achieved on the regular data-set. Although the chance of misclassifying a P as a C, in the last third of the runs, at around 20% is the same as for the regular data-set as can be seen in Fig. 6b. This decrease in accuracy is likely due to the experimental circumstances. The variable preview time runs were always performed at the end of the experiment for each subject, and only 1 run per condition was performed, not giving them the time to adjust to the new condition. It is also likely that the accuracy is worse due to the presence of outliers. Since the data-set is small it is more impacted by outliers, and in addition the runs were performed at the end of the experiments with no warm up runs, most likely causing even more outliers than usual. It is hypothesised that the accuracy would be improved significantly if this kind of condition was repeated multiple times, and training runs were provided akin to the regular data-set, but the performance

of such an experiment was outside of the scope of this research. Further research should be conducted investigating the response of the classifier to varying conditions during a run.

D. Validation

Random Labelling

It is important to validate the results obtained from a machine learning algorithm, since it is a black box approach, with the potential for many errors leading to inaccurate results.
Firstly the accuracies obtained by training the classifier on randomly labelled data for both the DI and SI data sets are presented in Fig. 10.

Figure 10. Accuracies of classification with random labels

As can be seen, the mean best accuracy was slightly above the expected 33.33% that would be obtained if the model guesses completely randomly. This is expected as it is due to the fact that, per model, the highest accuracy of any epochs is selected. Hence during the 25 epochs that each model is trained for, the accuracy oscillates around the 33.33% mark due to the stochastic nature of the training and testing process. This is further confirmed by finding the average across all the epochs of all the classifiers resulting in an accuracy of - 33.33% for SI and 33.41% for the DI. Hence the classifier is further validated.

Leave-one-out Cross Validation

The differences between HCs should also be considered, to both ensure that there were no subjects which appear anomalous but also in order to ensure that the classifier is able to handle HCs with different "characters". In order to test this a leave-one-out cross validation across both data sets was performed. This consisted of training the classifier on the data from 8 out of the 9 subjects and using the last subject as a complete validation set. The results are presented in Fig. 11. The results for the DI data-set can be compared with Fig. 12, which shows DI data-set leave-one-out cross validation results from a previous paper, in order to further validate the classifier.

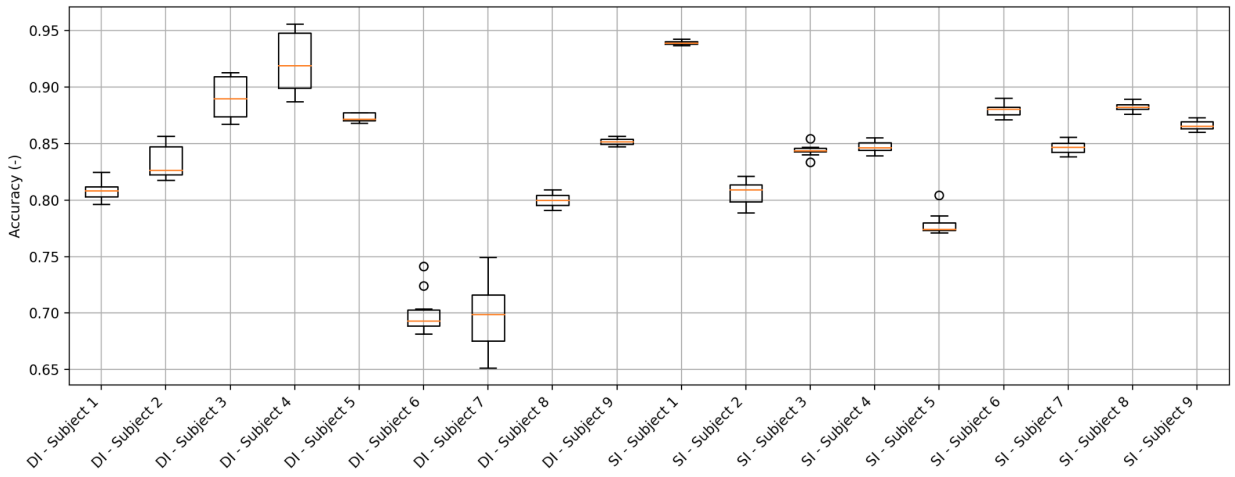


Figure 11. Leave-one-out Cross Validation Accuracies

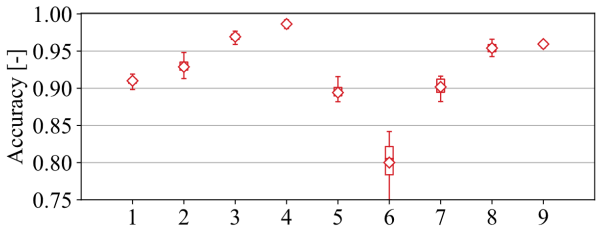


Figure 12. Leave-one-out Cross Validation Accuracies, previous research [9]

Firstly, the obtained results for the DI data-set should be compared with the results obtained in the previous experiment. It can be seen that the overall trend between subjects is replicated successfully. But there are some differences in both the absolute levels of accuracy obtained, and a difference in the accuracy trend for some subjects. Most notable Subject 7 and 8 exhibit a significantly lower accuracy in the current experiment. Both of these findings are most likely due to the differences in hyper-parameter choice for the classifiers.
Next it can also be observed that the difference between subjects was less for the SI data-set, this is likely due to the SI data-set having less noise as described in subsection V.A. The differences in accuracies per subject are likely due to some subjects performing better. It can be seen that for the DI data, the subjects (6, 7, and 8) for which the classifier obtained the lowest accuracies, also had the highest error variances as can be seen in Fig. 2a. This implies that those subjects performed more random actions than the average, making it more difficult for the classifier to classify them.
Lastly, it should also be noted that the highest accuracy across subjects was lower for the SI data, this was noted and explained in subsection V.A.
Based on these results it can be concluded that the classifier was setup successfully as it was able to replicate the results of the previous experiment, and that the classifier is not "cheating" as random labels led to an accuracy equivalent to guessing. Also that the SI data-set per subject does not display anomalous behaviour.

VI. Conclusion and Recommendations

The paper had several goals: to collect a single integrator controlled element dynamics data-set analogous the vdEl3 data-set for double integrator controlled element dynamics; to investigate the performance of a

deep learning based classifier at classifying the display type used by a human controller; to compare the performance to an equivalent classifier classifying a double integrator controlled element dynamics data-set; and to identify the potential for further research in the application of deep learning classifiers to the field of cybernetics.

The first goal was achieved successfully, the data-set was collected at the HMILab in TU Delft and is already available within the Control & Simulation group at TU Delft, and will be utilised in the near future. The quality of the data-set was ensured by following a strict experimental procedure drawn up together with Daan Pool who has previously supervised the collection of such data. Then the collected data was validated by examining the variances of the error and input signals together with Kasper van der El who collected and analysed the vdEl3 data-set among others. Subsequently a deep learning based classifier was applied to the data-set, thereby further verifying that all subjects performed appropriately. This leads to the conclusion that a high quality data-set was indeed collected.

The performance of the classifier was investigated on the newly gathered Single Integrator data-set, resulting in the best accuracy of 91.7% with 3 classes and using the 3 signals of the error, output, and input. Furthermore it was noticed that the performance of the classifier could be significantly improved by combining the Compensatory and Pursuit classes into one, while keeping the Preview separate, achieving a 99.9% accuracy. The high accuracy of the 2 class classifier, and the fact that the 3 class classifier incorrectly predicted compensatory for a pursuit sample 20.0% of the time confirms the hypothesis that there is a significant overlap in the human controller responses to the pursuit and compensatory display types when utilising a single integrator control dynamic. This kind of classifier could potentially be applied as a trigger for a vehicle safety system when a Human Controller appears to be losing control, as the ideal scenario has the Human Controller operating in preview mode exclusively. Even more interestingly, using the 2 classes described, but only inputting the input signal and its derivative still lead to a 89.0% accuracy, further demonstrating the robustness of the classifier and increasing its applicability for a triggering mechanism described above, due to the fact that the input signal is the easiest signal to measure in a vehicle. Hence it is recommended to investigate the potential of creating such a safety system further. This could be achieved in the HMI Laboratory in TU Delft, this time utilising the car driving simulator.

A limited number of runs using a Variable Preview time were also investigated in order to gauge how the classifier deals with real time changes in Human Controller behaviour. The classifier was able to correctly identify the switch in the control strategy, but with a lower accuracy, although a variety of previously unseen preview times were labelled correctly, indicating a robust classifier. The decrease in accuracy is likely due to the poor quality of the runs in this condition. Hence it is recommended to expand the collection of data available for this condition, and to investigate it further, possibly via applying a deep learning classifier to it, as this condition appears to be promising for studying real time Human adaptation more intimately.

Overall, there appears to be great potential for the application of deep learning to the field of cybernetics. For the first time there seems to be a tool as complex as the neuromuscular system, therefore it is a firm belief that this tool, one that goes far beyond the linear time-invariant models of the past, must be utilised in order to unlock the secrets of the human mind.

Acknowledgments

A. Kiselev would like to greatly thank their excellent supervisors for their extensive help and guidance with this research project, it would not have been possible without you. They would like to thank Daan Pool for their constant availability, willingness to help and chat, and the positive outlook injected into every meeting. They would also like to thank Max Mulder for willing to take a chance on them in the first place, for making them feel so welcome at the Control & Simulations, for their availability, and cheery presence. They would also like to thank Kasper van der El for joining them on this journey, and always being willing to help out with Matlab and cybernetics theory. Also they would like to thank Jos Sinke for coordinating the HBP and

creating this opportunity for the taking. Finally they would like to thank the Control & Simulations group for welcoming them in.

References

[1] 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., "Manual Control Cybernetics: State-of-the-Art and Current Trends," *IEEE Transactions on Human-Machine Systems*, Vol. 48, No. 5, 2018, pp. 468–485.

[2] Mulder, M., Pool, D. M., van der El, K., and van Paassen, M. M., "Probabilistic Perspective on Compensatory, Pursuit and Preview Manual Control," *Control and Simulation, Faculty of Aerospace Engineering, TU Delft*, ????

[3] Krendel, E. S., and McRuer, D. T., "A Servomechanisms Approach to Skill Development," *Journal of the Franklin Institute*, Vol. 269, No. 1, 1960.

[4] McRuer, D. T., and Jex, H., "A Review of Quasi-Linear Pilot Models," *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-8, 1967.

[5] van der El, K., Pool, D. M., Damveld, H. J., van Paassen, M. M., and Mulder, M., "An Empirical Human Controller Model for Preview Tracking Tasks," *IEEE Transactions on Cybernetics*, Vol. 46, 2016.

[6] Chan, S., and Siegel, E. L., "Will machine learning end the viability of radiology as a thriving medical specialty?" *The British journal of radiology*, Vol. 92, 2019.

[7] Nardi, D., and Sinke, J., "Design Analysis for Thermoforming of Thermoplastic composites: Prediction and Machine Learning-based Optimization," *Composites Part C: Open Access*, Vol. 5, 2021, p. 100126.

[8] van der El, K., Pool, D. M., van Paassen, M. M., and Mulder, M., "Effects of Target Trajectory Bandwidth on Manual Control Behavior in Pursuit and Preview Tracking," *IEEE Transactions on Human-Machine Systems*, Vol. 50, No. 1, January 2020.

[9] Verkerk, G. J. H. A., "Classifying Human Manual Control Behavior in Tracking Tasks with Various Display Types Using the InceptionTime CNN," Master's Thesis, Faculty of Aerospace Engineering, T.U. Delft, Delft, Netherlands, 2021.

[10] Fawaz, H. I., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D. F., Weber, J., Webb, G. I., Idoumghar, L., Muller, P.-A., and Petitjean, F., "InceptionTime: Finding AlexNet for Time Series Classification," *Data Mining and Knowledge Discovery*, Vol. 33, Nov 2020, pp. 1936–1962.

[11] Versteeg, R., "Classifying Human Manual Control Behaviour using LSTM Recurrent Neural Networks," Master's Thesis, Faculty of Aerospace Engineering, T.U. Delft, Delft, Netherlands, 2019.

[12] LeCun, Y., Bottou, L., Orr, G. B., and Muller, K., "Efficient BackProp," *Neural Networks: tricks of the trade*, 1998.

[13] Goodfellow, I., Bengio, Y., and Courville, A., *Deep learning*, The MIT Press, 2017.

[14] Oguiza, I., "tsai - A state-of-the-art deep learning library for time series and sequential data," Github, 2022. URL `https://github.com/timeseriesAI/tsai`.

[15] Oguiza, I., "tsai - InceptionTimePlus," Github, 2022. URL `https://timeseriesai.github.io/tsai/models.InceptionTimePlus.html`.

[16] Biewald, L., "Experiment Tracking with Weights and Biases," , 2020. URL `https://www.wandb.com/`, software available from wandb.com.

[17] Krendel, E. S., and McRuer, D. T., "A Servomechanisms Approach to Skill Development," *Journal of the Franklin Institute*, Vol. 269, No. 1, 1960, p. 19.