Master's thesis
Degree Programme in Physics

PREDICTING HAZARDOUS DRIVING BEHAVIOUR WITH QUANTUM NEURAL NETWORKS

Axi Holmström

18.11.2016

Supervisor: Prof. Edward Hæggström

Examiners: Prof. Edward Hæggström
Dr. Heikki Nieminen

UNIVERSITY OF HELSINKI
DEPARTMENT OF PHYSICS

P.O.Box 64 (Gustaf Hällströmin katu 2)
00014 University of Helsinki

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta/Osasto – Fakultet/Sektion – Faculty/Section	Laitos – Institution – Department
Matemaattis-luonnontieteellinen tiedekunta	Fysiikan laitos

Tekijä – Författare – Author

Axi Holmström

Työn nimi – Arbetets titel – Title

Predicting Hazardous Driving Behaviour with Quantum Neural Networks

Oppiaine – Läroämne – Subject

Fysiikka

Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages
Pro gradu	18.11.2016	61

Tiivistelmä – Referat – Abstract

Quantum Neural Networks (QNN) were used to predict both future steering wheel signals and upcoming lane departures for N=34 drivers undergoing 37 h of sleep deprivation. The drivers drove in a moving-base truck simulator for 55 min once every third hour, resulting in 31 200 km of highway driving, out of which 8432 km were on straights. Predicting the steering wheel signal one time step ahead, 0.1 s, was achieved with a 15-40-20-1 time-delayed feed-forward QNN with a root-mean-square error of RMSEtot=0.007 a.u. corresponding to a 0.4 % relative error. The best prediction of the number of lane departures during the subsequent 10 s was achieved using the maximum peak-to-peak amplitude of the steering wheel signal from the previous ten 1 s segments as inputs to a 10-15-5-1 time-delayed feed-forward QNN. A correct prediction was achieved in 55 % of cases and the overall sensitivity and specificity were 31 % and 80 %, respectively.

Kvantneuronätverk (QNN) användes för att förutsäga både framtida rattsignaler och filavkörningar för N=34 bilförare som genomgick 37 timmars vaka. Bilförarna körde 55 min var tredje timme i en lastbilssimulator på en rörlig plattform, vilket resulterade i 31 200 km landsvägskörning, varav 8432 km inföll på raksträckor. Ett 15-40-20-1-strukturerat tidsförskjutet, framåtkopplat QNN användes för att förutsäga rattsignalen ett tidssteg framåt, 0,1 s, vilket lyckades med ett kvadratiskt medelvärdesfel på RMSEtot=0.007 a.u., som motsvarar ett relativt fel på 0,4 %. Den bästa föutsägelsen av antalet filavkörningar under de följande 10 s uppnåddes genom att som in-signal till ett 10-15-5-1 tidsförskjutet, framåtkopplat QNN använda skillnaden mellan maximi- och minimivärdet i rattsignalen i de tio föregående 1 s segmenten. En korrekt förutsägelse uppnåddes i 55 % av fallen och den totala sensitiviteten var 31 % medan specificiteten var 80 %.

Kvanttineuroverkkoja (QNN) käytettiin ennustamaan tulevaa rattisignaalia ja tulevia kaistalta poikkeamisia 37 tuntia valvoneille N=34 kuljettajalle. Kuljettajat ajoivat liikuvapohjaisesssa rekkasimulaattorissa 55 min ajan joka kolmas tunti, eli kokonaisuudessaan 31 200 km maantieajoa, joista 8432 km olivat suorilla. Rattisignaalin ennustaminen yhden aika-askeleen eteenpäin, 0,1 s, suoritettin aikaviivästetyllä eteenpäinkytkeyllä QNN:llä, jolla oli 15-40-20-1 rakenne. Neliöllinen keskiarvollinen virhe oli RMSEtot=0.007 a.u., mikä vastaa 0,4 % suhteellista virhettä. Paras ennustus kaistalta poikkeamisten määrälle tulevan 10 s aikana saavutettiin käyttämällä sisäänmenona rattisignaalin suurinta huipusta huippuun amplitudia kymmenen edellisten 1 s pätkien ajalta ja aikaviivästettyä eteenpäinkytkettyä 10-15-5-1 QNN:ää. Oikeaa ennustusta saavutettiin 55 % tapauksista ja sensitiviteetti oli 31 % ja spesifisiteetti oli 80 %.

Avainsanat – Nyckelord – Keywords

Quantum Neural Networks, Qubit neuron, time series prediction, drowsy driving

Säilytyspaikka – Förvaringställe – Where deposited

E-thesis Helsingin yliopisto

Muita tietoja – Övriga uppgifter – Additional information

Till pappa, som lärde mig att älska kunskap.

Table of Contents

Predicting Hazardous Driving Behaviour with Quantum Neural Networks

Abstract

Quantum Neural Networks (QNN) were used to predict both future steering wheel signals and upcoming lane departures for *N*=34 drivers undergoing 37 h of sleep deprivation. The drivers drove in a moving-base truck simulator for 55 min once every third hour, resulting in 31 200 km of highway driving, out of which 8432 km were on straights. Predicting the steering wheel signal one time step ahead, 0.1 s, was achieved with a 15-40-20-1 time-delayed feed-forward QNN with a root-mean-square error of $RMSE_{tot}$=0.007 a.u. corresponding to a 0.4 % relative error. The best prediction of the number of lane departures during the subsequent 10 s was achieved using the maximum peak-to-peak amplitude of the steering wheel signal from the previous ten 1 s segments as inputs to a 10-15-5-1 time-delayed feed-forward QNN. A correct prediction was achieved in 55 % of cases and the overall sensitivity and specificity were 31 % and 80 %, respectively.

1 Introduction

It is a natural instinct of people to want to prepare for the future, and at its finest, to be able to avoid hazardous situations completely by becoming aware of them beforehand. Throughout history people have been trying to *predict* the future, at times by the most extraordinary means with little chance for success, but the desire to search for a successful prediction method has survived to this day. Weather forecasting is an everyday, and modern, example of scientific prediction, which has been achieved through the increase in computational power. In this thesis the term *prediction* will be used to describe exactly this type of prediction of future events. The term is also used in scientific literature to describe *estimation*, e.g. extrapolation of some variable's value when data is not available for the entire range of interest. Prediction and estimation are very much alike, the only difference is that prediction is temporal while estimation is not, i.e. prediction aims to determine something at a future time while estimation aims to determine something in e.g. a different place, temperature range, electric field etc. This distinction is only made here to clarify that this thesis concerns prediction of events at future points in time.

The enormous increase in computational power that has occurred during the last few decades has enabled modelling of increasingly complex systems and even developing artificial intelligence and machine learning to harness computers' superior ability to perform mathematical calculations. Machine learning is the field of study of computers' ability to learn to perform tasks without being specifically programmed, in practice, machine learning are algorithms that perform user-independent optimisation of their own structure according to some learning rules. The field of machine learning is already very large and it is increasing all the time. It contains several different subfields, e.g. Decision Trees [1, 2], Support Vector Machines [2, 3], Bayesian Networks [2, 4], Genetic Algorithms [5, 6], and Neural Networks [7, 8] (the references are examples of comprehensive book chapters on the listed subfields or recent review papers). As the title of this thesis suggests, Quantum Neural Networks (QNNs), a subcategory of neural networks, were used in this study. The reason for choosing neural networks is that they have performed well with time series prediction [9], i.e. predicting future time steps of a signal. The other listed subfields of machine learning are more suitable for classification tasks, i.e. they are developed to divide data into separate classes. As time series prediction is a modification of pattern recognition, which is a type of classification, many machine learning algorithms optimised for classification have been used successfully to predict time series, but several techniques (especially when applied to time series) require *a priori* knowledge of e.g. the underlying model or possible states, which is a limitation the neural networks do not have. Other prediction methods not utilising machine learning do exist, examples include autoregressive models (AR) [10, 11], moving average models (MA) [11, 12], combinations and evolutions of the aforementioned (e.g. ARMA [11, 13], ARIMA [11, 13], ARMAX [12, 13], GARCH [14-16]), Kalman filtering [17, 18], non-parametric regression [19-22], Markov chains [23, 24] etc., but these are also model-based and require even more care from the user than the ones based on machine learning, therefore these were not selected for this work.

Having briefly discussed the general background of machine learning, let us now turn our attention to neural networks and further to Quantum Neural Networks (QNNs). Biological neural networks in e.g. the brain have certain fascinating and useful characteristics: They are linked together in entities performing specific, even very complex, functions, even though a single neuron is a fairly simple biomechanical device with a simple function, and, perhaps most interestingly of all, they learn to optimise their function from previous experiences and subsequent outcomes. These are the traits that any computational Neural Network aims to simulate — they are constructed from simple components, they are shown data and a learning algorithm trains them to perform some task, which they can learn without a person determining the parameters and without anyone knowing exactly how the trained network performs its function. This autonomous learning is a major advantage because it enables the network to perform tasks that are too complex to define as a function or model, and it does this without requiring a user to be able to define how it should be done.

Artificial Neural Networks (ANNs) have a long history starting in 1943 when McCulloch and Pitts [25] described a logic network based on neuronal activities. Following Hebb's postulation that the connection between biological neurons that fire together is strengthened [26], Rosenblatt [27] developed the perceptron in 1958, an artificial neuron which is still the basic building block in many ANNs. Since then, several tens, if not hundreds, of network structures have been developed, each suitable for a slightly different task. For instance, some are suitable for time series prediction (e.g. time-delayed feed-forward NNs [28], radial basis function networks [29], recurrent NNs [30]), other for pattern recognition (e.g. recurrent NNs [31], Hierarchal Graph NNs [32]), regression (e.g. General Regression NNs [33]), and image processing (review article [34], mentions e.g. feed-forward, Kohonen, and Hopfield networks), etc., just to name a few. As the theories about quantum computers were emerging separately, the benefits of quantum computation algorithms, e.g. parallelism from superposition, were introduced to neural networks (see [35] for a comprehensive book chapter on the subject). The approach used in this thesis is the implementation of the qubit neuron [36], which exists in a superposition of states, thereby enabling parallelism and interference of states, and then organising them into a suitable network structure (2.1 Theory of Quantum Neural Networks). The benefits of QNNs compared to classical ANNs are their higher learning efficiency [36], i.e. learning requires fewer iterations than in similar classical ANNs to achieve the same precision, their ability to handle nonlinear signals and tasks [37], and their higher memory capacity [38].

Knowing that QNNs have been able to predict nonlinear signals, it was thought that they could be used to predict lane departures from the steering wheel signals of tired drivers. Rather unfortunately, almost every driver has some experience with feeling tired and unfocused when sitting behind the wheel. To provide an estimate of the prevalence and seriousness of the problem, a survey conducted on Finnish professional truck drivers reported that over 20 % of the participants (*N*=184) had nodded off at the wheel at least twice during the surveying period of only three months [39]. The real danger with driving while suffering from impaired alertness arises if the car leaves the lane. Therefore a warning system predicting lane departures would increase safety both for the sleepy driver and other drivers nearby. Research has been made into predicting the sleepiness of drivers ([40] represents the state-of-the art), but such attempts involve biometric signals recorded from the driver, which are cumbersome to collect and process. Car manufacturers have also developed camera-based systems that detect when the car is about to leave the lane, but they work poorly in difficult light and weather conditions when lane markers are not clearly visible [41]. Successfully predicting lane departures directly from the steering wheel signal would eliminate the need for biometric or camera-based systems.

To achieve steering-based lane departure prediction is, however, not an easy task. Steering wheel signals are erratic, nonlinear, and include transients (i.e. sudden large changes in the signal compared to the average signal behaviour), which makes them hard to model properly. In addition, the more tired a driver is, the more erratic the steering behaviour becomes. But since QNNs are known to be able to handle such signals and have been used to successfully predict very nonlinear and transient signals (e.g. sunspot activity [37], commodity prices [42], and short-term loads of power systems [43]) it was thought that they could provide a tool to achieve the lane departure prediction.

Many benefits of QNNs, and neural networks in general, have been presented in this introduction, with their main advantage being their ability to train themselves to perform a task. This enables prediction without having to construct complicated underlying models or having *a priori* knowledge, but all this user-independence has a price that strongly influences this entire thesis: A neural network is essentially a "black box", which does impose the restriction that a user can never determine exactly how any one parameter of the network influences its performance. In practice this means that any speculations into the inner workings of a neural network are just that, speculations — it is impossible to determine with absolute certainty why a QNN succeeds or fails. There are, however, certain general conclusions that have been reached during the years of neural network research which do provide the user with tools to make reasonable assumptions about the operation of the network. Along with the methods used and achieved results for Quantum Neural Network prediction, these reasonable assumptions, along with a motivation for them, are presented and discussed in this thesis.

2 Methods

The methods of this project encompass several separate parts, which motivates the division of this section. To alleviate the possible search for something specific in this section, a short description of the different subsections is now given. First, the theory of Quantum Neural Networks (QNNs) is presented (2.1 Theory of Quantum Neural Networks), with emphasis on the theory of the qubit neuron, of which different network structures can be assembled. The time-delayed feed-forward neural network, which was used in this work, is presented, but other network structures are not discussed in this section. Learning of the QNN is also presented. The data set is part of an extensive sleep deprivation study [44], but an adequate general description of the study and a detailed description of the collected driving data will be given in Data Set (2.2). In this section the relevant terms and theories related to sleep deprivation is also presented to the extent that is necessary for the reader to fully grasp the effects of the sleep deprivation on the QNN predictions. The last section, Prediction and Analyses (2.3), is further divided into Predicting Steering (2.3.1) and Predicting Lane Departures (2.3.2), because while both types of predictions have the same general steps (presented in 2.1 Theory of Quantum Neural Networks), the execution of steering wheel signal prediction and lane departure prediction are surprisingly different. Furthermore, due to the iterative nature of the work, where each attempt led to some assumed improvements in the next attempt, the main conclusions from each attempt will be discussed briefly in order to facilitate an easier read while the full results are presented in Results. All computer algorithms for the QNNs were written and implemented by the author in Matlab 2013b.

2.1 Theory of Quantum Neural Networks

Artificial Neural Networks (ANN), which is the name for neural networks performing classical computations, differ from their quantum counterpart, Quantum Neural Networks (QNN), only in the way a single neuron operates. These qubit neurons [36] can be assembled into any network structure in the same way as perceptrons [27], the neurons in ANNs, are assembled into ANNs. The advantage of the qubit neuron compared to the perceptron is that it is programmed to exhibit quantum effects, like superposition and interference of states. This parallelism should give it similar advantages over classical perceptrons that quantum computers will have over classical computers. There are several publications about QNNs and qubit neurons (see e.g. [35]), but the theory presented here is based on [36], which presents the theory and equations in a very detailed and understandable manner.

A qubit neuron resembles biological neurons in the sense that they have multiple inputs, each of which is weighted, similarly to biological neurons reacting more strongly to inputs from certain neighbouring neurons [26], and they have one output. All of this can also be said for perceptrons, but there is one important difference regarding the output — a biological neuron, as a perceptron, either fires or does not (the perceptron outputs a 0 or a 1) depending on the combined input, but a qubit neuron exists in a state that is a superposition of the weighted inputs and therefore gives out a number that can be between 0 and 1, which actually is the probability of the qubit neuron's output to be a 1. To add some mathematical rigour to this statement, the difference between a computational bit and a qubit should be explained. A bit is a real value that is either a 0 or a 1, not anything in between. A qubit, on the other hand can also take on any value in between, because the state of a qubit is described as a superposition of the states $|0\rangle$ and $|1\rangle$, according to:

$$|\varphi\rangle = a|0\rangle + b|1\rangle \tag{1}$$

where *a* and *b* are complex values called probability amplitudes describing the probability of the qubit to be in the corresponding state. So a qubit can exist in a state that is any combination of the states $|0\rangle$ and $|1\rangle$, as long as the probability of it being in some state (any state) is 1. Mathematically, this is requirement is equal to:

$$|a|^2 + |b|^2 = 1 \tag{2}$$

Due to the qubit neuron being modelled like a qubit, it is not a binary classifier, unlike the perceptron and essentially also the biological neuron, which gives it its name: qubit neuron.

In the same way that biological neurons can be assembled into networks that perform very complex functions, so too can perceptrons be assembled into ANNs and qubit neurons into QNNs. Both ANNs and QNNs are constructed to have in input layer to which information is fed, one or several hidden layers, and an output layer that delivers the output from the network. The parameters that govern the interaction between the neurons and the layers are determined by learning, so the network essentially trains itself to perform a function, much in the same way as our neurological networks learn to perform functions from experience. But let us first focus on the operation of a single qubit neuron, and after that return to the operation of the entire network and the learning. At this point it is sufficient to understand that the network is constructed with one input layer, one or more hidden layers and an output layer, and all parameters for each neuron in every layer are determined through learning.

To implement the superposition of states in a qubit neuron, all real inputs and parameters of the qubit neuron is mapped onto the complex plane as phase angles using the following mapping function:

$$f(x) = e^{ix} \tag{3}$$

where x is the real value (input or parameter) that is transformed into a phase angle describing a possible state of the neuron. The interference and superposition of states is then performed as phase rotations in the complex plane. The following relation also holds for this representation of complex numbers:

$$f(x_1 + x_2) = f(x_1) \cdot f(x_2) \quad (4)$$

which is useful to bear in mind, as any multiplication of two such mappings is the same as adding the arguments (angles) together, i.e. performing a rotation in the complex plane. The qubit neurons in the input layer of any QNN only performs this mapping of the real inputs, which are restricted to the interval [0, 1], to phase angles between [0, $\pi/2$]:

$$z_{input} = f\left(\frac{\pi}{2} \cdot input\right) \quad (5)$$

where z_{input} is the output of the input layer neuron that is then fed to the neurons in the hidden layer. The neurons in the hidden layers and the output layer perform a more complicated function. A schematic description of a qubit neuron is shown in Fig. 1. As these neurons are preceded by other qubit neurons (either in the input layer or the hidden layers), the inputs are now complex numbers $z_{input,k}$, where the index k simply denotes the designation of the neuron in the previous layer from which the input is received. Each input has a weight, θ_k, that is a real valued phase angle determining how much relevance the neuron assigns to the input from that particular connection (to neuron k in the previous layer). The biological analogy is that a neuron can learn to be more sensitive to inputs from one specific connection, or in the case of perceptrons, the weight from one input is higher, giving that input a larger influence in determining the final state of the perceptron.

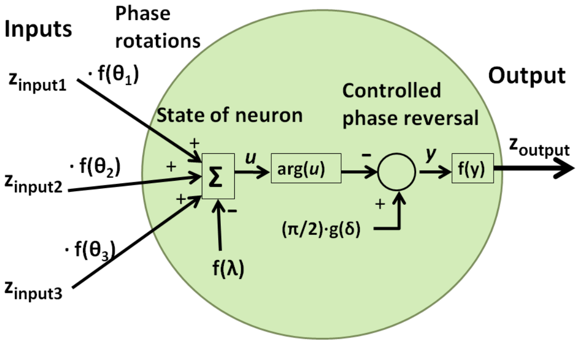

Fig. 1. Schematic of the operation of a qubit neuron in a hidden or an output layer. The complex inputs from the previous layer are rotated by the weights θ_k, the state of the neuron is determined as a superposition of states from the inputs, a controlled phase reversal is performed, and the complex output is fed to the next layer.

The weights in the qubit neuron, θ_k, assign the inputs into states through a phase rotation, i.e. the complex inputs $z_{input,k}$ are each multiplied by their corresponding complex numbers $f(\theta_k)$. The state of the neuron, u, is determined as the weighted sum of the input states minus a threshold:

$$u = \sum_{k}^{K} z_{input,k} \cdot f(\theta_k) - f(\lambda) \tag{6}$$

The sum over k is the sum of all weighted inputs (K is the number of inputs to the neuron) and λ is the threshold (also mapped onto the complex plane). The threshold serves to set the operating level of the neuron, which means that it functions as an offset or bias, around which the state of the neuron then fluctuates according to the inputs and weights. The value of the threshold is determined during learning. Having determined the state u of the neuron, it undergoes a controlled phase reversal, which is a generalised version of the controlled NOT gate used in quantum computing:

$$y = \frac{\pi}{2} \cdot g(\delta) - \arg(u) \tag{7}$$

$$g(\delta) = \frac{1}{1+e^{\delta}} \tag{8}$$

where arg() takes the argument of u, i.e. determines the real-valued phase angle of u, $g()$ is the sigmoid function (8) producing a value in the range [0, 1], and δ is the reversal parameter, which is determined during learning. The function of the controlled phase reversal is such that if $g(\delta)$ is close to 0, then sign of the phase angle of u is flipped but the magnitude remains unchanged, and because the observation probability is the square of the probability amplitude a change in sign won't affect the output. If $g(\delta)$ is close to 1, the reversal essentially swaps the probability amplitudes between the states $|0\rangle$ and $|1\rangle$.

The controlled phase reversal produces a real-valued phase angle, y, and the last operation in the neuron is to map the state back to the complex plane, producing the output z_{hidden}:

$$z_{hidden} = f(y) \tag{9}$$

The subscript "hidden" refers to the output from a neuron in a hidden layer, so z_{hidden} is then fed forward to the next hidden layer or to the output layer. The neurons in the output layer perform the same operations as those in the hidden layers (eq. 6-9), but the output from the output layer is transformed to a real value between 0 and 1 describing the probability of observing a 1 through:

$$output = \left|\mathrm{Im}(z_{output})\right|^2 \qquad (10)$$

where Im() is the imaginary part of z_{output}. In conclusion, the adjustable parameters of one qubit neuron are the weights θ_k, the threshold λ, and the reversal parameter δ. During the learning of the network all these parameters need to be determined for all neurons in the entire network.

From these qubit neurons the actual QNN can be assembled. The selected structure for this thesis was a time-delayed feed-forward neural network (Fig. 2) (see e.g. [28]) because this particular structure is well suited for e.g. time series prediction [28]. A feed-forward network is a network in which all connections are in the direction from the input layer towards the output layer, so it is the opposite of a feedback network. A feed-forward network is a simple network structure, and as such is not as computationally heavy as many other networks, resulting in shorter learning times. With the added complexity from the qubit neurons, a feed-forward network was thought to suffice and to keep learning times reasonable. The "time delay" in the name only means that the predicted output is based on inputs from the current and previous time steps, and it is implemented such that each input neuron feeds one time step (the signal at t, t-1, t-2, etc.) to the network. Therefore the number of previous time steps to feed to the network is only limited by the number of input neurons. This structure requires that the input signal has an even sampling frequency, because the only concept of time in the system is that the network will assign relevance to the different time steps, and quite possibly the change in the signal between time steps (quite possibly, because the inner workings of each trained network is different and so it is impossible to make such a statement with absolute certainty). As time progresses, the inputs move from one input neuron to the next and if the time steps have changed, the signal will appear to have a different shape than it really has due to the uneven sampling frequency.

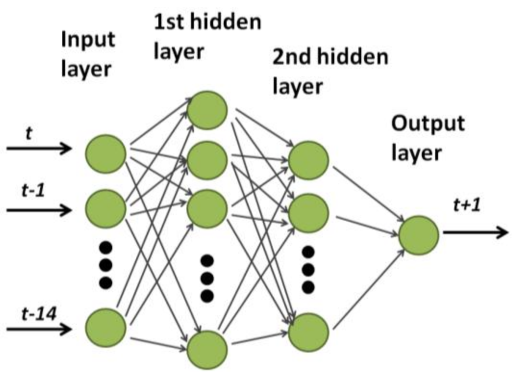

Fig. 2. Structure of a time-delayed feed-forward neural network with four layers. The fifteen neurons in the input layer are each assigned to one time step of an input signal starting from the current time step t and continuing through all previous time step to t -14. The output is the predicted signal at $t+1$.

The final phase before using a QNN for prediction (or classification or any other type of task) is the learning. There are three major learning schemes for neural networks, all of which have some kind of cost or error function to be minimised: supervised learning, unsupervised learning, and reinforcement learning. Even though supervised learning was used, a short description of each scheme is given in order to justify the choice of supervised learning over the other types. Supervised learning means that, during learning, input data is given to the network and the output is compared to the desired or known output. The cost function to be minimised is some function describing the distance between the output and the desired output, which is why supervised learning is very useful for e.g. prediction or classification. In unsupervised learning there is no desired output, only a cost function that is designed to produce a desired outcome. The cost function in unsupervised learning is therefore dependent on the task, or model, and requires good *a priori* assumptions of what the network should achieve, to learn optimally. Therefore it is best suited for estimation problems such as filtering, clustering etc. Reinforcement learning is really a Markov chain where some starting input is given to the network, which then generates an action, i.e. output, which causes a reaction from an environment, and a cost. The reaction is then fed back to the network as an input and the same procedure is repeated. To perform reinforcement learning one must have a given set of actions, reactions, and costs. Therefore reinforcement learning is suitable for decision making and control problems, but do require extensive assumptions about the dynamics of the entire system.

The choice of data on which the network is trained does, naturally, influence the performance of the network. In any machine learning, a training set is chosen for learning and a test set, which should be different from the training set, is then used for testing and evaluation of the learning. Validation sets are can also be used in machine learning to e.g. select one of several trained algorithms for the test set or tuning parameters. A validation set is not often used for neural networks because it is impossible to know the effect of changing any particular parameter of the network after training. If overfitting has occurred during learning, the results from the testing will be poor. If, however, the training set has been chosen well as a versatile representation of the data, then the learning should not cause overfitting and any failures should be caused by other factors, such as unsuitable network structure, detection of traits other than desired or learning to perform a different function than expected.

As previously mentioned, supervised learning was chosen for this work, and while there are several learning algorithms that perform supervised learning, the backpropagation with gradient descent was used. It is a very common and fairly simple learning algorithm, but it was chosen because the equations are more easily transferrable to qubit neurons than most other algorithms. In backpropagation with gradient descent the adjustable parameters are updated according to the gradient of the error function, with respect to each parameter, in an attempt to find the minimum of the error function. The term backpropagation reflects the fact that the gradients for the neurons in the input layer are calculated from the gradients in the next layer, and so forth, until the output layer is reached. In practice this mean that first the gradients are calculated for the neurons in the output layer, and the gradients in the previous hidden layer depend on those gradients, and these will, in turn, influence the gradients in the layer before, and so on until the input layer is reached.

In the first iteration of backpropagation, random values are assigned to all adjustable parameters (weights θ_k, thresholds λ, and reversal parameters δ for all neurons in the network), input data is fed to the QNN and an output signal is produced. The error function E is:

$$E = \frac{1}{2}\sum_t^T (desired\ output - output)^2 \qquad (11)$$

The squared distance between the produced output and desired output for each time step t is summed together over all time steps. The output is the produced predicted signal (T time steps long) and the desired output is the signal in the training set that should have been predicted. The adjustable parameters are updated according to:

$$\theta_k^{new} = \theta_k^{old} - \eta \frac{\partial E}{\partial \theta_k}$$
$$\lambda^{new} = \lambda^{old} - \eta \frac{\partial E}{\partial \lambda}$$
$$\delta^{new} = \delta^{old} - \eta \frac{\partial E}{\partial \lambda} \qquad (12)$$

where E is the error function, the adjustable parameters are the weights θ_k, thresholds λ, and reversal parameters δ with the superscript "old" referring to the current values and "new" referring to the updated values (for a derivation of the gradients, see [45]). η is the learning rate, a value that determines the influence of the gradient, with typical values ranging from 0.1 to 0.8. A low learning rate will allow the parameters to steadily approach the nearest minimum of the error function, but it also prevents large exploration of the parameter space. This results in a faster convergence to the minimum, but because the variations are so limited, there is a risk that the minimum is only a local minimum and not a global one. A high learning rate will cause the network to fluctuate more before reaching a minimum, and there is no guarantee that it ever will, but if a minimum is found it is less likely to be a local one because more of the parameter space has been explored. After updating the parameters, a new iteration is begun, the parameters are updated, and this is repeated until a predetermined target error is or a maximum number of iterations is reached. The trained QNN can then be used for prediction

2.2 Data Set

The data set used in this work was collected as part of a large sleep deprivation study [44] conducted at Työtehoseura in Vantaa, Finland. During the sleep deprivation the participants stayed awake for 37 hours, starting from 6:00 in the morning, and they drove in two moving-base high-fidelity driving simulators for 55 min every three hours, resulting in twelve driving bouts at different time awake (as the term describes, the time that a person has been awake). Thirty-four driver students in the age range 18-55 years old participated in the study (N=34), and only four participants did not complete the entire 37 h sleep deprivation, with the shortest time awake being 21 h. While sleepiness research is an entire field of study, the most relevant knowledge about sleepiness for the purpose of this thesis is the following: The sleepiness of a person is governed by the homeostatic and circadian sleep regulating processes [46]. The homeostatic process attempts to maintain the performance level in the long run, so when a person wakes up in the morning, the sleep pressure (desire to sleep) increases exponentially throughout the day with increasing time awake. This pressure is relieved when the person sleeps. The circadian process, however, is a function of time of day and likely stems from an evolutionary desire to be awake and active during light hours of the day and to sleep during the dark night. The circadian rhythm is the reason for starting to feel more alert in the morning even if one has stayed up all night. The relevance of this, for the work in this thesis, is that during the 37 h of sustained wakefulness the participants grew increasingly tired, peaking at approximately 25 h time awake (at 7:00 the next morning), after which they felt more alert and also drove somewhat better than during the night. Due to research aims other than predicting lane departures using QNNs being included in the study, the first sixteen drivers (IDs 1-16) drove in day-time light conditions while the other 18 drove in night-time light conditions, causing the night-condition drivers to be much less alert during the entire study (as measured by the Psychomotor Vigilance Task [47]).

The two moving-based driving simulators at Työtehoseura were highly realistic; one of the simulators was the driver's compartment of a truck and the other was a half of a bus. Even though the chasses were different, the driving scenario and the programmed truck was the same in both simulators. The driving scenario was projected on the windscreens really emerging the driver into the scenario. The 110 km long driving track was designed to resemble normal, uneventful, Finnish rural roads with some slight turns and hills and no oncoming traffic. A map of the track along with the height differences are shown in Fig. 3. Along the track were 21 straight road segments longer than 400 m, which were defined as straights. Twelve different starting positions were selected along the track and every driving scenario started at a different randomly selected starting position, with a randomly selected direction around the track, to ensure that the drivers did not learn the road by heart. The drivers were instructed to keep both hands on the wheel, stay within their own lane, and maintain a speed of 80 km/h during their entire drive. Staying within the lane was obviously important to not cause additional lane departures and maintaining the hand position was an attempt to eliminate unnecessary differences between the steering wheel signals from different individuals. Both of the instructions were given to increase the chance of success for the QNN to predict lane departures.

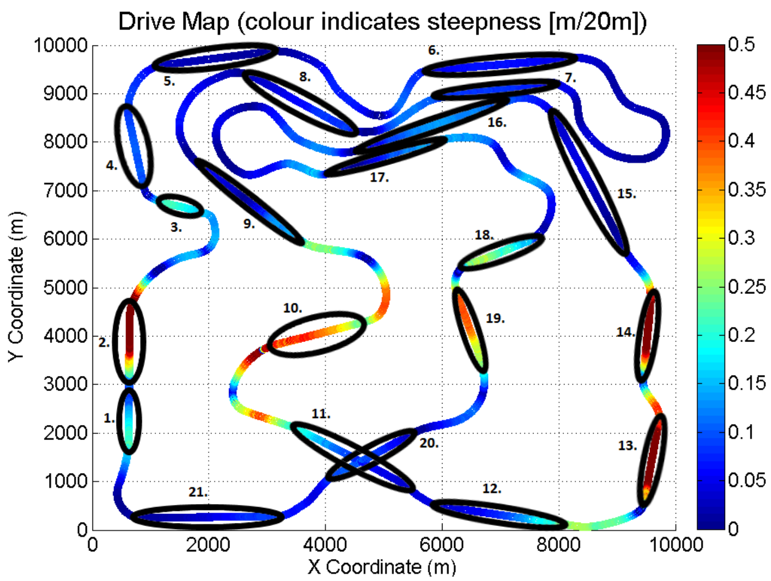

Fig. 3. Map of the simulated 110 km driving track. The numbered circled parts were defined as straights.

The simulators logged steering wheel angle and lateral lane position with a sampling frequency above 80 Hz, varying slightly depending on the required computation and communication time to render the scenery correctly. The steering wheel angle was a value ranging from [-1, 1] with -1 corresponding to 2.5 turns of the wheel to the right (clockwise) and 1 corresponding to 2.5 turns of the wheel to the left (counter clockwise).

The steering wheel angle was never converted to degrees or radians because the QNN requires the inputs to be between 0 and 1, so for each trained QNN, the steering wheel signal was scaled to a suitable level (details in sections 2.3.1 Predicting Steering and 2.3.2 Predicting Lane Departures). The lateral lane position signal measured the truck's position in the lane as the distance between the centre of the lane and the centre of the front of the truck (Fig. 4). When the centre of the truck was situated on the right half of the lane (towards the edge of the road) the lane position signal was negative. The lane width was 3.5 m and the width of the truck was 2.5 m, leaving 0.5 m of space for movement to either side without departing the lane.

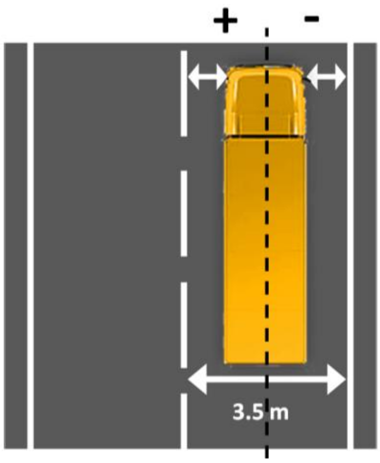

Fig 4. Definition of lane position as the distance between the centre of the lane and the centre of the front of the truck. When the centre of the truck moves to the right of the dashed line the lane position becomes increasingly negative and vice versa. The lane width was 3.5 m and the width of the truck was 2.5 m.

The entire data encompasses a total of 374 h of driving, resulting in 31 200 km of driving data. The data from the straights (Fig. 3) contained 8432 km of the data. Only data from the straights were used in this thesis to avoid additional steering traits and possible deviations from the lane caused by driving through curves.

2.3 Prediction and Analyses

Initially, the aim was to be able to predict the lane position signal directly from the steering wheel signal, using a QNN, and consequently allowing prediction of lane departures. Before delving into the details of the prediction methods, some terminology should be defined. First of all, what is a *lane departure*? A lane departure is an instance in which some part of the vehicle is outside the lane. As such, a lane departure does not necessarily mean that the vehicle has left the lane entirely, but it is sufficient for one wheel to cross the lane markings on either side. To be more precise, in this thesis a lane departure was defined as an event in which two criteria were met: 1. The lane position signal was ≤ -0.5 m or ≥ 0.5 m, corresponding to either side of the truck being on top of or outside either lane marking (the truck could move 0.5 m to the right (-) or to the left (+) from the centre position in the lane without crossing a lane marking, 2.2 Data Set, Fig. 4) and 2. the previous data point was not defined as a lane departure. The second criterion ensures that when the truck departs the lane, all movement outside the lane before returning back inside the lane, is counted as only one lane departure. Secondly, a few terms related to prediction should be defined. In the case of predicting the lane position signal from the steering wheel signal, the steering wheel signal is the *predictor variable*, i.e. the input that the prediction is based upon. The QNN that performs the prediction is a *predictor*. When predictions have been made, the *prediction horizon* determines how far into the future the predictions are reliable. Especially when predicted points are used to predict even further, it is easily understood that while the first predicted point may have a small uncertainty arising from the prediction method, the next predicted point will have an uncertainty that is the combination of both the predictor's and the previous predicted point's uncertainties, and so forth. So even with a very accurate and precise predictor, the uncertainties will increase drastically with each time step, and at some point the uncertainties make the predictions unreliable. If predictions are made just one time step forward at a time, then the prediction horizon is the length of that time step. These terms, lane departure, predictor, predictor variable, and prediction horizon will be used to describe the QNN predictions.

While the initial aim was to predict the lane position signal from the steering wheel signal and then estimate the upcoming lane departures from that, it was soon discovered that it was not feasible. The variations in the steering wheel are so much smaller in amplitude, and of much higher frequency, than the variations in lane position so the QNN simply perceived the lane position signal to be some kind of constant offset (an example is shown in Fig. 5 a). Scaling the signals was attempted, but no successful scaling was found that could produce any meaningful result for all times awake. When the drivers were most tired, both steering movements and subsequent drifts on the road were so large that all remotely useful scaling caused the signals to exceed the range [0, 1], to which the QNNs are restricted. One conclusion could still be drawn from all the failed attempts at lane position prediction, the high frequency behaviour of the steering wheel signal was always present in the predictions, and if the average of the predicted lane position signal was subtracted, the remaining predicted signal resembled the steering wheel signal to a surprisingly high degree (Fig. 5 b). These findings were the rationales for first attempting to predict only the steering wheel signal from its previous values and to then find some alternate way to predict lane departures.

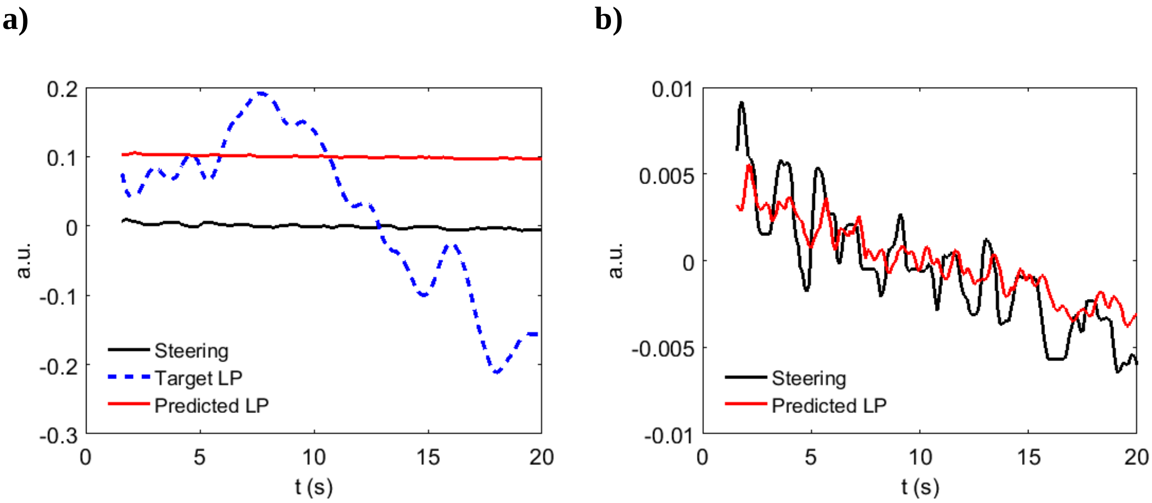

Fig. 5. Example of lane position predicted from steering. **a)** The predicted lane position signal (red) does not resemble the target lane position signal (blue dashed). **b)** The average of the predicted lane position signal was subtracted, leaving a predicted lane position signal that resembles the steering wheel signal.

All computer algorithms for the QNNs were written and implemented by the author in Matlab 2013b.

2.3.1 Predicting Steering

A time-delayed feed-forward QNN (2.1 Theory of Quantum Neural Networks) was used to predict the future steering wheel signal from previous values of the signal. The structure of the network was the following: an input layer with 15 qubit neurons, two hidden layers, the first with 40 qubit neurons and the second with 20, and an output layer with a single qubit neuron. A common notation to describe the number of neurons in each layer, which will henceforth be used, is 15-40-20-1, where one number is the number of neurons in one layer, and the layers start from the input layer on the left and end in the output layer on the right. The fifteen neurons in the input layer were each assigned one previous time step of the signal, starting from the current time t and going back to t-14, so fifteen previous time steps were used to predict the signal at the next time step t +1. The number of hidden layers, and neurons in them, determine the complexity of the network and therefore also the possible complexity of the predicted signal. Because steering wheel signals are quite erratic with both high frequency jitter and larger transients, the number of neurons in the first hidden layer should be at least twice the number of neurons in the input layer. Increased complexity also means increased computation load, especially during learning. To keep the learning time reasonable, the first hidden layer was chosen to have 40 neurons. The second hidden layer, with 20 neurons, served to detect more general traits before feeding the information forward to the single neuron in the output layer.

The steering wheel signals were downsampled to an even 10 Hz sampling frequency, resulting in a time step of 0.1 s. This also meant that the prediction horizon was 0.1 s. A few tests were made with lower sampling frequencies to produce a longer prediction horizon, but too many traits were lost from the signals. The QNN trained for the prediction one time step ahead (i.e. 0.1 s) was also fed predicted points as inputs in an effort to achieve a longer prediction horizon, but unfortunately the predictions diverged almost immediately and the start of the divergence was heavily dependent on the signal shape near the starting point, so this line of research was abandoned.

2.3.1.1 Training Set, Test set, and Learning

Out of the 8432 km of data from the straights (2.2 Data Set), 15 % of the data was selected for the training set. Data from the daylight condition (IDs 1-16, 2.2 Data Set) was used to see if there would be a difference in prediction performance due to both individual differences (since not all individuals were included in the training set) and average difference in vigilance between the groups. A lack of difference in performance could indicate that a QNN trained on a population would be able to perform well also for other individuals and that moderate differences in sleepiness would not jeopardise the prediction performance. Driving bouts (corresponding to different times awake) and straights were randomly selected from each of the daylight-condition drivers to ensure that the training set was diverse and representative. The remaining 85 % of the data was used as a test set.

Using only a 15 % training set for such a complex network (15-40-20-1) might seem like there would be a risk of overfitting, but that was not the case, because neural networks are generally overfitted due to three main reasons, none of which was applicable: 1. The training set is too monotonous, which was avoided by selecting multiple straights from multiple bouts from multiple drivers 2. the training set covers too much of the data (too much is not clearly defined, but the risk increases if more than half of the data is used for training), and 3. the number of adjustable parameters in the network is approximately equal to the number of data points in the training set. A 15-40-20-1 feed-forward structure has 1542 adjustable parameters and there was more than 50 000 data points in the training set, so the third cause for overfitting was also easily avoided.

The learning was performed with the supervised learning and backpropagation with gradient descent that was described in (2.1 Theory of Quantum Neural Networks). A learning rate of $\eta = 0.6$ was used to ensure sufficient exploration of the parameter space. Online learning was used, i.e. the first 15 points of a signal was used to predict the next point, the error between the desires and predicted signal was calculated (11) and the adjustable parameters were updated (12), the 15-point window was then moved one time step forward, a new prediction was made, the parameters were updated etc. until the end of the signal was reached. Another alternative would be to use batch learning, where several predictions are made at a time and then the parameters are updated. The advantage is that the gradient can be averaged from many predictions reducing oscillations over the parameter space, but the algorithms are more complicated to implement, especially for QNNs, and the improved learning due to the parallelism of the QNNs (compared to classical neural networks) was thought to make online learning sufficient. The QNN was trained twice on each signal in the training set.

All steering wheel signals (in both training and test sets) had to be scaled to the interval [0, 1], which is the value range of the QNN, while the signal's own range was [-1, 1]. Most parts of the signals remained approximately in the interval [-0.01, 0.01] and all signals had to be comparable to one another for the QNN to learn properly. Straightforward normalisation was not an option because the amplitude had to be increased while maintaining the same relative differences between all steering wheel signals. In the end, the scaling was performed by first multiplying the signals by 20 (to increase the amplitude) and then adding 0.5 to force the zero level of the signal to the middle of the QNN interval, thereby enabling the largest possible dynamic range. All outputs of the QNN was rescaled back to the original range by performing the reverse operations of the scaling.

The trained QNN was then used on the test set to predict the steering wheel signal one time step ahead (0.1 s).

2.3.1.2 Performance Evaluation

The predictions from the test set were evaluated using the root-mean-square error (RMSE), a common measure of the average discrepancy between a signal and a reference signal, in this case between the predicted steering wheel signal and the measured signal. It is the standard deviation of the prediction errors, i.e. the difference between predicted, $\hat{y}$, and measured, y, values:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}(\hat{y}_t - y_t)^2}{N}} \tag{13}$$

where the sum is taken over all data points. The RMSE has the same unit as the predicted and measured variables and will also have a numerical value in the same range. In this case, as the steering wheel signal had an arbitrary unit and was restricted to the interval [-

1, 1], the RMSE had the same arbitrary unit and the values equated to the same value in the steering wheel signal.

The results showed the presence of an almost constant offset in each signal and an effort was made to try to remove it, thereby improving the predictions. The most suitable method to correct the offset would probably have been to calculate the average offset in the entire signal and then subtract it, but such an extreme *post hoc* approach would not be a possibility in any real prediction system. Instead, because the offset was so constant, the difference between the first predicted and measured point was calculated and that difference was then subtracted from all other predictions of the signal. This approach requires simultaneous knowledge of the value of only one measured and predicted point of the signal. The RMSE was then calculated for the offset-corrected signals.

2.3.2 Predicting Lane Departures

Prediction of the steering wheel signal, even successfully, does not immediately translate into lane departure prediction, which was the goal of this project. If the transfer function from steering wheel angle to lane position was known, the future lane position could be calculated, but each car would have a different transfer function that would also be strongly influenced by the environment, such as road conditions, which are everything but constant. This was the motivation for making four attempts at lane departure prediction using QNNs. It was realised that the steering wheel signal as such would not be a suitable predictor variable, at least not for any long prediction horizon, so a new strategy was used: The signals were separated into 10 s bins and a steering metric from one bin was used to predict the number of lane departures in the next bin. This approach was still a prediction one time step ahead, but now the time step (and simultaneously prediction horizon) was 10 s instead of 0.1 s. The steering metric, scaling of the signals, and QNN varied with each attempt and the details of all four attempts are presented below in the sections 2.3.2.1 Steering metrics, 2.3.2.2 QNN, 2.3.2.3 Training set, Test set, and Learning, and 2.3.2.4 Performance Evaluation.

2.3.2.1 Steering metrics

In the failed attempts to predict lane position from the steering wheel signal (2.3 Prediction and Analyses) it was noticed that the high frequency traits from the input signal was transferred to the output even though the desired output signal used for learning had no such traits. Calculating some steering metric describing the variations in the 10 s bin was thought to produce an input without high frequency jitter but that would still include the necessary amount of information about the steering wheel signal required for lane departure prediction. Finding a suitable steering metric is crucial for the success of this approach, because it would need to encompass the necessary features of the steering while still being represented with only a few points that could be fed to the QNN as inputs. The standard deviation, for instance, is not a suitable steering metric because as it is calculated for e.g. a 10 s bin it is almost constant from one bin to the next. Any transients, if short enough, would not influence it significantly.

For Attempt 1, the maximum peak-to-peak amplitude of the steering in a 10 s bin was used as the steering metric. The maximum peak-to-peak amplitude was defined simply as the difference between the maximum and minimum value of the steering signal in that 10 s bin, producing one value to describe the steering in that bin. This steering metric does represent the range of the steering movements, but it does not include the variations between the extremes.

Attempt 1 barely produced any lane departure predictions, but the explanation was thought to lie in the temporal scarceness of data points. Therefore the maximum peak-to-peak amplitude was also used as the steering metric in Attempt 2, but this time the 10 s bins were divided into ten 1 s segments and the steering metric was calculated for each segment. This produced ten steering metric values that were fed to the QNN to predict the number of lane departures in the next 10 s bin. This produced a better result, but reliable lane departure prediction was still not achieved.

A new steering metric was tested in Attempt 3: the integral of the steering wheel signal for each 1 s segment in the 10 s bin. The integral is clearly not a good measure of volatility or transients in a signal, but the rationale was that as the truck acts as a very powerful low-pass filter between the steering and the lane position, there should be some correlation between the integral of the steering and where in the lane most time is spent. It seems intuitive that if you spend a long time moving towards either edge, then the chance of crossing the lane marker would increase, and that if you drift back and forth evenly around the centre of the lane, then the integral would be close to zero and you would not be as likely to cross the lane markings. It was assumed that the QNN could learn some traits or correlations between these hypothetical probabilities of lane departures and the actual number of measured lane departures. Unfortunately, this did not improve the results, only change them by increasing the number of predicted lane departures but doing so incorrectly, and there still seemed to be a problem caused by having too many bins with no lane departures in the training set. Because this steering metric still had produced the most predicted lane departures it was kept the same for Attempt 4 but the training set was chosen more selectively to have more lane departures.

2.3.2.2 QNN

A feed-forward QNN was used for all four attempts to predict the number of lane departures in the next 10 s bin, but the network structure was changed slightly based on changes in input, i.e. change of steering metric.

The steering metric in Attempt 1 was the maximum peak-to-peak amplitude of the steering calculated for the entire 10 s bin producing only one input value from the bin. A 1-10-5-1 structure was used with only one neuron in the input layer for the one input value, ten neurons in the first hidden layer to allow complexity of the network, five neurons in the second hidden layer to reduce the complexity of the first layer and detect general features, and one output neuron in the output layer. The steering metric was scaled for the QNN by multiplying it by 30, which was small enough to keep the scaled steering metric within the [0, 1] interval, but the values of normal uneventful driving were very low, in the order of 0.05. The number of lane departures in the 10 s bins were divided by ten to ensure all but the most extreme cases would stay within the interval [0, 1]. The assumption was made that a lane departure would last approximately 1 s, allowing time for ten lane departures in one 10 s bin. The scaling of the lane departures was kept the same in all attempts.

Attempt 2 used the same steering metric as in Attempt 1 (maximum peak-to-peak amplitude of the steering), but it was calculated separately for each 1 s segment of the 10 s bin, producing an input with ten values. A QNN with a 10-15-5-1 structure was chosen. The steering metric was scaled to a slightly higher dynamic range by multiplying it by 60 (instead of 30 as in Attempt 1) causing the largest peak-to-peak amplitude to exceed the range [0, 1].

The steering metric was changed for Attempt 3 to the integral of the steering of each 1 s segment of the 10 s bin, which kept the number of input neurons at ten, but due to unsatisfactory prediction performance of Attempt 2, the number of neurons in each hidden layer was increased by five to allow more complexity leading, to a 10-20-10-1 QNN. Because the new integral of the steering could have negative values, the steering metric was scaled by multiplying it by 60 and by adding and offset which pushed the lowest value to zero.

The steering metric and QNN structure was kept the same in Attempt 4 as they were in Attempt 3, only a more selective training set was used that contained more lane departures. The scaling was performed in the same way as in Attempt 3.

2.3.2.3 Training set, Test set, and Learning

Each QNN, for all four attempts, was trained on different training sets, those for Attempts 1-3 constituting 42 % of the data from the straights, 3538 km of driving, and the training set for Attempt 4 constituting 51 % of the data. The test sets were always all remaining data not included in the training set. The training data was selected from all drivers, not just the ones from the daylight condition as in the steering signal prediction. In Attempts 1-3, 6.3 % (=√40%) of the bouts were randomly selected for each driver, and out of those 6.3 % of the straights from each bout, which should provide 40 % of the data, but due to the differences in lengths of the straights, the actual percentage of training data was calculated, and in each case the real percentage was approximately 42 %.

It seemed, from the experiences from Attempts 1-3, like the training set contained too many 10 s bins with no lane departures, causing the network to predict little else. To get a feeling for how the training set should be chosen to contain more lane departures, the number of lane departures for each driver and time awake was plotted (Fig. 6). The largest numbers of lane departures occur around 25 h time awake, so the training set for Attempt 4 was chosen as all data from bouts 4-9, corresponding to 12-27 h time awake. This choice of training set was 51 % of the data, which is quite a large percentage, but bouts 4 and 5 were included to ensure that there would be straights with no lane departures.

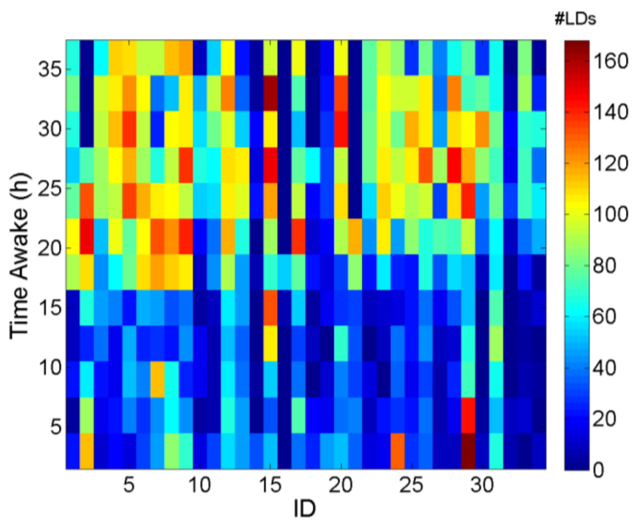

Fig. 6. The number of lane departures (colour bar) as a function of both driver ID (x-axis) and time awake (y-axis) from the straights.

The learning of the QNNs was performed using the supervised learning and backpropagation with gradient descent that was described in (2.1 Theory of Quantum Neural Networks) eqs. (11) and (12). The learning rate was 0.6 in all cases. As in (2.3.1.1 Training Set, Test set, and Learning), online learning was used, i.e. the steering metric from one 10 s bin was used to predict the number of lane departures in the next 10 s bin, parameters were updated, then the next bin was used as input, prediction was performed, parameters were updated etc. The learning was repeated for four iterations in Attempt 1 and for three iterations in Attempts 2-4.

2.3.2.4 Performance Evaluation

The root-mean-square error (13) was used to determine the average discrepancy between the predicted and measured number of lane departures for all four attempts. In this case, the RMSE had no unit (the number of lane departures is a unitless quantity) and the value equated the number of lane departures. The percentage of bins with correctly predicted lane departures was also calculated. A prediction of no lane departures, if there were none, was also a correct prediction.

The predicted number of lane departures during one 10 s bin is a discrete event and the correct (i.e. measured) number is known, so the sensitivities and specificities of the QNN predictions could be calculated when the QNN was thought of as a binary classifier, i.e. it predicted either at least on lane departure in a bin or none. Sensitivity and specificity are commonly used to evaluate the performance of classifiers, e.g. medical tests. Let us call one outcome positive and the other negative, in this case "at least one lane departure" is a positive and "no lane departures" a negative. The sensitivity is the proportion of positives that are correctly identified as such while the specificity is the proportion of negatives that are correctly identified as such [48]. In the case of lane departure prediction, the sensitivity is the proportion of bins with at least one lane departure that are correctly predicted as such and the specificity is the proportion of bins with no lane departures that are correctly predicted as such. The equations for sensitivity and specificity are:

$$Sensitivity = \frac{\sum TP}{\sum TP + \sum FN} \tag{14}$$

$$Specificity = \frac{\sum TN}{\sum TN + \sum FP} \tag{15}$$

where the capital sigmas are the sums over all of the letter combinations. The letter combinations mean the following: TP, True Positive, is a bin with at least one lane departure and for which the QNN has predicted at least one lane departure. FP, False Positive, is a bin with no lane departures but for which the QNN has predicted at least one lane departure. TN, true negative, is a bin with no lane departures and for which the QNN has predicted no lane departures. FN, False Negative, is a bin with at least one lane departure but for which the QNN predicted no lane departures. With these definitions the sensitivity and specificity can perhaps be better understood. Looking at (14) one can understand that the sum of true positives and false negatives is the total amount of bins with at least one lane departure in them, but the false negatives are not predicted to have that. Therefore, sensitivity is the proportion of bins with at least one lane departure that are correctly predicted as such. Then again, looking at (15) one can understand that the sum of true negatives and false positives is the total amount of bins that have no lane departures in them, but the false positives are still predicted to have lane departures. Therefore, specificity is the proportion of bins with no lane departures that are correctly predicted as such.

If both sensitivity and specificity are high, then it is indicative of a good classifier, or in this case, predictor, because then most upcoming lane departures will be predicted (sensitivity) and the predictor will not predict any lane departures when there are none (specificity). However, one must be a bit careful with these quantities, because either the sensitivity or the specificity can always be forced to 100 %. If a predictor is made, which predicts lane departures all the time, then the sensitivity will be close to 100 % because no actual lane departure will be missed. In this case the specificity drops drastically, of course, because there will be a big number of false positives. On the other hand, if a predictor is made, which never predicts any lane departures, then the specificity will be 100 % but the sensitivity will be minute, because no lane departures will be predicted. If a predictor is good, then both sensitivity and specificity can be high, but there is usually a trade-off between the two.

The results from Attempt 3 also showed an offset in the predictions which was also translated to the RMSE as a function of driver, very much like the offsets that were present in the prediction of steering (2.3.1.2 Performance Evaluation). A *post hoc* offset correction was performed for both Attempts 3 and 4 to explore whether or not the offset correction could be beneficial. The correction was made by subtracting the mean of the driver-specific RMSEs from the predicted number of lane departures.

3 Results

The results are divided into two sections, 3.1 Predicting Steering and 3.2 Predicting Lane Departures, as in Methods, because predicting a semi-continuous signal from a similar signal (in this case the previous parts of the same signal), as is the case for the steering, and predicting a more discrete signal from summary statistics of another signal are two surprisingly different tasks.

3.1 Predicting Steering

The steering wheel signal was predicted one time step ahead (0.1 s due to the 10 Hz sampling frequency) with the time-delayed feed-forward QNN with a 15-40-20-1 structure (2.3.1 Predicting Steering). The test set covered the remaining 85 % of the data from the straights, excluding the training set. Even though the training set only contained data from driver IDs 1-16, the root-mean-square error (RMSE) was quite stable across the different drivers (Fig. 7). Interestingly enough, the largest RMSE for any single driver occurred for a driver that was part of the training set. This shows that the QNN also worked for the night-time driving scenario even though it was trained only on the daylight scenario. This result shows promise that a QNN-based prediction system implemented in a car might not need to be trained separately for driving in dark conditions.

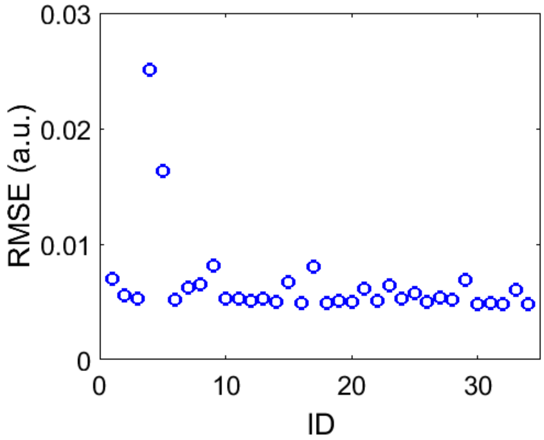

Fig. 7. Root-mean-square error (RMSE) of the predictions from the test set calculated separately for each driver. The unit of the RMSE is the same arbitrary unit as that of the steering wheel signal, constrained to [-1, 1]. The training set contained signals from driver IDs 1-16, who drove the daylight scenario.

The RMSE was also examined as a function of time awake to determine the QNN's prediction ability as the drivers' drowsiness increased. The results are presented in Fig. 8. Aside from a few outliers at 33 h time awake, the RMSE is remarkably stable even across time awake. Again there is no difference between the groups driving the day- and night-time scenarios. It is worth mentioning that while it might look unintuitive that the RMSEs as a function of time awake reach higher values than the RMSEs for one driver, the explanation is simple: The RMSE is the average of the squared deviation from the measured signal, which means that the length of the signal matters. If one driver has a few poorly predicted signals during one bout (i.e. one time awake) it will substantially influence the RMSE for that one time awake, but as the RMSE is calculated in its entirety for all of the drivers' signals, the effect will be much smaller.

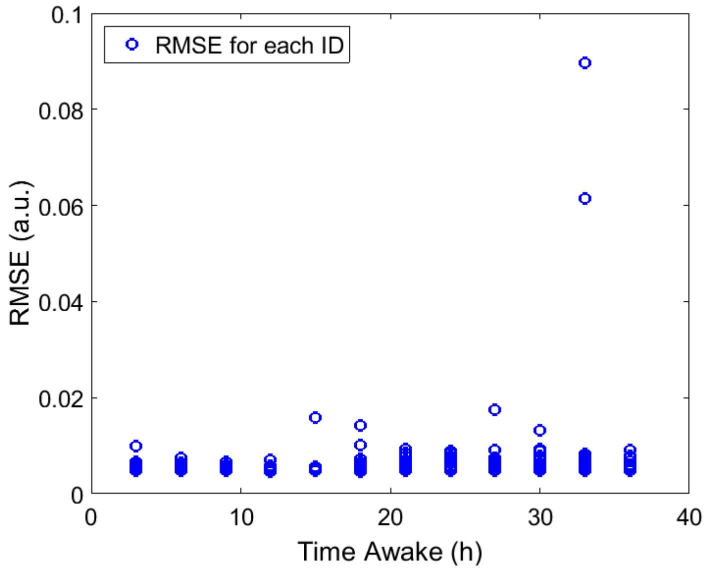

Fig. 8. Root-mean-square error (RMSE) of the predictions from the test set calculated separately for each driver and time awake. One circle is the RMSE for one driver at one particular time awake. The unit of the RMSE is the same arbitrary unit as that of the steering wheel signal, constrained to [-1, 1]. The training set contained randomly selected signals from each time awake.

The RMSE for the entire test set was $RMSE_{tot}$ = 0.007 a.u. (same arbitrary units as in the steering wheel signal). Calculated as an average for the different drivers, the error was RMSE = 0.007 ± 0.004 a.u. (μ ± σ). As the RMSE is only a measure of the average deviation of the prediction from the measured signal, the numerical value is, naturally, linked to the numerical value of the signal. Furthermore, the RMSE does not in itself say much about how the predicted signal deviates from the measured one. Because the steering wheel signal could vary between [-1, 1], the RMSE corresponds to a 0.4 % relative error. The desirable error level during learning of QNNs is usually at most 1 %, so this network performs within desired parameters. On the other hand, while driving on straights the steering wheel signal usually only varied between [-0.01, 0.01], meaning that the RMSE could cause even a 36 % relative error. To also gain an understanding of how the QNN predicted the signals, and subsequently when it failed to predict the signals, examples of the predicted signal with a small RMSE and a large RMSE are presented in Fig. 9. From Fig. 9 a) an offset between the predicted and the measured signal is discernible. A similar offset is visible in Fig. 9 b). The predicted signals follow the shape of the measured signals very well, but in Fig. 9 b) something unusual has clearly occurred in the measured signal that the QNN could not handle. However, as the measured signal returns to normal, the QNN quickly recovers and resumes prediction with the same performance as before the disruption.

a) **b)**

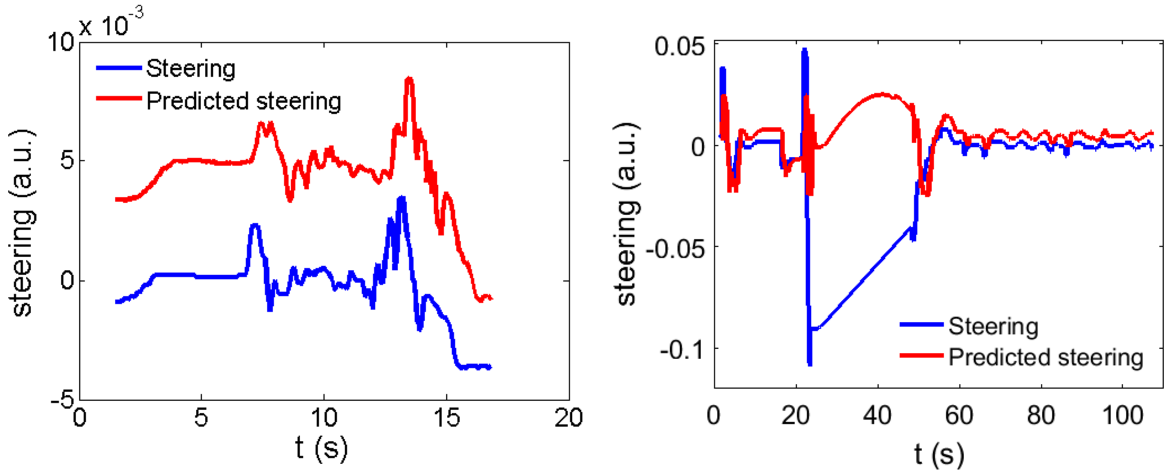

Fig. 9. Examples of predicted steering wheel signals. The steering is in arbitrary units between [-1, 1] with -1 being 2.5 turns of the wheel to the right and vice versa. **a)** Predicted signal with a small RMSE. **b)** Predicted signal with a large RMSE with a clear atypical event after t=20 s. When the measured signal returns to normal the QNN recovers its prediction ability. An offset is visible in both predicted signals.

Due to the almost constant offset present in the signals, an attempt was made to remove it by subtracting the difference between the first predicted and measured point from all predicted points in the signal. The RMSE for the entire test set and as a function of driver and time awake were calculated (Fig. 10 a and b, respectively). Surprisingly, removing the offset in this manner caused a slight increase in RMSE, $RMSE_{tot}$ = 0.011, and calculated as an average between different drivers the error was RMSE = 0.007 ± 0.009 a.u. ($\mu \pm \sigma$). Compared to the non-corrected approach, the average RMSE for the drivers remained the same but the variation had increased.

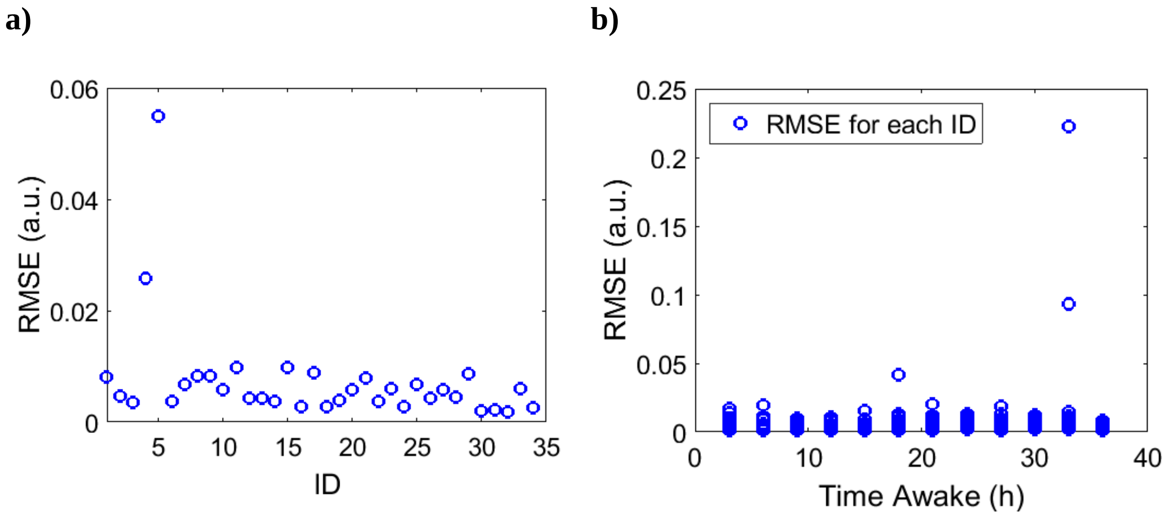

Fig. 10. Root-mean-square error (RMSE) of the offset-corrected predictions from the test set calculated separately for **a)** each driver and **b)** time awake. One circle is the RMSE for one driver at one particular time awake. The unit of the RMSE is the same arbitrary unit as that of the steering wheel signal, constrained to [-1, 1]. The training set contained randomly selected signals from each time awake from driver IDs 1-16, who drove the day-time scenario.

To understand why the RMSE increased, two example signals are shown in Fig. 11. The signal in Fig. 11 a) is the same signal as in Fig. 9 a), now with the offset removed. In this case removing the offset, only using the difference between the first predicted and measured point, has brought the prediction closer to the measured signal. However, Fig. 11 b) shows that the overall RMSE has increased compared to the original prediction with the offset. The explanation is found looking at Fig. 11 c), which is the beginning of the signal in b): The very first predicted data point has a smaller value than the predicted signal, which means that subtracting that negative offset from the predicted signal yields an offset-corrected signal that has larger values than the original predicted signal. When the measured signal begins to rise, the predictions overshoot, and with the offset-corrected signal containing even larger values than the original prediction, the offset-corrected signal is actually farther away from the measured signal causing a higher RMSE.

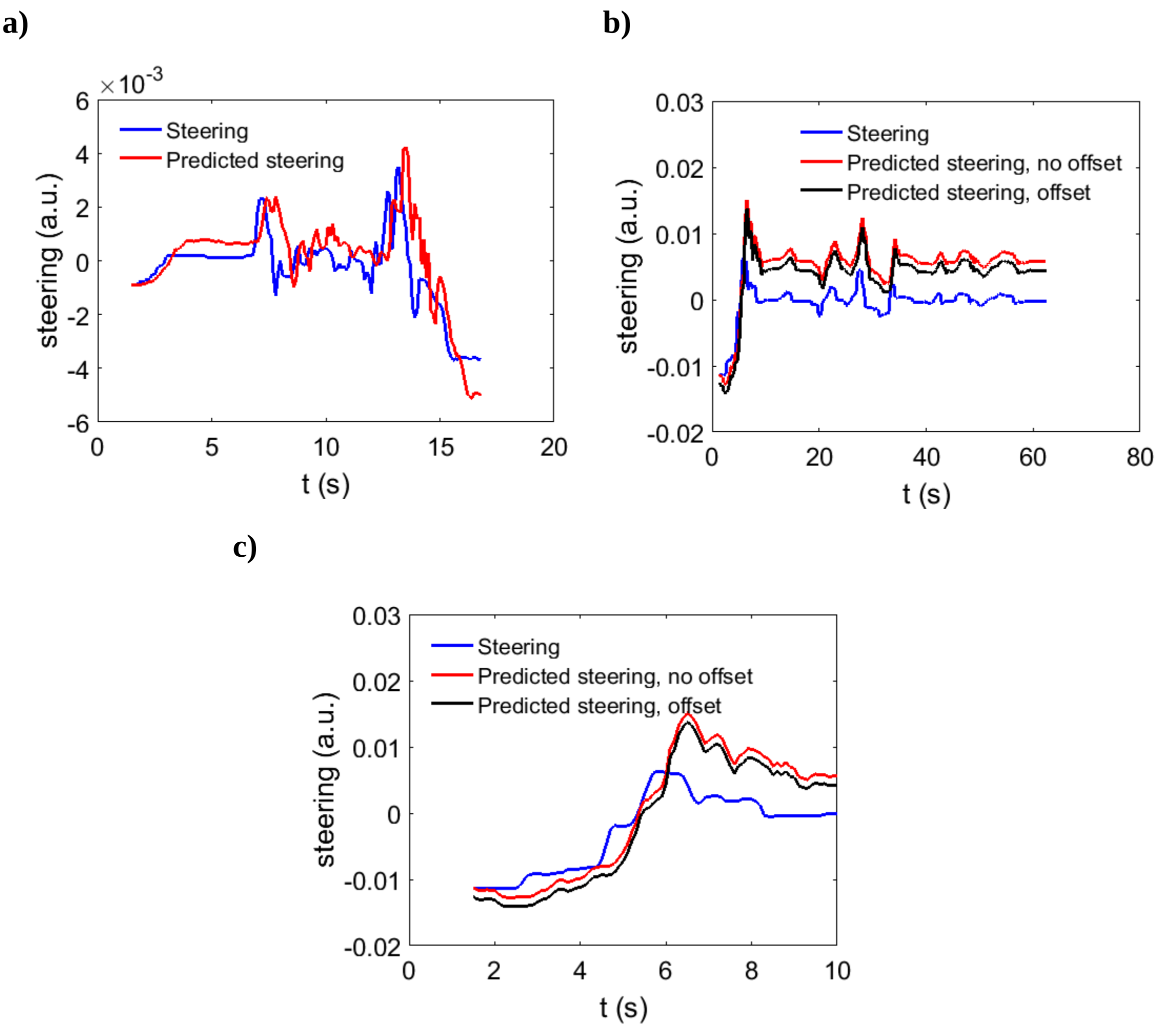

Fig. 11. Examples of offset-corrected predicted steering wheel signals. The steering is in arbitrary units between [-1, 1] with -1 being 2.5 turns of the wheel to the right and vice versa. **a)** Predicted signal with a small RMSE (Fig. 9 a) that was improved by the correction. Now the signals are clearly closer together. **b)** Predicted offset-corrected signal with a larger RMSE than the original prediction. The original prediction (black) is closer to the measured signal (blue) than the offset-corrected signal (red). **c)** Beginning of the signal in b). At first the predicted signal is smaller than the measured, causing the offset-corrected signal to be larger than the original prediction, which, after the overshoot at 7 s, leads to an even larger RMSE than in the original prediction.

A 0.1 s prediction horizon is rather short, so an attempt was made to use the QNN trained for predicting one time step ahead to predict several time steps ahead. Unfortunately, using predicted points to feed back to the QNN ended in failure. An example is shown in Fig. 12. As is seen from Fig. 12 a), the prediction diverges very fast. Fig. 12 b) shows the first part of the signal, and the predictions start oscillating almost immediately. Exactly where the divergence occurs is also not a constant, but a function of the shape of the signal at the first 15 measured points used for the first predicted point; a flatter measured signal generally prolongs the prediction horizon because then the QNN does not assume any large transients. This causes the starting point of the prediction to have too much of an impact for this type of further prediction to succeed.

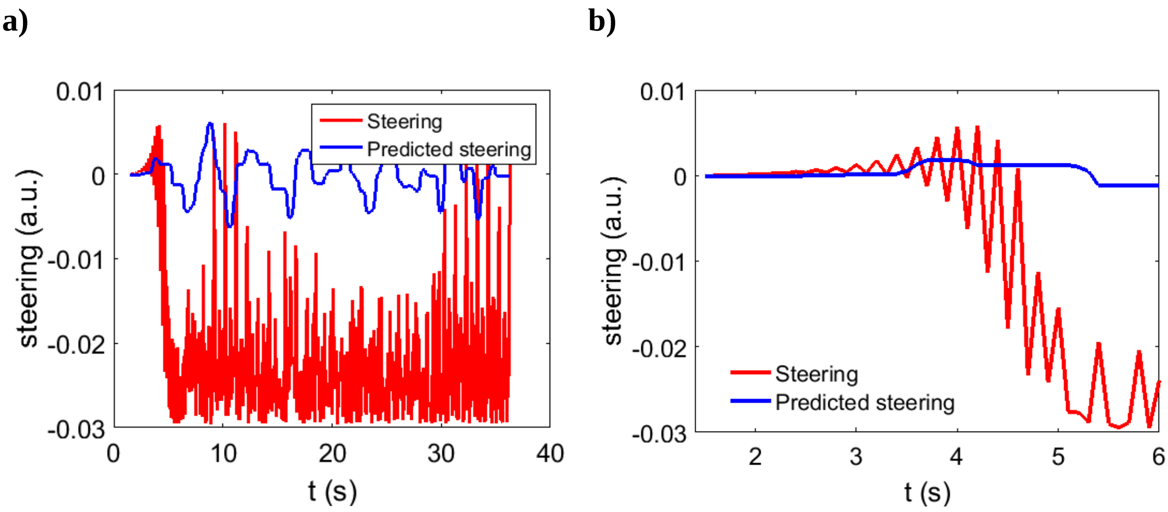

Fig. 12. a) Example of divergence when predicted points are used as predictor variables. **b)** Beginning of signal in a).

3.2 Predicting Lane Departures

Four attempts were made to predict lane departures from steering. As each attempt led to modifications to improve the next attempt, the main conclusions were already mentioned briefly in Methods, but here the results are presented in the same chronological order as in which they were conducted.

In the first attempt to predict lane departures from the steering wheel signal the data was separated into 10 s bins. The maximum peak-to-peak amplitude in the steering wheel signal from one bin was used to predict the number of lane departures in the next 10 s bin. A 1-10-5-1 structure was used for the QNN and the test set consisted of the remaining 58 % of the data not used in the training set. The root-mean-square error (RMSE) as a function of driver ID and of time awake are presented in Fig. 13 a) and b), respectively. This simple attempt at lane departure prediction was clearly not adequate to produce any useful prediction. Upon closer inspection it was discovered that the QNN only predicted four lane departures in the entire test set, suggesting that it only learned to predict no lane departures almost regardless of the steering wheel signal.

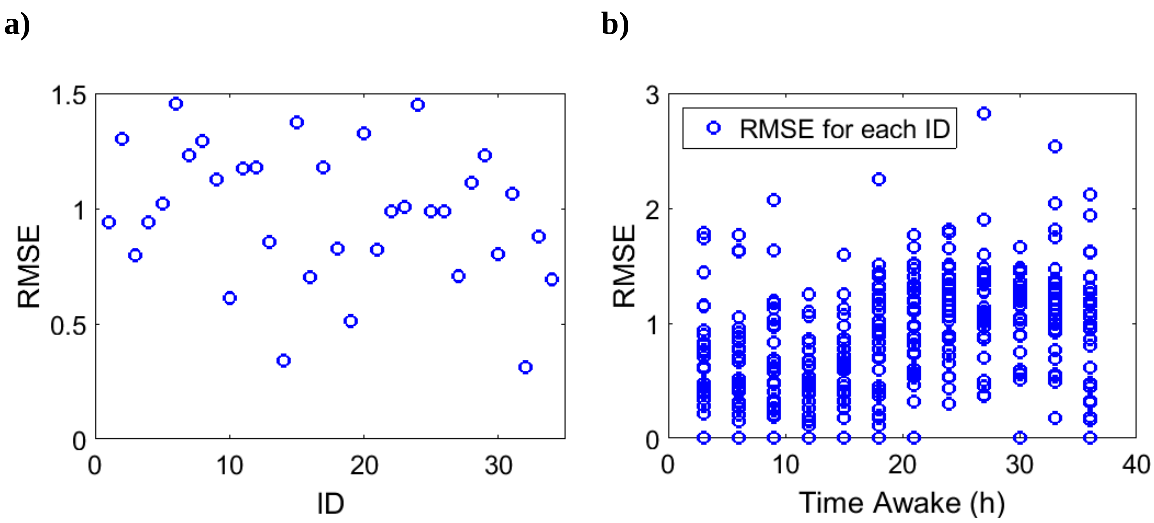

Fig. 13. Root-mean-square error (RMSE) of the predicted number of lane departures for **a)** each driver and **b)** each driver at each time awake, when the maximum peak-to-peak amplitude of the steering from one 10 s bin was used to predict the number of lane departures in the next 10 s bin. One circle represents the RMSE for one driver, and in b) also for one time awake. The RMSEs have no units as the predicted quantity is the number of lane departures during the next 10 s.

In the next attempt to predict lane departures from the steering wheel signal, the 10 s bin of the steering wheel signal was further split into ten 1 s segments. The maximum peak-to-peak amplitudes in the steering wheel signal for each 1 s segment was used to predict the number of lane departures in the next 10 s bin. The QNN had a 10-15-5-1 structure (one input neuron for each 1 s segment) and the QNN was tested on a new test set also comprising 58 % of the data from all straights. The RMSE as a function of driver ID and of time awake are presented in Fig. 14 a) and b), respectively.

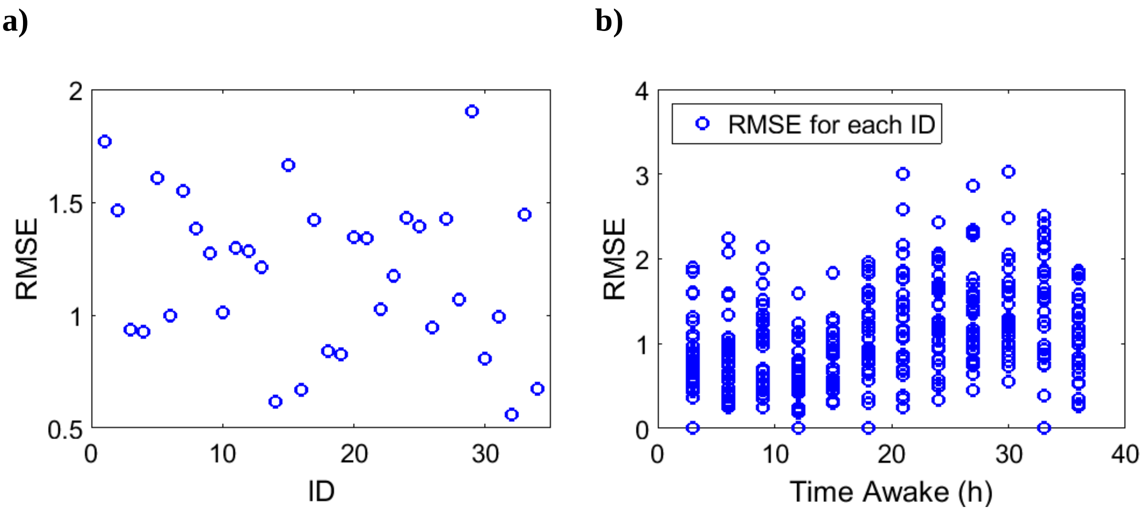

Fig. 14. Root-mean-square error (RMSE) of the predicted number of lane departures for **a)** each driver and **b)** each driver at each time awake, when the maximum peak-to-peak amplitude of the steering from ten 1 s segments was used to predict the number of lane departures in the next 10 s bin. One circle represents the RMSE for one driver, and in b) also for one time awake. The RMSEs have no units as the predicted quantity is the number of lane departures during the next 10 s.

This method of prediction yielded a correct prediction in 55 % of the cases (including predicting no lane departures when there were none) and it predicted an amount of lane departures that reached 71 % of the total number of lane departures in the test set, meaning that the problem with predicting only no lane departures was, at least partially, amended. The sensitivity and specificity (2.3.2.4 Performance Evaluation) calculated as a binary classifier, i.e. either predicting at least one (or more) lane departure or predicting no lane departures, are presented in Table 1. The high specificity compared to the low sensitivity does, however, suggest that there are many predictions of no lane departures, but also that there are quite a few 10 s bins in the test set that contain no lane departures.

Predictor variable (steering)	Maximum peak-to-peak amplitude during ten previous 1 s segments.
% correct predictions	55 %
Sensitivity	31 %
Specificity	80 %

Table 1. Performance of 10-15-5-1 QNN to predict number of lane departures during the next 10 s from the predictor variable. % correct predictions includes predicting no lane departure correctly. Sensitivity and specificity are calculated for a binary classifier, i.e. either at least one or no lane departures.

As apparently splitting the 10 s bins into 1 s segments of the steering wheel signal produced better predictions, this segmentation was kept the same for the next iteration and attempts were made to improve other features. The integrals of the 1 s steering segments were used as new predictor variables and the complexity of the QNN was increased in the new 10-20-10-1 structure (adding five neurons to both hidden layers). The QNN was tested on a yet another test set comprising 58 % of the data from all straights. The RMSE as a function of driver ID and of time awake are presented in Fig. 15 a) and b), respectively.

a) **b)**

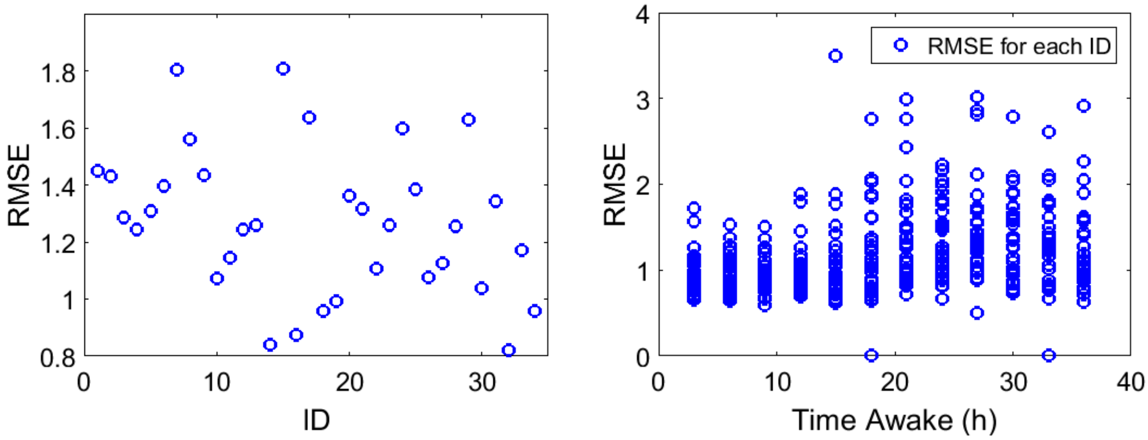

Fig. 15. Root-mean-square error of the predicted number of lane departures for **a)** each driver and **b)** each driver at each time awake, when the integral of the steering from ten 1 s segments was used to predict the number of lane departures in the next 10 s bin. One circle represents the RMSE for one driver, and in b) also for one time awake. The RMSEs have no units as the predicted quantity is the number of lane departures during the next 10 s.

The percentage of correct lane departure predictions, as well as the sensitivity and specificity (2.3.2.4 Performance Evaluation) (calculated as a binary classifier, i.e. either predicting at least one lane departure or predicting no lane departures), are presented in Table 2. The higher sensitivity and lower specificity compared to the previous attempt indicate an increase in predicted lane departures, but incorrectly, causing the specificity to decrease while the sensitivity increased.

Predictor variable (steering)	Integral of steering during ten previous 1 s segments.
% correct predictions	39 %
Sensitivity	55 %
Specificity	44 %

Table 2. Performance of 10-20-10-1 QNN to predict number of lane departures during the next 10 s from the predictor variable (i.e. steering metric). % correct predictions includes predicting no lane departure correctly. Sensitivity and specificity are calculated for a binary classifier, i.e. either at least one or no lane departures.

An offset in the predictions seemed to be present, especially looking at Fig. 15 b, as was also the case in the prediction of steering. To explore whether or not removal of such an offset could be beneficial, a *post hoc* offset removal was performed by subtracting the mean of the driver-specific RMSEs from the predicted number of lane departures. The offset-corrected RMSEs are shown in Fig. 16.

a) **b)**

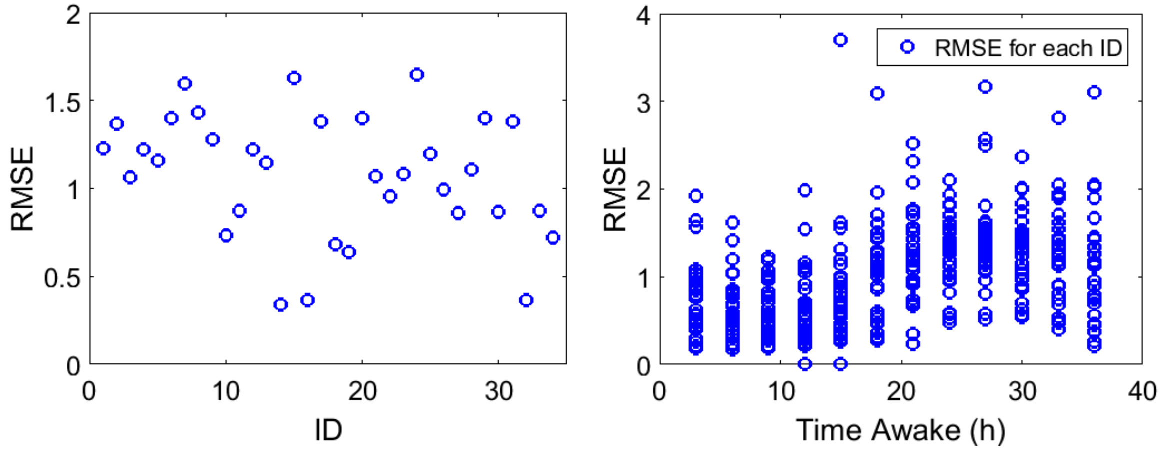

Fig. 16. Root-mean-square error of the offset-corrected predicted number of lane departures for **a)** each driver and **b)** each driver at each time awake, when the integral of the steering from ten 1 s segments was used to predict the number of lane departures in the next 10 s bin. One circle represents the RMSE for one driver, and in b) also for one time awake. The RMSEs have no units as the predicted quantity is the number of lane departures during the next 10 s.

The percentage of correct lane departure predictions, as well as the sensitivity and specificity (2.3.2.4 Performance Evaluation) (calculated as a binary classifier, i.e. either predicting at least one lane departure or predicting no lane departures), of the offset-corrected predictions are presented in Table 3.

Predictor variable (steering)	Offset-corrected integral of steering during ten previous 1 s segments.
% correct predictions	59 %
Sensitivity	12 %
Specificity	96 %

Table 3. Performance of 10-20-10-1 QNN to predict offset-corrected number of lane departures during the next 10 s from the predictor variable (i.e. steering metric). % correct predictions includes predicting no lane departure correctly. Sensitivity and specificity are calculated for a binary classifier, i.e. either at least one or no lane departures.

The increase in correct predictions after the offset correction combined with the decrease in sensitivity and increase in specificity speak of a clear preference to predict no lane departures. To gain insight into this, the histogram of correct and incorrect predictions, both with and without offset correction, is presented in Fig. 17. The most numerous correct prediction is quite clearly the prediction of no lane departures, which also explains why removing the offset, which pulls the predictions closer to zero, gives a higher percentage of correct predictions.

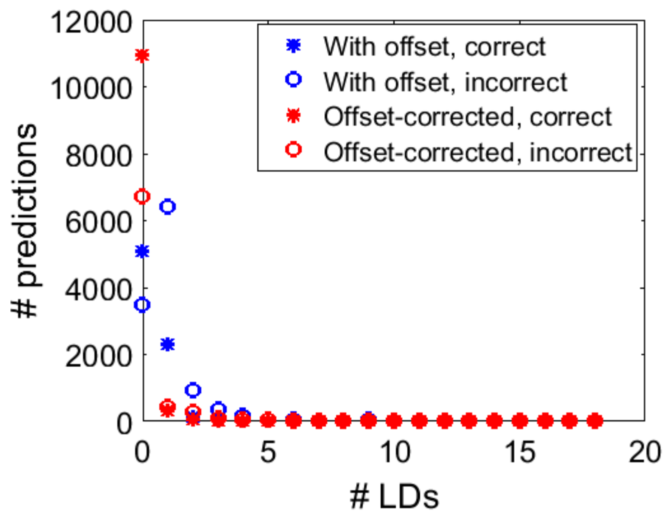

Fig. 17. Histogram of lane departure predictions from the integral of the steering during ten previous 1 s segments. Blue markers show predictions given by the QNN, red markers show offset-corrected predictions. Asterisks show the number of times the prediction was correct while circles show the number of times the prediction was incorrect.

A final attempt was made to improve the lane departure predictions by making sure that there were enough lane departures in the training set. The same prediction was performed as in the previous attempt (predicting number of lane departures during in the next 10 s bin from the integral of the steering in the ten previous 1 s segments using a 10-20-10-1 QNN, both with and without offset correction), but with the 51 % training set chosen to include only straights from bouts 4-9 corresponding to 12-27 h time awake (2.3.2.3 Training set, Test set, and Learning, Fig. 6). This produced a slight improvement in the original predictions, but not with the offset correction. The percentage of correct lane departure predictions, sensitivity, and specificity (2.3.2.4 Performance Evaluation) are presented in Table 4. The histogram of correct and incorrect predictions, both with and without offset correction, is presented in Fig. 18.

	Original prediction	**Offset-corrected prediction**
Predictor variable (steering)	Integral of steering during ten previous 1 s segments.	Integral of steering during ten previous 1 s segments.
% correct predictions	54 %	59 %
Sensitivity	28 %	7 %
Specificity	80 %	95 %

Table 4. Performance of 10-20-10-1 QNN to predict number of lane departures during the next 10 s from the predictor variable (i.e. steering metric). % correct predictions includes predicting no lane departure correctly. Sensitivity and specificity are calculated for a binary classifier, i.e. either at least one or no lane departures. Training set was limited to 12-27 h time awake.

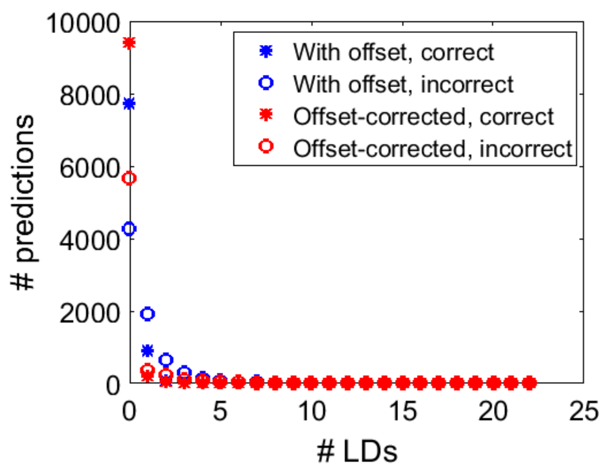

Fig. 18. Histogram of lane departure predictions from the integral of the steering during ten previous 1 s segments. Blue markers show predictions given by the QNN, red markers show offset-corrected predictions. Asterisks show the number of times the prediction was correct while circles show the number of times the prediction was incorrect. Training set was limited to 12-27 h time awake.

4 Discussion

One of the most attractive features of Quantum Neural Networks (QNNs), or of any neural network, is their ability to learn to perform several different and complex tasks on their own. This learning enables them to find traits and features in data that a person might easily miss, and this, in turn, eases the burden of the user to be able to construct complicated models for complex systems, a task which is cumbersome, if not impossible in some cases. But all this independence of the network does have significant drawbacks. Learning on its own also means that the user cannot know exactly how the network functions. The parameters of the neurons can, of course, be checked from the trained network, but the reasons for them being exactly what they are can never be determined. As an example, the QNN that predicted the future steering wheel signal one time step ahead (2.3.1 Predicting Steering), and did so rather successfully, had 1542 adjustable parameters. The equations for each qubit neuron are known, so the network's operation could be written as a function with 1542 parameters and with the steering wheel signal as an independent variable. A function with 1542 parameters is in itself something of a monstrosity to deal with, but more importantly, there are latent dependences between many of the parameters and therefore it cannot be determined exactly how one parameter influences the performance of the network. All is well as long as the network performs its assigned function at a sufficient performance level and the user does not need to consider any details; the real difficulty arises when the network does not perform the desired function, as was the case for the lane position and lane departure predictions. Because it is impossible to determine exactly how each parameter influences the entire network, it is also impossible to know which parameters cause the network to do something other than desired. In essence, this inability to understand the inner workings of neural networks makes them "black boxes", i.e. when they do work properly there is no way to know why and when they do not work properly it is equally difficult to know why. Not knowing why a network fails makes it very difficult to improve it, but there are some general causes of failures, e.g. unsuitable training sets, convergence to a local minimum during learning, unsuitable structures etc., which are discussed below in the more detailed discussion on each of the predictions presented in this thesis.

The failure to predict the lane position signal directly from the steering wheel signal provided some interesting insight into the behaviour of the time-delayed feed-forward QNN. An obvious factor in determining the learned function of the QNN was the scaling (2.3, Fig. 5 a). If there is a big difference in scale between the input and output signals, the QNN will have trouble connecting features from the two. With very different input and output signals, care should always be taken with the scaling to increase the chances of successful learning. The enormous difference in frequency content between the steering wheel signal and the lane position signal was another distinct problem — the high-frequency steering movements were not correctly translated to low-frequency lane drifting, instead the QNN perceived the lane position signal as a constant offset (2.3, Fig. 5 a). In hindsight this is understandable, the input took the 15 previous time points to predict the next and then moved the time window by one time step, which in seconds translates to using 1.5 s of steering wheel signal to predict the next 0.1 s lane position signal and then moving the entire window by 0.1 s. The fluctuations in the steering wheel signal are visible in this time scale while the lane position signal remains constant over several 0.1 s time steps. The surprise is that removing the offset from the predicted lane position signal produced a signal that resembled the input, the steering wheel signal, instead of being completely constant (2.3, Fig. 5 b). Apparently, traits of the input signal are easily transferred through the network to the output, which does explain why much neural network-based time series prediction is done using previous parts of the same signal. This last attribute was the inspiration for predicting future steering from previous steering.

The QNN succeeded rather well in predicting the steering wheel signal one time step ahead, i.e. 0.1 s, from previous parts of the signal (3.1). Apparently the 15 % training set was both large, diverse, and representative enough as such a good result was achieved. It was encouraging to see that the QNN performed well also for drivers that weren't part of the training set (3.1, Fig. 7) and for moderate differences in alertness (3.1, Fig. 8) (the daylight group used for training was, on average, more alert at all times [44]). This result showed that if QNNs are developed to be part of some sleepiness prediction system based on steering wheel signals, then it could be possible to use a selected population undergoing sleep deprivation for learning instead of forcing each customer to undergo sleep deprivation in order to train the QNN specifically for them. There was, however, a clear constant offset between the predicted and measured signal which would need to be addressed (3.1, Fig. 9). It most probably arises as an average effect from the entire training set, and considering the training set contained a combination of 16 drivers, each with their own driving style, twelve times awake, each driver having a different susceptibility to drops in vigilance, and 21 different straights driven in both directions, it is perhaps not surprising that there will be differences between signals, which on average could cause this type of offset. Using online learning instead of batch learning could also be a contributing factor, because all parameters were updated based on each prediction (online) instead of as an average from several predictions (batch). The attempt to remove the offset using the difference between the first predicted and measured point was not successful, i.e. it did not lead to an improvement from the original predictions. Using only one predicted point was not enough, but using a few short (e.g. 5 s) signals to calculate the average offset for one driver could perhaps lead to improvements (3.1, Fig. 11 c). Despite the ability of the QNN to predict the shape of the signal, the error was too large to produce any useful predictions further ahead using predicted points as inputs, even when the offset was corrected (as in the example in Results, Fig. 12). Fifteen previous time steps, i.e. 1.5 s, is, after all, quite a short part of the signal, which makes the QNN very sensitive to the starting point. If the starting point is very flat, then the QNN makes predictions that are close to the previous points, but if the starting point happens to have a steep slope, then the predictions will diverge almost immediately as the QNN will assume that this trend will continue and no measured point is used to show that the transient will change direction and even out (2.3.1 and 3.1, Fig. 12). How to increase the prediction horizon to e.g. a few seconds is not self-evident. Using several output neurons, each one assigned to a different future time step, could perhaps be a solution. Then again, adding neurons to the output layer might require an increase in neurons in the hidden layers, or more hidden layers, which will make learning very computationally heavy. Using a similar QNN structure but with the output neuron assigned to a different time step, e.g. $t+10$, could also be a solution, but determining which time step it should be is not trivial and simply training several alternative QNNs is possible but takes time, especially using online learning which can't be parallelised (because the parameters are updated after each time step).

Even though predicting lane departures was the primary goal of this research, it proved to be an elusive task. None of the four attempts (2.3.2 Predicting Lane Departures) were successful enough to show promise. The first attempt (Attempt 1) was clearly suffering from lacking information, which is to say that a single value describing 10 s of steering was simply not sensitive or descriptive enough to portray the steering wheel movements. Predicting only four lane departures in the entire test set also showed that the QNN had detected the most common event, staying inside the lane. But for the rest of the attempts the possible explanations become numerous and vague due to both the QNNs' inherent indescribable operation and the results' lack of trends or other drawable conclusions. Increasing the input from one to ten points (for a 10 s bin) in Attempt 2 had an expected positive effect on the predictions (with correct predictions in 55 % of cases), because the input could now contain more information about the variations in the steering. The predicted number of lane departures also reached 71 % of the amount of measured lane departures, which showed that increasing the number of inputs also forced the QNN to make predictions other than no lane departures. The specificity of Attempt 2 was at a reasonable level (80 %), but the sensitivity was very low, only 31 %, signifying that a prediction of upcoming lane departures was made only in 31 % of cases where there was an actual lane departure. That was clearly not satisfactory. Considerations were made about several factors that could be changed to achieve better performance: It is possible that the training set contained too many signals with no lane departures which would cause the QNN to produce excessive amounts of predictions of no lane departures (2.1

and 3.2, Fig. 17.), the scaling of both input and output might have been unsuitable (2.3.2.2), the backpropagation learning might have caused the QNN to converge to a local minimum (2.1), the online learning might have been too easily influenced by each prediction causing the error gradient to fluctuate too much to converge at all, the structure of the QNN might have been unsuitable (section 1), and finally, the chosen steering metric might have been suboptimal (2.3.2). None of the raised questions had certain answers, but speculations were made. The training set did contain hundreds of bins containing lane departures, and because the average driver does not drive outside the lane once every 10 s, the training set should contain comparably more 10 s bins without lane departures than bins with lane departures. Choosing the training set randomly should also ensure that it was representative (2.3.2.3). A learning rate of 0.6 should be high enough to cause sufficient exploration of the parameter space to avoid converging to a local minimum, or at least not to the least optimal local minimum (2.1). The scaling of the output should utilise most of the dynamic range of the QNN, but the rescaling of the steering metric might have been insufficient to cover most of the dynamic range (2.3.2.2). However, even during the steering prediction, which rescaled the steering to less than half of the rescaling of the steering metric, there were still several occurrences in which the rescaled signal exceeded the range of the QNN. The superposition of states and parallelism of the qubit neurons should, at least theoretically, be able to compensate for the limitations of the online learning. For the remaining considerations, that is, the structure of the network and having a suboptimal steering metric, no conclusion could be reached. Therefore the steering metric was changed for Attempt 3 and the QNN was expanded by five neurons in each hidden layer to allow more complexity. The new steering metric, the integral of the steering, should have a higher correlation with the position in the lane because the truck acts as a powerful low-pass filter (2.3.2.1). The amplitude of the steering movements does not contain any temporal information, and anyone who has sat in a car knows that the longer the steering wheel is kept at an angle, the farther away the car moves. The integral of the steering captures this temporal information and should therefore be a suitable steering metric. Unfortunately, as Table 2 shows, the percentage of correct predictions decreased. The rise in sensitivity and drop in specificity from Attempt 2 does indicate that more lane departures were predicted, but incorrectly (2.3.2.4 and 3.2). Removing the average offset in root-mean-square error

(RMSE) for all drivers did raise the percentage of correct predictions above that of the previous attempt, but the sensitivity of 12 % after the offset correction did indicate that far too few lane departures were predicted. The histogram (Fig. 17) confirmed the problem with having too many predictions of no lane departures, prompting the selection of a training set containing more lane departures for Attempt 4. Even having more lane departures in the training set did not improve the QNNs ability to predict lane departures.

A final comment should be made about the lane departure prediction — the RMSE as a function of time awake does show something of a wave-like behaviour. It is a small effect, but it does resemble the measured lack of vigilance [44] (as measured with the Psychomotor Vigilance Task [47]). It is natural to assume that it would be more difficult for the QNN to predict successfully as the driving becomes more erratic with increased sleepiness, but for some reason there is no such effect in the prediction of steering. This discrepancy also speaks to the fact that it is easier for a QNN to predict a signal from previous parts of the same signal than it is to predict something entirely different.

Future efforts into Quantum Neural Network prediction of hazardous driving behaviour should aim to find a suitable method to predict lane departures and to achieve a longer prediction horizon. Based on the results obtained in this thesis, the first steps towards that goal would be to attempt to find a suitable steering metric for lane departure prediction and to attempt training on a smaller training set with many lane departures. Predicting steering several time steps ahead could also be attempted by selecting the predicted time step to be $t + n$, with $n > 1$. The time-delayed feed-forward network structure is also simple compared to other network structures, so selecting a different network structure could improve both steering and lane departure prediction. It is possible that a network structure suited for pattern recognition could be beneficial for the lane departure prediction. Unfortunately, the selection possibilities for network structures, parameters, steering metrics, and training sets are endless, but because the current QNN worked so well for steering prediction it could be possible that selecting a different time step or adding neurons to the output layer (for different time steps) would suffice, and because lane departure prediction in its current form resemble a regression or classification task more than time series prediction, perhaps choosing a network structure suitable for regression, e.g. a General Regression NN [33], could lead to improvements.

5 Conclusions

Predicting the steering wheel signal one time step ahead, 0.1 s, was achieved with a relative error of 0.4 % using a time-delayed feed-forward Quantum Neural Network (QNN) with a 15-40-20-1 structure. The training set was only 15 % of the data, and it contained signals from only the drivers driving in daylight conditions. Despite this restriction on the training set, the prediction performance was still stable across drivers and time awake. Predicting lane departures from the steering wheel signal, on the other hand, was not successful to a useful degree (best result was correct predictions in 55% of cases, sensitivity 31 % and specificity 80 %) and the main reasons for failure seemed to be the lack of a sensitive enough steering metric to truly capture the variations in the steering wheel signal and a tendency of the QNN to predict no lane departures, probably due to the training set. The prediction performance remained the same even when the training set was selected to contain more lane departures, suggesting that the selected steering metrics, maximum peak-to-peak amplitude and integral of the steering wheel of the previous ten 1 s segments, were not descriptive enough to allow the QNN to detect traits during learning.

6 References

[1] L. Rokach and O. Maimon, *Data mining with decision trees: theory and applications*: World scientific, pp. 77-84, 2014.

[2] F. Chen, P. Deng, J. Wan, D. Zhang, A. V. Vasilakos, and X. Rong, "Data mining for the internet of things: literature review and challenges," *International Journal of Distributed Sensor Networks,* vol. 2015, p. 12, 2015.

[3] L. Wang, *Support vector machines: theory and applications* vol. 177, pp. 1-48: Springer Science & Business Media, 2005.

[4] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine learning,* vol. 20, pp. 197-243, 1995.

[5] D. E. Goldberg, *Genetic algorithms*: Pearson Education India, pp. 59-88, 2006.

[6] J. Wang, O. K. Ersoy, M. He, and F. Wang, "Multi-offspring genetic algorithm and its application to the traveling salesman problem," *Applied Soft Computing,* vol. 43, pp. 415-423, 2016.

[7] S. Haykin and N. Network, "A comprehensive foundation," *Neural networks,* vol. 2, 2004.

[8] E. Guresen, G. Kayakutlu, and T. U. Daim, "Using artificial neural network models in stock market index prediction," *Expert Systems with Applications,* vol. 38, pp. 10389-10397, 2011.

[9] E. M. Azoff, *Neural network time series forecasting of financial markets*: John Wiley & Sons, Inc., 1994.

[10] M. Marcellino, J. H. Stock, and M. W. Watson, "A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series," *Journal of econometrics*, vol. 135, pp. 499-526, 2006.

[11] G. E. Box and G. M. Jenkins, *Time series analysis: forecasting and control, revised ed*: Holden-Day, 1976.

[12] M. Lydia, S. S. Kumar, A. I. Selvakumar, and G. E. P. Kumar, "Linear and non-linear autoregressive models for short-term wind speed forecasting," *Energy Conversion and Management*, vol. 112, pp. 115-124, 2016.

[13] A. Misiorek, S. Trueck, and R. Weron, "Point and interval forecasting of spot electricity prices: Linear vs. non-linear time series models," *Studies in Nonlinear Dynamics & Econometrics*, vol. 10, 2006.

[14] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of econometrics*, vol. 31, pp. 307-327, 1986.

[15] R. C. Garcia, J. Contreras, M. Van Akkeren, and J. B. C. Garcia, "A GARCH forecasting model to predict day-ahead electricity prices," *IEEE transactions on power systems*, vol. 20, pp. 867-874, 2005.

[16] R. Mendes, A. Paiva, R. Peruchi, P. Balestrassi, R. Leme, and M. Silva, "Multiobjective portfolio optimization of ARMA–GARCH time series based on experimental designs," *Computers & Operations Research*, vol. 66, pp. 434-444, 2016.

[17] Y. Zhang and T. Choudhry, "Forecasting the Daily Time-Varying Beta of European Banks During the Crisis Period: Comparison Between GARCH Models and the Kalman Filter," *Journal of Forecasting*, 2016.

[18] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50-64, 2014.

[19] A. Sharma, R. Mehrotra, J. Li, and S. Jha, "A programming tool for nonparametric system prediction using Partial Informational Correlation and Partial Weights," *Environmental Modelling & Software*, vol. 83, pp. 271-275, 2016.

[20] L. Györfi, W. Härdle, P. Sarda, and P. Vieu, *Nonparametric curve estimation from time series* vol. 60: Springer, 2013.

[21] G. Biau, K. Bleakley, L. Györfi, and G. Ottucsák, "Nonparametric sequential prediction of time series," *Journal of Nonparametric Statistics,* vol. 22, pp. 297-317, 2010.

[22] X. B. Chen, J. Gao, D. Li, and P. Silvapulle, "Nonparametric Estimation and Forecasting for Time-Varying Coefficient Realized Volatility Models," *Journal of Business & Economic Statistics,* pp. 1-39, 2016.

[23] M. H. Pesaran, D. Pettenuzzo, and A. Timmermann, "Forecasting time series subject to multiple structural breaks," *The Review of Economic Studies,* vol. 73, pp. 1057-1084, 2006.

[24] R. Wang and S. M. Lukic, "Review of driving conditions prediction and driving style recognition based control algorithms for hybrid electric vehicles," in *2011 IEEE Vehicle Power and Propulsion Conference*, 2011, pp. 1-7.

[25] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics,* vol. 5, pp. 115-133, 1943.

[26] D. O. Hebb, *The organization of behavior: A neuropsychological approach*: John Wiley & Sons, 1949.

[27] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review,* vol. 65, p. 386, 1958.

[28] R. Sitte and J. Sitte, "Neural networks approach to the random walk dilemma of financial time series," *Applied Intelligence,* vol. 16, pp. 163-171, 2002.

[29] E. Chng, S. Chen, and B. Mulgrew, "Gradient radial basis function networks for nonlinear and nonstationary time series prediction," *IEEE transactions on neural networks,* vol. 7, pp. 190-194, 1996.

[30] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks,* vol. 5, pp. 240-254, 1994.

[31] K.-i. Kamijo and T. Tanigawa, "Stock price pattern recognition-a recurrent neural network approach," in *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, 1990, pp. 215-221.

[32] B. B. Nasution and A. I. Khan, "A hierarchical graph neuron scheme for real-time pattern recognition," *IEEE transactions on neural networks,* vol. 19, pp. 212-229, 2008.

[33] D. F. Specht, "A general regression neural network," *IEEE transactions on neural networks,* vol. 2, pp. 568-576, 1991.

[34] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern recognition,* vol. 35, pp. 2279-2301, 2002.

[35] A. A. Ezhov and D. Ventura, "Quantum neural networks," in *Future directions for intelligent systems and information sciences,* ed: Springer, 2000, pp. 213-235.

[36] N. Kouda, N. Matsui, H. Nishimura, and F. Peper, "Qubit neural network and its learning efficiency," *Neural Computing & Applications,* vol. 14, pp. 114-121, 2005.

[37] X. Li, C.-T. Cheng, W.-C. Wang, and F.-Y. Yang, "A study on sunspot number time series prediction using quantum neural networks," in *Genetic and Evolutionary Computing, 2008. WGEC'08. Second International Conference on*, 2008, pp. 480-483.

[38] D. Ventura and T. Martinez, "Quantum associative memory with exponential capacity," in *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, 1998, pp. 509-513.

[39] H. Häkkänen and H. Summala, "Sleepiness at work among commercial truck drivers," *Sleep,* vol. 23, pp. 49-57, 2000.

[40] A. Vadeby, Å. Forsman, G. Kecklund, T. Åkerstedt, D. Sandberg, and A. Anund, "Sleepiness and prediction of driver impairment in simulator studies using a Cox proportional hazard approach," *Accident Analysis & Prevention,* vol. 42, pp. 835-841, 2010.

[41] P.-Å. Fröberg, "Volvo Cars introduces new systems for alerting tired and unconcentrated drivers," ed. Press Information 2007-08-28: Volvo Car Corporation, 2007.

[42] R. Mahajan, "Hybrid quantum inspired neural model for commodity price prediction," in *Advanced Communication Technology (ICACT), 2011 13th International Conference on*, 2011, pp. 1353-1357.

[43] W. Sun, Y. J. He, and M. Meng, "A novel quantum neural network model with variable selection for short term load forecasting," in *Applied Mechanics and Materials*, 2010, pp. 612-617.

[44] P. Forsman (ed.), M. Sandström, A. Holmström, E. Lampsijärvi, G. Maconi, S. Ahmadzai, E. Haeggström, "Arrive Alive: Detect and Predict Lane Departures from the Steering Wheel Signal," Unigrafia, Helsinki ISBN 978-951-51-1860-8 (hardcover), ISBN 978-951-51-1861-5 (pdf), 2015.

[45] S. S. Mukherjee, R. Chowdhury, and S. Bhattacharyya, "Image restoration using a multilayered quantum backpropagation neural network," in *Computational Intelligence and Communication Networks (CICN), 2011 International Conference on*, 2011, pp. 426-430.

[46] A. A. Borbély, "A two process model of sleep regulation," *Human Neurobiology,* 1982.

[47] M. Basner and D. F. Dinges, "Maximizing sensitivity of the psychomotor vigilance test (PVT) to sleep loss," *Sleep,* vol. 34, pp. 581-591, 2011.

[48] D. G. Altman and J. M. Bland, "Diagnostic tests. 1: Sensitivity and specificity," *BMJ: British Medical Journal,* vol. 308, p. 1552, 1994.