Journal of Physics: Conference Series **2431** (2023) 012103 doi:10.1088/1742-6596/2431/1/012103

Estimating unknown qubit phase under telegraph noises using recurrent neural network

Suttavee Rojanasirivanit[1], Poramet Pathumsoot and Areeya Chantasri[2]

Optical and Quantum Physics Laboratory, Department of Physics, Faculty of Science, Mahidol University, Bangkok, 10400, Thailand

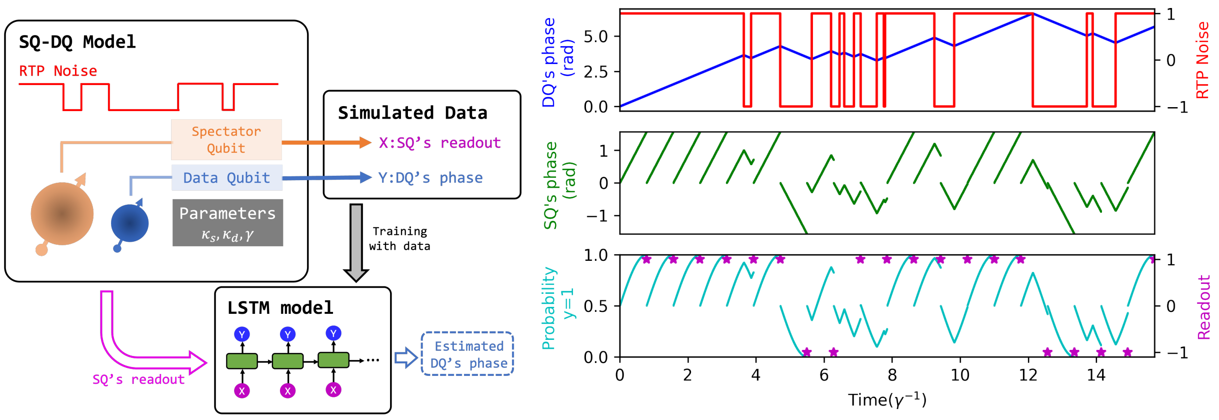
E-mail: [1]suttavee.roj@student.mahidol.ac.th; [2]areeya.chn@mahidol.ac.th

Abstract. We consider a system of qubits affected by an environmental random-telegraph process (RTP) noise and investigate the recently proposed idea of using a spectator qubit (SQ) as a noise probe to remove noise effects from a data qubit (DQ). By measuring the SQ at various times with particular measurement bases, their measurement readouts can be used to extract the information of the noise effects. In this work, we propose the use of recurrent neural network (RNN), with the Long short-term memory (LSTM) layers, to process the SQ's readouts and train the model to estimate the errors occurred in the DQ. We present numerical results of the DQ's and SQ's phases and show that the LSTM can estimate the unknown DQ's phase with the estimation accuracy depending on the SQ's noise sensitivity.

1. Introduction

Decoherence from environmental noises is one of the major obstacles to achieve full-capability quantum computing. The noises can randomly disturb qubits' phases and effectively result in losing the quantum coherence, which is the important resource for computation. From the recent work of [1,2], a decoherence-mitigation technique was proposed using data and spectator qubits, where the spectator qubit (SQ) was used as a probe to measure the dephasing random-telegraph noise, i.e., a type of low-frequency noises found solid-state qubit experiments [3], and the measurement readouts was used in correcting noise errors in the data qubit (DQ). Because of quantum mechanics, measuring the SQ will only provide stochastic outcomes depending on the phase of the SQ with different probabilities and one cannot directly calculate the actual phase errors from the measurement outcomes. In [1,2], a Bayesian technique was proposed to estimate the DQ's phase errors from the SQ's measurement results, showing that their error correction technique could lead to a significant decoherence reduction. However, the Bayesian technique can be too model-specific to be implemented in real experiments, especially when unexpected events or uncontrolled parameters occur outside of the constructed theoretical model.

Therefore, in this work, we explore an alternative approach using a recurrent neural network (RNN) to help estimate the correct phase of DQ, only by supervised training the network with acquired data. As shown in figure 1a, we use the SQ's measurement readouts as the machine input and the DQ's phase errors as the output. We choose to work with RNN, specifically with the multi-layer long short-term memory (LSTM) [4], because it is particularly designed for the time-sequence data. To get the best trained LSTM model, we also perform a hyper-parameter



(a) (b)

Figure 1. (a) Diagrams show how we trained the LSTM model to estimate the DQ's phase from the SQ's measurement readouts. We first simulated the SQ-DQ data and used the SQ's readouts and DQ's phases as the LSTM input and output respectively. (b) A single realization of simulated data SQ-DQ. This includes a RTP noise (red line) randomly generated with jump rate $\gamma_{\uparrow,\downarrow}$, a trajectory of the DQ's phase (blue line) associated with the noise and the SQ's phase (green line) also associated with the same noise but with discontinuity caused by the measurements (the phase is reset to $\phi_\mathrm{s} = 0$ after each measurement). In the bottom panel, we show measurement results from measuring the SQ in $\hat{\sigma}_y$ basis (pink stars) and their probabilities calculated from the SQ's phase (see text).

tuning to search over appropriate parameters for the LSTM model. Our numerical results show that the accuracy is not only dependent on parameters of the LSTM model, but also strongly dependent with the noise sensitivity of the SQ probe.

2. DQ-SQ model and trajectory simulation

Our model includes one DQ experiencing dephasing from the RTP noise and one SQ to probe the noise in time. The RTP noise is characterized by a signal $z(t)$ that alternates between two states, denoted by values ±1, where the switching rates from 1 to -1 and -1 to 1 are given by $\gamma_\downarrow$ and $\gamma_\uparrow$, respectively. Let us assume that the Hamiltonian of the DQ that represents how the noise affects the qubit is given by

$$\hat{H}_\mathrm{d} = \frac{\kappa_\mathrm{d}}{2} z(t)\hat{\sigma}_z^\mathrm{d}, \tag{1}$$

where κ_d is the DQ's noise sensitivity and $\hat{\sigma}_z^\mathrm{d}$ is the z-pauli matrix of the DQ. The noise sensitivity is a parameter that describes how strongly a qubit is sensitive to noise, which can be related to the physical distance between the qubit and the source of noise or their coupling strength. Using the Schrödinger equation with the qubit Hamiltonian in equation (1), one can find that the DQ's phase evolution in the Bloch sphere can be written as a function of an accumulated noise during the time interval of interest. That is

$$\phi_\mathrm{d}(t) = \kappa_\mathrm{d} X(t_0, t); \text{ (assuming } \hbar = 1). \tag{2}$$

where $X(t_0, t) = \int_{t_0}^{t} z(t')\mathrm{d}t'$ is the accumulated noise. The RTP noise randomly rotates the DQ's phase around the z-axis which results in the qubit dephasing [5].

In order to sense the noise, we introduce an additional qubit called a SQ, which we assume that the SQ's phase rotates randomly in the same way as the DQ but located near the DQ, but no interaction between them with a different noise sensitivity. That is, the SQ's phase is $\phi_{\rm s}(t) = \kappa_{\rm s} X(t_0, t)$. We also consider the regime

$$\kappa_{\rm d} \ll \gamma_{\uparrow,\downarrow} \ll \kappa_{\rm s}, \tag{3}$$

where the DQ's noise sensitivity $\kappa_{\rm d}$ is small compared to the RTP flip-rates $\gamma_{\uparrow,\downarrow}$ and the SQ's noise sensitivity $\kappa_{\rm s}$ is larger than $\gamma_{\uparrow,\downarrow}$, making the SQ capable of tracking the RTP noise. The SQ is then measured at different times in the y-basis (i.e., with a measured observable $\hat{\sigma}_{\pi/2} = \cos\frac{\pi}{2}\hat{\sigma}_x + \sin\frac{\pi}{2}\hat{\sigma}_y$), giving the readout probabilities,

$$P(y = +1|\phi_{\rm s}) = \cos^2[(\pi/2 - \phi_{\rm s})/2], \tag{4a}$$

$$P(y = -1|\phi_{\rm s}) = 1 - \cos^2[(\pi/2 - \phi_{\rm s})/2] \tag{4b}$$

see blue curves in the bottom panel of figure 1b. The measurement readouts can be used in estimating the DQ's phase error caused by the noise. Since the SQ's state collapses to one of the observable eigenstates due to the measurement, we need to reset its phase to $\phi_{\rm s} = 0$ in order to probe the noise in the next time period.

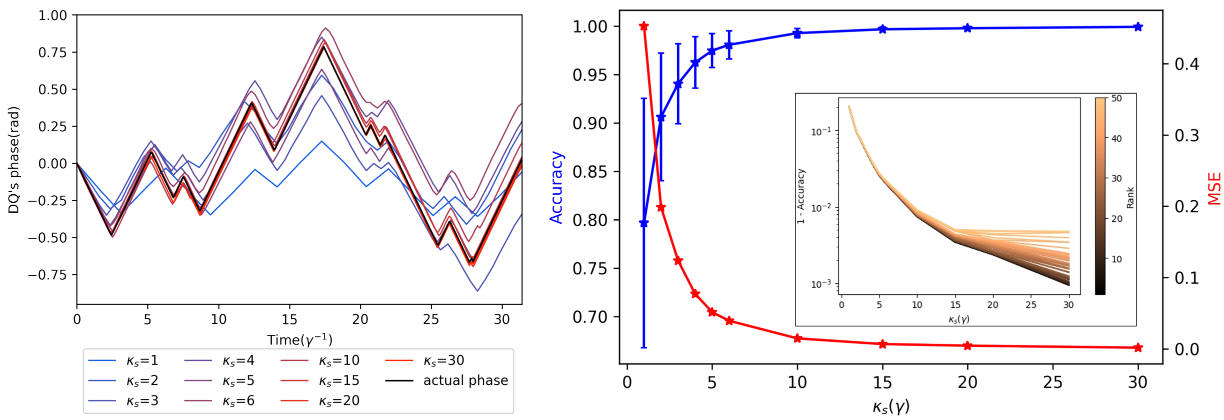
In this work, we use the supervised learning to predict the DQ's phase error. We first simulate trajectory data to be used in the machine training using Python programming language. We show in figure 1b an example of the SQ-DQ data realization, including a RTP noise, the DQ and SQ phases, and the SQ's measurement readouts (see caption for more detail). In our simulation, we set $\gamma_\uparrow = \gamma_\downarrow = \gamma = 1$, so that we can use γ^{-1} as our time unit. We set $\kappa_{\rm d} = 0.2$ which is on order of magnitude smaller than γ. The total time is chosen as $T = 10\pi$ and the interval time between measurements is $\tau = \pi/2\kappa_{\rm s}$, which is the near optimal value proved in [2]. We chose to vary the noise sensitivity of SQ such that $\kappa_{\rm s} = 1$, 2, 3, 4, 5, 6, 10, 15, 20 and 30. The data we used in training the machine includes the SQ's measurement readouts as the input and the DQ's phase as the output. Our simulated data are divided into two sets, for the training dataset and the test dataset, which are 8192 and 1024 realizations, respectively.

3. DQ's phase estimation using LSTM

We have used Pytorch [6], a Python open source for machine learning framework, for our LSTM model. The LSTM model specifically is designed to predict a sequence of output data, which in this case is the DQ's phases at various times when the measurements were performed on the SQ. The LSTM model learnt through the training dataset, where its learning algorithm used Adam (adaptive moment estimation) [7] as optimizer. The algorithm is designed to adjust the weights and biases of the model to minimize the loss function, which is the mean square error (MSE). The parameters related to the training process are the Adam learning rate (LR), the batch size, and the number of epochs. We also performed a hyperparameter tuning, exploring how different parameters yield different performance in the DQ's phase estimation. We defined the performance using the phase accuracy,

$$\text{accuracy} = \left|\langle e^{i(\phi_{\rm r}(T) - \phi_{\rm e}(T))}\rangle\right|, \tag{5}$$

where $\phi_{\rm r}(T)$ is a real final DQ's phase in each realization from the dataset and $\phi_{\rm e}(T)$ is a LSTM estimate of the final DQ's phase. The parameters we explored were: the SQ's noise sensitivity $\kappa_{\rm s} = 1$, 2, 3, 4, 5, 6, 10, 15, 20 and 30, the LR values being 0.1, 0.01 and 0.001, and the number of epochs being 5, 10, 50, 100 and 500. For the part of the multi-layer LSTM, we searched for the hidden size of 1, 32 and 128, and the number of layers being 1, 2 and 3. There are other



(a) **(b)**

Figure 2. (a) The DQ's phase as a function of time. The black curve is a true DQ's phase from the numerical simulation, whereas the colored curves are the estimated DQ's phase using our LSTM model with different values of $\kappa_{\rm s}$. (b) The performance of LSTM model shown as the accuracy (blue line with error bars) and the MSE (red line with error bars). We note that the error bars in the accuracy and MSE plots get smaller when $\kappa_{\rm s}$ get larger (stronger SQ's noise sensitivity). This indicates that there is more information about the unknown noise gained from measuring the SQ with larger sensitivivity. The inset shows $1 -$ accuracy for the top 50 sets of LSTM's hyperparameters in the log-scale.

parameters that we set as constant, such as the dropout rate at 0.1 and the bidirectionality set to true. The quality of the test dataset can then be calculated using the same accuracy equation (5) for all of the parameter combinations. We also calculated the mean square error (MSE) of the final DQ's phases shown in figure 2b, which can be calculated as MSE $= \langle(\phi_{\rm r}(T) - \phi_{\rm e}(T))^2\rangle$.

4. Result and discussion

We show in figure 2a an example of the DQ's phase estimation by LSTM for different values of $\kappa_{\rm s}$. The black line is a true DQ's phase and the other lines are the DQ's phases estimated by LSTM. We can see that the DQ's phase estimation results are close to the real phase when the SQ's noise sensitivity $\kappa_{\rm s}$ is high. From the hyper hyperparameter tuning, we found that the accuracy and the MSE of LSTM's estimates strongly depend on the DQ's noise sensitivity $\kappa_{\rm s}$ shown in figure 2b and insignificantly affected by the other LSTM's parameters. As expected from the results in figure 2b, we show in figure 2b that the accuracy increases with the value of $\kappa_{\rm s}$ (reaching the value of 1 when $\kappa_{\rm s} \approx 30$ and the MSE decreases to 0 when $\kappa_{\rm s}$ gets large). By varying other LSTM's parameters, we found that they did not significantly affect the accuracy nor the MSE, as we can only see the difference from the log-scale plots in the inset of figure 2b. However, we also found that the model could fail to train the data when the number of epochs is too large (≈ 500 epochs) or when the LR is too high (≈ 0.1).

5. Conclusion

We have investigated the SQ-DQ model, where qubits were affected by the environmental RTP noise. The SQ was used as a noise probe and its readouts were used in correcting the noise effects in the DQ. We implemented the RNN technique with LSTM layers and trained our model with numerically generated SQ's measurement readouts and the DQ's errors. We have found that our LSTM model can estimate the actual DQ's phases, with a good agreement between the phase estimation and the actual values, especially when the SQ's sensitivity κ_{s} is large enough. For the future work, the LSTM model can still be developed or adjusted to improve its performance for the phase estimation.

Acknowledgments

This work was supported by National Research Council of Thailand (NRCT) grant N41A640120. SR also acknowledges the support from Sri Trang Thong scholarship, Faculty of Science, Mahidol University.

References

[1] Song H, Chantasri A, Tonekaboni B and Wiseman H M 2022 arXiv:2205.12567
[2] Tonekaboni B, Chantasri A, Song H and Wiseman H M 2022 arXiv:2205.12566
[3] Bergli J, Galperin Y M and Altshuler B L 2009 *New J. Phys.* **11** 025002
[4] Hochreiter S and Schmidhuber J 1997 *Neural Comput.* **9** 1735–1780
[5] Culcer D, Hu X and Das Sarma S 2009 *Appl. Phys. Lett.* **95** 073102
[6] Paszke A *et al* 2021 *PyTorch: An Imperative Style, High-Performance Deep Learning Library* (Red Hook, NY, USA: Curran Associates Inc.)
[7] Kingma D P and Ba J 2014 arXiv:1412.6980