A Machine Learning Approach for Automated Fine-Tuning of Semiconductor Spin Qubits

Julian D. Teske,1 Simon Humpohl,1 René Otten,1 Patrick Bethke,1 Pascal Cerfontaine,1 Jonas Dedden,1 Arne Ludwig,2 Andreas D. Wieck,2 and Hendrik Bluhm1

1) JARA-FIT Institute for Quantum Information, Forschungszentrum Jülich GmbH and RWTH Aachen University, 52074 Aachen, Germany
2) Lehrstuhl für Angewandte Festkörperphysik, Ruhr-Universität Bochum, 44780 Bochum, Germany

(Dated: 8 January 2019)

While spin qubits based on gate-defined quantum dots have demonstrated very favorable properties for quantum computing, one remaining hurdle is the need to tune each of them into a good operating regime by adjusting the voltages applied to electrostatic gates. The automation of these tuning procedures is a necessary requirement for the operation of a quantum processor based on gate-defined quantum dots, which is yet to be fully addressed. We present an algorithm for the automated fine-tuning of quantum dots, and demonstrate its performance on a semiconductor singlet-triplet qubit in GaAs. The algorithm employs a Kalman filter based on Bayesian statistics to estimate the gradients of the target parameters as function of gate voltages. The algorithm’s design is focused on the reduction of the number of required measurements. We demonstrate the ability experimentally to change the operation regime of the qubit within 3 to 5 iterations, corresponding to 10 to 15 minutes of lab-time.

Keywords: Kalman filter, automated tuning, semiconductor spin qubits, quantum computation, quantum dots, machine learning

Gate-defined quantum dots are a promising realization of qubits for the construction of a universal quantum computer1. Semiconductor spin-qubits have been realized and intensively studied in the past decades2–14, with tremendous progress in gate15 and measurement16 fidelities, and also in spin-cavity coupling17,18. A central starting point for the operation of qubits based on gate-defined quantum dots is the so-called tuning of the system, i.e. the procedure for identifying the voltages that need to be applied to the electrostatic gates to capture and to tunnel-couple individual electrons. Tuning by a human operator is very time consuming and will be impractical for multi-qubit systems with more than a handful of qubits. Efficient tuning is particularly pertinent for gate-defined quantum dots because of the large number of tunable dot parameters and gate voltages to control them, but is also relevant for other systems. For qubits based on quantum dots, tuning is a two-step procedure, which comprises both the formation of quantum dots and their depletion into the few electron regime – which we refer to as coarse-tuning – and, subsequently, the adjustment of parameters which define the operation conditions of the qubit such as the exchange interaction or the initialization time. This procedure is referred to as fine-tuning of the qubit19.

The initial coarse-tuning of gate-defined quantum dots relies mainly on the recognition of certain features in charge stability diagrams (CSD), which reflects the dot occupancy as a function of gate voltages2,4. Some recent attempts to automate this procedure use image processing tools like the Gabor filter and template matching7 to recognize quantum dot regimes. Others feature machine learning techniques like convolutional neural networks20 to quantify characteristics of CSDs by their similarity to simulated or measured reference CSDs. The subsequent fine-tuning is complicated by the nonlinear dependence of the tunnel couplings on the gate voltages. Furthermore, gate-defined quantum dots exhibit a strongly coupled response, meaning that the voltage applied to each electrode has a considerable influence on several chemical potentials and tunnel couplings. A method to disentangle the various parameters to linear order is the use of "virtual gates", i.e. linear combinations of gate voltages, each of which predominantly affects a single qubit parameter21.

Here, we propose and experimentally demonstrate an algorithm for the fine-tuning of qubits based on gate-defined quantum dots which exploits machine learning for improving the efficiency of the tuning procedure. Our algorithm combines a gradient-based optimization algorithm with an adapted implementation of a Kalman filter. The latter allows efficient tracking the gradients of the parameters in the multidimensional voltage space. Each measurement of the parameters at a new point in the voltage space is compared to the previous measurement and used to update the gradient. This approach results in a full automation of the tuning procedure, which requires only 3 to 5 iterations to change a tunnel coupling by a factor 2, corresponding to 10-15 minutes of lab-time for the parameter extraction procedures currently used.

For the experimental realization we use a double quantum dot in a AlGaAs/GaAs heterostructure in the same experimental setup as Botzem et al.19 and build on the parameter-extraction procedures developed there. The gate layout is shown in Fig. 1. The double quantum dot is designed to be used as singlet-triplet qubit, and features a neighboring sensing-dot for qubit readout based on spin-to-charge conversion. All measurements are based on

1. Semiconductor Spin Qubits
2. Machine Learning Approach
3. Automated Fine-Tuning
4. Quantum Computation
5. Quantum Dots
6. Kalman Filter
7. Bayesian Statistics
8. Gate-Defined Quantum Dots
9. Charge Stability Diagrams
10. Image Processing
11. Gabor Filter
12. Template Matching
13. Convolutional Neural Networks
14. Virtual Gates
15. Nonlinear Dependence
16. Tunnel Couplings
17. Chemical Potentials
18. Linear Combinations
19. Coarse-Tuning
20. Fine-Tuning
21. Automatic Tuning
of the inter-dot transition in gate voltage space, of the inter-dot tunneling, characterized by the width qubit parameters that we want to tune are the strength are DC-coupled to an arbitrary waveform generator. The scans used to extract the parameters of interest and the chemical potentials are controlled with PA and PB. The gates marked in red are used for tuning the tunnel couplings. Gates T and N are designed to control the inter-dot tunnel coupling whereas SA and SB are meant to control the tunnel coupling to the electron reservoir.

RF-reflectometry and performed in a dilution refrigerator.

Although there are 9 DC-gates defining the quantum dots, only 4 of them are used to fine-tune the parameters, namely the gates N, T, SA and SB. The voltages on SD1, SP and SD2 are used to control the sensing dot, and the chemical potentials are controlled with PA and PB. The scans used to extract the parameters of interest are performed with the RF-gates RFA and RFB, which are DC-coupled to an arbitrary waveform generator. The qubit parameters that we want to tune are the strength of the inter-dot tunneling, characterized by the width of the inter-dot transition in gate voltage space, \( w \), and the time required to reload a singlet \( t_{sr} \). The latter characterizes the tunnel coupling between one of the dots and its neighboring electron reservoir and is measured via the dependence of the load efficiency on the corresponding waiting time in the reload operation. Tuning these two parameters is sufficient to obtain a fully operational singlet-triplet qubit. The tunnel coupling to the other lead is almost closed to allow for latching readout.

Fine-tuning \( w \) and \( t_{sr} \) to some target values \( w^* \) and \( t^*_{sr} \) is complicated by the fact that \( w \) and \( t_{sr} \) are non-linear functions of four different voltages. To optimize their values, we employ a quasi-Newton method, which is a gradient-based optimization algorithm and ideally requires only a few iterations to converge. In the \( k \)th iteration of the algorithm, the gate voltages are updated according to the following formula:

\[
\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} - \left( \mathbf{J}^{(k)} \right)^{-1} \Delta^{(k)},
\]

where \( \mathbf{v}^{(k)} = (v_N, v_{SA}, v_{SB}, v_T)^T \) is a vector containing the voltage configuration at step \( k \), \( \Delta^{(k)} = (t^*_{sr} - t^{*}_{sr}, w^{*} - w^{*})^T \) contains the distance of the measured parameter values at step \( k \) from the target ones, and \( \mathbf{J}^{(k)} \) is the Jacobian, with \( J^{(k)}_{ij} = \partial p_i / \partial v^{(k)} \) for \( j \in (N, SA, SB, T) \) and \( p_i \in (w, t_{sr}) \). \( (\mathbf{J}^{(k)})^{-1} \Delta^{(k)} \) is to be understood as solution of the corresponding system of linear equations with minimal euclidean norm.

Our tuning procedure combines the Gauss-Newton algorithm with the Kalman filter as shown in the UML diagram in Fig. 2. Each iteration begins with a new measurement of the parameters. If the measured values are already in the desired parameter range, the voltages are accepted as final state and the algorithm terminates. Otherwise, gradient updates are performed by a Kalman filter. The new gradients are then used by the Gauss-Newton algorithm Eq. (1) to calculate the new voltages to be applied to the gates. The algorithm mimics the learning process of a human operator, who changes the voltages according to a certain expectation of how this will affect the qubit parameters, performs measurements to verify the result, and uses the information gained from these measurements to refine the understanding of the behavior of the system.

Since the Gauss-Newton algorithm is based on a linearization of the dependence of \( w \) and \( t_{sr} \) on the applied voltages, we restrict the maximal voltage change to 10 mV, because the linear approximation by the gradient becomes less accurate the larger the steps are. On the other hand, the restriction should be chosen as large as possible because the ratio of the physical changes to the fluctuations of the parameters due to noise and disorder grows with the step size. In addition, smaller steps will...
lead to a larger number of steps for substantial changes. After the new voltages are set, the contrast in the sensing dot is optimized with the gates SD1 and SD2 and the chemical potential is corrected with the plunger gates PA and PB (see supplementary material).

Since the parameters are in general strongly nonlinear functions of the gate voltages, a naive implementation of a gradient-based optimization algorithm would require the Jacobian $J^{(k)}$ to be remeasured by finite differences in every iteration. This is a time-consuming operation because it requires many measurements at different voltages. To avoid this issue, our algorithm takes a machine-learning approach and uses a Kalman filter to estimate the Jacobian at step $k$ using the knowledge of $J^{(k-1)}$ and the information drawn from a single set of measurements.

The Kalman filter is an algorithm designed to estimate a system of normally distributed random variables given noisy measurements at discrete steps $k$ and knowledge of their dynamics. We use an adapted version of the Kalman filter specific to our needs, which are the estimation of the gradient $g_i^{(k)} = \langle \frac{\partial p^{(k)}}{\partial v_i} \rangle$ and the corresponding covariance matrix $C_{i,j}^{(k)} = \langle (\frac{\partial p^{(k)}}{\partial v_i} - g_i^{(k)}) (\frac{\partial p^{(k)}}{\partial v_j} - g_j^{(k)}) \rangle$ of a parameter $p$ in iteration $k$ as functions of the control voltages $v_i,j$ for $i,j \in \{N, SA, SB, T\}$. Expectation values $\langle \cdot \rangle$ refer to the distribution of the uncertain parameters being tracked. For each $p \in \{v, t\}$, we use a Kalman filter to track the estimation of its gradient described by $g$ and $C$. Each instance of the Kalman filter approximates a row in the Jacobian $J$ by the mean of its distribution. The initial values $g^{(0)}$ and $C^{(0)}$ are measured by finite differences as discussed in the supplementary material.

In each iteration, the Kalman filter uses the information gained from a new measurement of the parameters $p^{(k)}$, to update the values of $g$ and $C$ according to the following formulas:

$$g^{(k)} = g^{(k-1)} + K^{(k)} (z^{(k)} - H^{(k)} g^{(k-1)}),$$

$$C^{(k)} = (I - K^{(k)} H^{(k)}) C^{(k-1)} + Q,$$

Here $H^{(k)}$ is an observation model (consisting in our application of 1 by 4 matrices) that maps the “state space” of the Kalman filter (i.e. the space of $g^{(k)}$) onto the measurement space (i.e. the space of $p^{(k)}$), with elements $H_{i,j}^{(k)} = v_i^{(k)} - v_j^{(k-1)}$. The product $H^{(k)} g^{(k-1)}$ then represents the predicted change in the parameter $p$ due to the change in $v$ at step $k$, which is compared in Eq. (2) to the measured change $z^{(k)} = p^{(k)} - p^{(k-1)}$. The matrix $K^{(k)}$ is the so-called Kalman gain

$$K^{(k)} = \frac{C^{(k-1)} H^{(k)} \, T}{(H^{(k)} C^{(k-1)} H^{(k)} \, T + \Delta z^{(k)} 2)},$$

which depends on both the uncertainty of our knowledge of the gradients represented by the covariance matrix $C^{(k)}$, and on the measurement uncertainty $\Delta z^{(k)} 2 = \delta p^{(k)} 2 + \delta p^{(k-1)} 2$, with $\delta p^{(k)}$ the error on the measurement of $p^{(k)}$ (see supplementary material). An inaccurate measurement has a very large $\Delta z^{(k)} 2$ and therefore

![FIG. 3. Experimental demonstration of the tuning algorithm. The iteration number $k$ is given by the number of cycles in the diagram shown in Fig. 2. (a) The parameters are plotted in red. The horizontal blue lines are the current set points and vertical blue lines mark changes in the set point. Different set points are counted by roman numerals. The parameters are tuned within 3 to 5 iterations into the desired range with an accuracy limited by parameter fluctuations due to statistical noise in the measurement. (b) Gradients of the parameters with respect to the voltages on the gates N, SA, SB, and T, tracked by the Kalman filter. The error bars are the square roots of the diagonal elements in the gradient’s covariance matrix, which are larger in the lower graph because the measurements of the inter-dot transition are generally less accurate. The gradients in region Ia and Ib show some differences although the set points are identical and the parameters are similar. This can be explained by the influence of the initial gradient estimation and different gate voltages as the system is underdetermined with two parameters controlled by four voltages. (Gate voltages and extension in the supplementary material.)](image-url)
a small gain. The information gained with the measurement not only contributes to updating the value of the gradient (see Eq. (2)), but it also reduces the covariance matrix $C^{(k)}$ by a factor determined by the Kalman gain (first term in Eq. (3)). The algorithm becomes Broyden’s method in the limit $\Delta z \to 0$. In addition, we include a fixed increased by the term $Q$, which accounts for the additional uncertainty related to our lack of knowledge how the gradient changes while changing voltages, i.e. of how $g^{(k)}$ deviates from $g^{(k-1)}$.

In our case, $Q$ is chosen heuristically with the constraint that the estimated increase in the uncertainty of the parameter evaluation, $H^{(k)}QH^{(k)}$, must be of the same order of magnitude as the measurement uncertainty $\Delta z^{(k)}$ for typical values of $H^{(k)}$. This ensures a reasonable Kalman gain $K^{(k)}$ (compare Eq. (4)) and hence change in the prediction $g^{(k)}$. Note that too small values for $Q$ lead to an almost constant $g$, which can cause slow convergence or oscillatory behavior of Eq. (1). Too large values on the other hand lead to abrupt changes in $g$, which may be problematic if individual measurements occasionally give wrong results, e.g., because the underlying fit does not converge (see supplementary material).

To test the algorithm, we cyclically changed the target parameters $(t_{sr}, w)$ in the sequence $(20 \text{ ns}, 270 \mu \text{V})-(70 \text{ ns}, 270 \mu \text{V})-(180 \mu \text{V})-(20 \text{ ns}, 180 \mu \text{V})-(20 \text{ ns}, 270 \mu \text{V})$. For each pair of set points, we performed 21 iterations regardless of the convergence. The resulting voltage changes are not necessarily cyclical as the system is underdetermined. The choice of the lower set point of $t_{sr}$ is limited by the bandwidth of the data acquisition hardware and the lower set point of $w$ is limited by temperature broadening. We chose the upper limits such that we restrict the range to values typically used in experiments. The data in Fig. 3(a) demonstrates that the parameters can be tuned individually within three to five steps with a total duration of 10 to 15 minutes. This time is determined by the measurement time, while the computational time is negligible. Further performance improvements could be achieved by optimizing the way we extract $w$ and $t_{sr}$.

The estimates of the gradients $g^{(k)}$ are plotted in Fig. 3(b). We see that abrupt changes in the gradient estimates mostly occur in the first steps of each region right after set points are changed. In the upper graph in Fig. 3(b), the absolute values of the elements $\partial t_{sr}/\partial V_{SB}$ and $\partial t_{sr}/\partial V_T$ are much larger than those of the other two elements because they form the tunnel barrier to the lead next to RFB (see Fig. 1), which is used for the electron exchange in the singlet reload mechanism. The gradient elements $\partial t_{sr}/\partial v_j$ for $j \in \{N, SA, SB, T\}$ are almost constant, indicating only weak effects due to non-linearity.

The gradient of the transition width in the lower graph in Fig. 3(b) is more complicated, presumably because all gates control the positions of the quantum dots and hence the inter-dot tunnel coupling and the transition width. The element $\partial w/\partial V_{SB}$ changes by a factor of 3 and the other elements change their sign during the experimental demonstration. These changes in the gradients confirm that the dependence of the qubit parameters on the control voltages is strongly non-linear, which indicates that tuning procedures based on pre-calibrated gradients would be inefficient and underpins the advantage of tracking the changes of the gradients during the tuning procedure.

To facilitate the adoption of this approach, we provide an implementation of the algorithm as a python package named $\text{q}tune^{28}$. It contains our implementation of the Kalman filter and more tools that simplify an automated fine-tuning program. The implementation complies with good software engineering practices by including a full documentation and unit tests with high coverage. A general interface makes the package adaptable to other setups.

In conclusion, we used the Kalman filter to construct a fully automated fine-tuning procedure. Thereby, we demonstrated the ability of the Kalman filter to be used in combination with a gradient based optimization algorithm to efficiently solve a non-linear optimization problem without re-measuring the gradient. Thus, the algorithm does not only save time and resources, but also provides valuable information about the qubit in form of the gradient of its characteristics as a function of voltages, which can be used for evaluating its tunability.

Further performance improvements could likely be achieved by choosing $Q$ depending on the size of the voltage step in each iteration. In comparison with other quasi-Newton methods like Broyden’s methods, our procedure offers the advantage of taking statistical errors into account. The resulting performance advantage is yet to be demonstrated by detailed benchmarks.

The algorithm has been tested on a semiconductor spin qubit in a AlGaAs/GaAs double quantum dot but could also be applied to other types of qubit, including any qubit based on gate defined quantum dots. Automated tuning procedures will not only be needed for operating quantum processors but can also be very valuable for the systematic characterization and optimization of qubit technology and reproducibility. In fact, the possibility to tune a qubit is a key criterion for its functionality, and is intimately related to the tuning procedure employed. Hence, “smart” algorithms can make any given qubit design more successful.

We thank F. Haupt for helpful input on this article and Robert P. G. McNeil for the fabrication of the sample. We acknowledge support by the Impulse and Networking Fund of the Helmholtz association, the Helmholtz Nano Facility (HNF) at the Forschungszentrum Jülich, the Deutsche Forschungsgesellschaft under grant BL 1197/2-1 and BL 1197/4-1 and the Excellence Initiative of the German federal and state governments. A. Ludwig and A. D. Wieck gratefully acknowledge support of DFG-TRR160, BMBF - Q.Link.X 16KIS0867, and the DFH/UFA CDFA-05-06.

\(^{28}\)D. Loss and D. P. DiVincenzo, Phys. Rev. A 57, 120 (1998).
Supplementary Material

The supplements contain an extension of the experimental data in section I and additional information on the tuning procedure in section II. More instructions for the employment of the Kalman filter can be found in section III, IV and V.

I. EXTENSION OF EXPERIMENTAL DATA

We include an extension of the data in Fig. 4 for completeness and to demonstrate the stability of the algorithm. The voltages shown in Fig. 5 are not similar for equal set points, which explains different noise levels at similar set points (e.g. for regions Ia and Ib).

II. SENSING DOT AND CHEMICAL POTENTIALS

The measurement of $t_{sr}$ and $w$ requires a certain contrast in the spin-to-charge conversion and the exact knowledge of the chemical potentials. Any time gate voltages on the DC-gates defining the double quantum dot are changed, we sweep either one or both of the voltages on the gates SD1 and SD2. The resulting CSD features Coulomb oscillations and we set the voltages of SD1 and SD3 to the steepest point on either side of the highest Coulomb peak to maximize the readout contrast.

The measurement of $t_{sr}$ and $w$ is performed by pulsing the RF-gates to different voltage points in the CSD of RFA and RFB. The positions of these is chosen relative to the charge transition lines and can be kept fixed by compensating any shift of the transition lines caused by changes in the chemical potentials.

In the step ”Set New Voltages” of the tuning process as described in Fig. 2, we use virtual gates - as discussed by Botzem et al. - to calculate a linear compensation on PA and PB to correct for the change in the chemical potential, caused by the change of the voltages on N, SA, SB and T. Subsequently, we correct for the non-linearity and inaccuracies in the compensation using an algorithm running in two loops. The inner loop tunes the sensing dot while the outer loop compensates shifts of the transition lines in the CSD. The outer loop employs the same Kalman-filter based tuning algorithm described in the main text by using the position of the charge transition lines in the CSD as the target parameters instead of $t_{sr}$ and $w$.

III. INITIAL CONDITIONS

In the initial step ($k = 0$), the gradient is measured several times by finite differences and $g^{(0)}$ and $C^{(0)}$ are calculated by averaging over these measurements. $C^{(0)}$ is a diagonal matrix having as elements the variances of the measured gradients. Even if only a rough estimate of $g^{(0)}$ is available, the algorithm can be adjusted by choosing $C^{(0)}$ in the same order of magnitude as the squares of the elements in $g^{(0)}$. In this case the gradient will be effectively remeasured in the tuning process while profiting from the guessed information in $g^{(0)}$.

IV. MEASUREMENT UNCERTAINTY

The largest issue to the stability of the tuning procedure during the experimental demonstration arises from statistical noise on the raw measurement data. To eval-
We apply a pulse while sweeping one pulse parameter $x$ and measuring the signal $y$ via RF-reflectometry. This yields a raw data set consisting of measurement points $(x_i, y_i)$ with $1 \leq i \leq n$. We fit a function $f_p(x)$ to extract the parameter $p$ being $t_{sr}$ or $w$.

We quantify the statistical noise on a parameter fit with the following procedure. After fitting $f_p$ to the data, we calculate the quadratic sum of the residuals

$$\sigma = \sum_i \left( \frac{f_p(x_i) - y_i}{\Theta_p} \right)^2.$$  

where $\Theta_p$ is a rescaling factor describing the sensitivity in the sensing dot. $\Theta_w$ is chosen as height of the inter-dot transition and $\Theta_{t_{sr}}$ is chosen as range of $(y_1, \ldots, y_n)$, i.e. both describe the response in the sensing dot signal to the transition of an electron within the double quantum dot.

$\sigma$ is averaged over many measurements before the tuning to calculate a reference value $\sigma_0$. The standard deviation of this set of measurements is denoted by $\delta p_0$ and we estimate the uncertainty due to statistical errors as $\delta p \approx \delta p_0 \sigma / \sigma_0$.

V. CHOICE OF $Q$

A good choice of the parameter uncertainty added in each step, $Q$, is crucial for the performance and stability of the Kalman filter as it controls the Kalman gain. We used for the singlet reload time $Q_{t_{sr}} = (0.075 \text{ns mV}^{-1})^2 \cdot I$ and for the transition width $Q_w = (0.3 \mu\text{V mV}^{-1})^2 \cdot I$. For an update step of maximal step size ($10 \text{mV}$) $H Q H^T = \text{diag}((0.75 \text{ns})^2, (3 \mu\text{V})^2)$ is of the same order of magnitude as typical values for the measurement noise ($\delta t_{sr}^2, \delta w^2$) as estimated from repeated measurements at fixed gate voltages. However, in most update steps the voltage change is less than $10 \text{mV}$ yielding lower values in $H$ so that the presence of $\Delta z^2$ in the denominator of Eq. (4) leads to a substantially smaller adjustment of the gradient estimated than would be obtained for perfect measurements with $\Delta z^2 = 0$.

If $Q$ is chosen too small, the procedure leads to a slow adaptation of the gradient based on new measurements in each iteration, which may lead to the same disadvantages as using Eq. (1) with constant gradients. These include slow convergence to the target values if the gradient elements are estimated to large, and oscillation or divergence if the gradient elements are estimated to small so that each step overcompensates the deviation from the target values. On the other hand, a large $Q$ can be problematic if individual measurements sometimes produce outliers, e.g., due to undetected bad fit convergence, as these than lead to a large erroneous change in the gradient that may compromise the continuation of the algorithm. Furthermore, previously available information is then discarded very quickly. Smaller $Q$ dampen the response to such events and make better use of the information in the history of previous iterations. Hence, one must choose $Q$ just large enough so that excessive oscillations are avoided. A potential improvement would be to choose $Q^{(k)}$ such that it increases with $v^{(k)} - v^{(k-1)}$ so that larger adjustments are made to the gradient if larger changes are to be expected whereas information is retained if can be expected to be valuable.