

# 1 Discussion

In this chapter the methods, results and analyses will be discussed, first for the qubit state transfers and secondly for the cat code encodings.

## 1.1 Qubit State Transfers and Performance of Krotov's Method

### 1.1.1 $|0\rangle \rightarrow |1\rangle$ state transfer

We start by looking at the  $|0\rangle \rightarrow |1\rangle$  transfer. As the system was set up to realise a  $\pi$ -rotation for a Blackman pulse with length  $T = 6\sigma = 6 \times 3 \text{ ns} = 18 \text{ ns}$ , it is no surprise that the starting fidelity is almost unity down to pulse lengths of 18 ns. Shorter than that, the amplitude constraint affects the starting pulse; it is maximised but starts to become shorter. This explains the drop in starting fidelity all the way down to 4.25 ns. Further, for pulse lengths shorter than 10.75 ns the optimization cannot reach the fidelity goal of  $F > 0.99999$  which is expected as the amplitude constraint means that there just is not enough time to realise the state transfer. This is supported by the fact that the imaginary part is maximised for pulse lengths shorter than 10.75 ns. It can be assumed that the optimization is trying to compensate for the shorter pulse time.

One possible reason for the fluctuations in the number of iterations for the pulse lengths which did not reach the fidelity goal is that the steepness around the (assumed) local minima could differ significantly from each other. This is very much in contrast to the low and steady number of iterations needed when the fidelity goal is reached. Even though the increase in pulse length should increase the number of possible solutions, increasing the chance of reaching the fidelity goal, the fast convergence is most probably due to the fact that the initial guess pulse brings the state so close to the target. One might consider exploring this solution space, however it could potentially be infinite in size. This is not needed to use the optimized pulse shapes as any solution which realises the transfer should be good enough.

With the previous statement in mind, the fact that the solutions found for pulse lengths shorter than 18 ns show a small increase in the  $|2\rangle$  population after the population inversion could point to a need to penalize this state. That is because if the state is too occupied during the transfer one cannot be sure that the states above that will remain unoccupied. This could be checked by simulating the system using the optimized pulses, but with a larger Hilbert space. If the final state reaches the target, the truncation is a good approximation. `krotov` suggests to add dissipation to the states which should be penalized as there is currently no support for state-dependent constraints. However this also increases the computation time as the Hamiltonian needs to either be expressed as a Liouvillian or propagated with collapse terms in the master equation.

The spectrum of the pulse shapes somewhat explains the “strategy” of the optimization. For the longer pulse lengths there is no problem in keeping the pulse frequency mostly around the first transition frequency, only driving the qubit to  $|0\rangle$ . As the pulses become shorter the peak becomes wider, but the solutions try to keep the amplitude at the second transition frequency  $\omega_r + \kappa_q$  low to keep the qubit from going into  $|2\rangle$ . Eventually for short enough pulses this level becomes a little bit occupied, as stated before, but then drops to zero. Perhaps the solution tries to “take a shortcut” through  $|2\rangle$ , which may be a viable strategy, but if that is the case then one needs to set  $N_q = 4$  and penalize  $|3\rangle$  to make sure that the truncation is a valid approximation.

### 1.1.2 $|0\rangle \rightarrow |2\rangle$ state transfer

Now we continue with the  $|0\rangle \rightarrow |2\rangle$  transfer. The “strategy” for all pulse lengths in this case is to first invert the population and then at the halfway point start simultaneously driving the second transition level to start pumping to  $|2\rangle$ . The pulse shapes also reflects this as the oscillation that starts after the plateau has a frequency of  $\kappa_q$ , which is also visible in the frequency spectrum. While the peak at  $\omega_q + \kappa_q$  was expected, the peak at  $\omega_q - \kappa_q$  for pulse lengths shorter than 29 ns is a surprise with no clear explanation as to why. It is also not clear why the solutions drive harder at this frequency the shorter the pulse length.

For this optimization the initial guess pulses were very far from any solution, with the fidelity almost starting at zero for all pulse lengths. The fidelity goal  $F > 0.99999$  was reached down to  $T = 30 \text{ ns}$  which is a bit longer than the  $|0\rangle \rightarrow |1\rangle$  transfer. Looking at the occupation dynamics and pulse shapes could give a clue as to why this is the case. One reason could be that there needs to be enough periods of oscillations to drive to  $|2\rangle$ , which is not the case for the  $|0\rangle \rightarrow |1\rangle$  transfer where the integral of the pulse shape is of importance. Further, it

appears that the fidelity drops in the same way as the  $|0\rangle \rightarrow |1\rangle$  transfer for shorter pulse lengths. However we cannot draw any conclusions for pulse lengths shorter than 22 ns. Perhaps with a low  $\Delta F$  one might reach a solution that follows the trend, but this is only speculation.

The number of iterations also fluctuate for pulse lengths which don't reach the fidelity goal, just as the  $|0\rangle \rightarrow |1\rangle$  transfer.

Interestingly, for both transfers the optimization converges quickly to a relatively high fidelity while the majority of the time is spent fine-tuning the pulse shapes. This could be a side-effect of the static step size. While the number of iterations were large, each iteration step was faster than 1 second, so the total optimization might not be a problem. This optimization time could potentially grow exponentially for larger Hilbert spaces though.

## 1.2 Cat code encoding