Efficient Quantum Circuit for Encoding and Decoding of the [[8, 3, 5]] Stabilizer Code

Ping Dong · Jun Liu · Zhuo-Liang Cao

Abstract A general protocol for constructing a complete efficient encoding and decoding quantum circuit of the [[8, 3, 5]] stabilizer code is proposed. The [[8, 3, 5]] stabilizer code is an eight-qubit code that protects a three-qubit state with up to one error, which is very important for quantum information processing. Single-qubit operations, two-qubit controlled gates and Toffoli gates are required in the proposed circuit. The current protocol can be generalized to all quantum stabilizer codes satisfying quantum Hamming bound, and implemented in some quantum systems.

1 Introduction

Quantum computer has attracted much attention since its potential parallel computing power based on the principle of quantum coherent superposition and quantum entanglement. However, quantum states are vulnerable to various noise during the process of quantum information processing (QIP), i.e., the quantum states will be destroyed (quantum decoherence), thus any weak noise may lead to final calculation error. To overcome this shortcoming, quantum error prevention and correction becomes more and more important. One method is to build a decoherence-free subspace for quantum information storage, transmission and computing [1–6], another one is to generate Calderbank-Shor-Steane (CSS) code and quantum

P. Dong (✉) · J. Liu
The Key Laboratory of Intelligent Computing and Signal Processing, Anhui University, Hefei 230039, China
e-mail: dongping9979@163.com

J. Liu
e-mail: junl@hftc.edu.cn

P. Dong · J. Liu · Z.-L. Cao
School of Electronic and Information Engineering, Hefei Normal University, Hefei 230601, China

Z.-L. Cao
e-mail: zlcao@hftc.edu.cn

stabilizer codes, which can construct an important class of quantum error-correction codes. Due to its less resource-intensive, error correction and good results, quantum error correction is widely used in quantum secure communication, fault-tolerant quantum computing and other fields [7].

Since the original quantum error-correction code, [[9, 1, 3]] code, is proposed by Shor in 1995 [8], and the [[7, 1, 3]] code is proved and produced by Calderbank et al and Steane, independently, in 1996 [9, 10], significant progress has been made on the development of quantum code, especially for quantum stabilizer code. Then, a number of error-correction schemes and constructing schemes for the quantum stabilizer code are proposed [9, 11–16]. Quantum circuit for encoding and decoding of the stabilizer code is also very important for applying the quantum error-correction code on fault-tolerant quantum computation. Recently, the quantum circuits for efficient encoding and decoding of the [[5, 1, 3]] stabilizer code has been presented, where the controlled phase flip gates are used [16]. The [[5, 1, 3]] code, encodes one qubit in five qubits and protects against one-qubit error [17, 18], is only a simple example of quantum error-correction code. Therefore, how to construct an efficient quantum circuit that encoding and decoding for any stabilizer code becomes imperative.

In this letter, we investigate an efficient quantum circuit for encoding and decoding of the [[8, 3, 5]] stabilizer code, which is a typical example for protecting multi-qubit state with up to one error. Meanwhile, it is optimal in the sense of quantum Hamming bound [12]. On the other hand, the [[8, 3, 5]] stabilizer code is the simplest one with generators including all the Pauli operators "I", "X", "Y", "Z". Thus, it can be generalized to all quantum stabilizer codes satisfying Hamming bound. The single-qubit operations, two-qubit controlled gates and three-qubit Toffoli gates are required in current protocol. Here, we first introduce the encoding rule of the [[8, 3, 5]] stabilizer code based on depolarizing channel and stabilizer codes, and then logically analyze the encoding code and give its detailed quantum circuit. To generate the [[8, 3, 5]] stabilizer code words, we choose five generators and three seed generators. This method is more predominant than Ref. [13], which used a systematic approach to generate code words by employing Gaussian elimination on bit flip matrix, because some unsolved difficulties exist in the latter. Then, we describe the decoding circuit of the [[8, 3, 5]] stabilizer code that can be used to correct one error from three qubits. Finally, discussions and conclusion are given.

2 Encoding of the [[8, 3, 5]] Stabilizer Code

Most quantum error-correction code deal with the depolarizing channel which has four basic error operators (Pauli operators) on a qubit: I, X, Y and Z, which denotes "no error", "bit flip", "bit and phase flip" and "phase flip" [19]. In the n-qubit depolarizing channel, error operators can be expressed as $G_n = \pm\{I, X, Y, Z\}^{\otimes n}$, where $\otimes n$ denotes the n-fold tensor product. To correct one of the above errors, one can first find out a quantum stabilizer code. Arbitrary stabilizer code [[n, k, d]] is based on a stabilizer group [16]

$$S = \left\{\prod_{i=1}^{n-k} M_i^{b_i}, b_i \in \{0, 1\}, i = 1, 2, \ldots, n - k\right\} \qquad (1)$$

where $M_1, M_2, \ldots, M_{n-k} \in G_n$ are the generators of S. Then, one can select seed generators $N_1, N_2, \ldots, N_k \in G_n$ that make $\{M_1, M_2, \ldots, M_{n-k}, N_1, N_2, \ldots, N_k\}$ an independent commuting set contained in the normalizer of S. The code words of a stabilizer code are

eigenvectors associated with eigenvalue 1 of the stabilizer code, i.e.,

$$C(S) = \{|\psi\rangle : M|\psi\rangle = |\psi\rangle, \forall M \in S\} \qquad (2)$$

Thus the encoder of a stabilizer code can generate a code word

$$|c_1 \cdots c_k\rangle_L = \frac{1}{\sqrt{2^{n-k}}} \left[\prod_{i=1}^{n-k} (I + M_i) \right] N_1^{c_1} \cdots N_k^{c_k} |0 \cdots 0\rangle_n \qquad (3)$$

Finally, one can efficiently construct quantum circuit of the stabilizer code by means of Eq. (3).

Next, we will discuss the case of $n = 8$, $k = 3$ and $d = 5$ ([[8, 3, 5]] stabilizer code). This stabilizer is optimal in the sense of quantum Hamming. According to Eq. (1), one can select five generators M_1, M_2, M_3, M_4 and M_5, i.e.,

$$M_1 = XXXXXXXX, \qquad M_2 = XZIYIYXZ$$
$$M_3 = XIXIZYZY, \qquad M_4 = XIYZXIYZ \qquad (4)$$
$$M_5 = ZZZZZZZZ$$

where M_1, M_2, M_3, M_4 are linearly independent primary generators and M_5 is called secondary generators [12]. The symplectic matrix corresponding to M_1, M_2, M_3, M_4 and M_5 can be expressed as

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad (5)$$

To choose the code words, we augment these generators with a set of three seed generators, which are chosen so that the seed and primary generators are linearly independent. The three seed generators can be chosen as

$$N_1 = XXIIZIZI, \qquad N_2 = XIXZIIZI$$
$$N_3 = XIIZXZII \qquad (6)$$

According to Eq. (3), we can obtain a standard formula of the quantum code word of the [[8, 3, 5]] stabilizer code (see the Appendix).

According to Eq. (8), we can design the effective circuit for encoding [[8, 3, 5]] code, as shown in Fig. 1. It is noted that the states c_1, c_2 and c_3 are all used to control phase flip of the code words. The main operators are single-qubit rotations, controlled phase gates and Toffoli gates. Here, c_1, c_2 and c_3 are encoded into eight qubits successfully.

3 Decoding of the [[8, 3, 5]] Stabilizer Code

As a stabilizer code, the important application for the [[8, 3, 5]] stabilizer code is to correct one error from three qubits, thus the design for decoding circuit of the [[8, 3, 5]] stabilizer
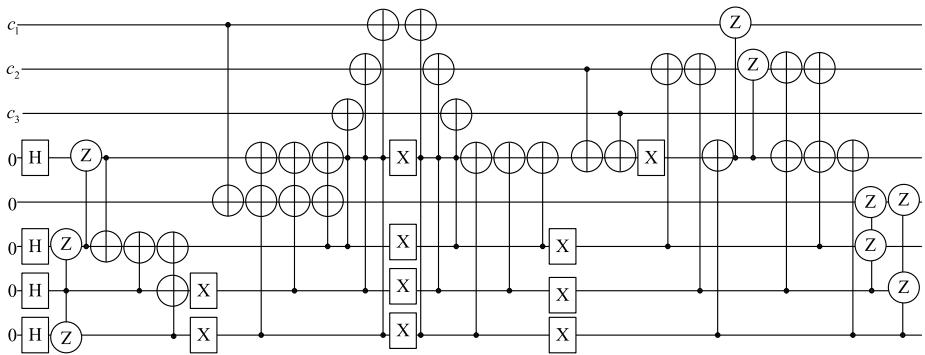
Fig. 1 Circuit for encoding the [[8, 3, 5]] code. "H" denotes Hadamard gate, "●" denotes Controlled operator and "⊕" denotes Not operator

Table 1 Syndromes (S) and error (E) operators

| E	S	E	S	E	S
X_1	00001	Y_1	11111	Z_1	11110
X_2	01001	Y_2	11001	Z_2	10000
X_3	00011	Y_3	10101	Z_3	10110
X_4	01011	Y_4	10011	Z_4	11000
X_5	00101	Y_5	10111	Z_5	10010
X_6	01101	Y_6	10001	Z_6	11100
X_7	00111	Y_7	11101	Z_7	11010
X_8	01111	Y_8	11011	Z_8	10100

code is necessary. Assume that the syndromes of a received state $|\psi\rangle_1$ is defined as $S = |s_1 \cdots s_5\rangle$ where $s_i \in \{0, 1\}$ ($i = 1, 2, \ldots, 5$) and $M_i|\psi\rangle_1 = (-1)^{s_i}|\psi\rangle_1$. One can know any error from above encoded eight qubits as shown in Sect. 2 by means of the syndrome s_i. According to the symplectic Matrix H, it can be observed that the Matrix is composed of the left part and the right part, by which one can obtain syndromes S for different error operations. The syndromes S and error operators E are listed in Table 1, where E denotes X_j, Y_j or Z_j ($j = 1, 2, \ldots, 8$).

Then, one can operate controlled X gates and controlled Z gates on $|\psi\rangle_1$ and $|s_i\rangle$ with $|s_i\rangle$ being controlled qubits. These operations will lead state $|\psi\rangle_1$ to a right state $|\psi\rangle_2$. The decoding quantum circuit of the [[8, 3, 5]] stabilizer code is illustrated in Fig. 2. It can be observed that this decoder is composed of (left) syndrome generation and (right) error correction parts. The main operations are Hadamard operations, controlled gates and multi-controlled gates.

4 Discussions and Conclusions

The [[8, 3, 5]] stabilizer code is another typical example for stabilizer code besides the [[5, 1, 3]] code proposed in previous works [16–18]. We proposed another protocol for the construction and verification of quantum circuit for the [[8, 3, 5]] stabilizer code comparing with Ref. [13]. The complete encoding and decoding circuit has been designed where the
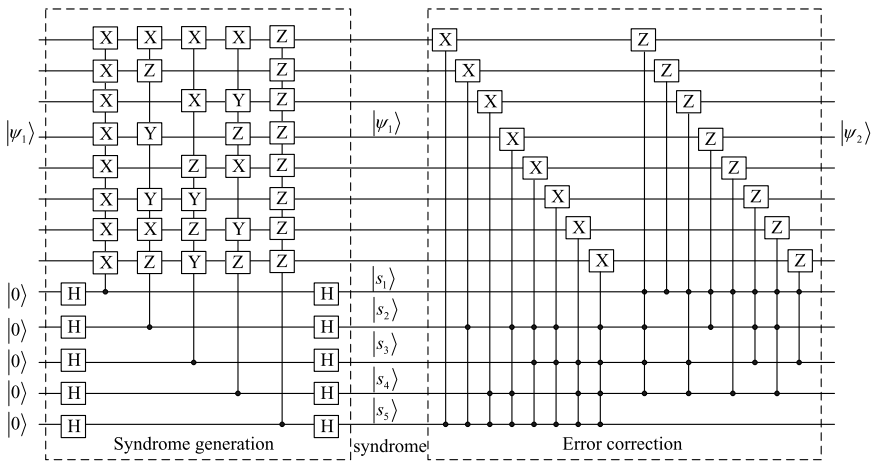
Fig. 2 The effective circuit diagram for decoding the [[8, 3, 5]] stabilizer code

main operators include any single-qubit operations, controlled-phase gates, Toffoli gates, controlled-controlled-X gates and controlled-controlled-Z gates. The quantum circuit can successfully correct one error, which is very important for QIP. The current method can be generalized to implement quantum circuit of any [[n, k, d]] stabilizer code.

The application of quantum error-correction code in several quantum systems is also worth considering for building quantum computer. Recently, some schemes for quantum computing have been investigated including single qubit operations, two-qubit controlled gates [20–31], Toffoli gates [32–38], multi-qubit controlled gates et al [39]. If we can implement any-qubit controlled gates in these systems, the quantum encoding and decoding circuit of the [[8, 3, 5]] stabilizer code would be implemented. It is noted that quantum error-correction code has been used to solve realistic noise in cavity-QED setup [40], where the three-qubit quantum error-correction code can preserve the prepared atomic state against the noise due to fluctuating electric fields, which randomly shift the atomic energy levels via the quadratic stark effect. Besides, the quantum error-correction code can be applied to some of special dephasing noise due to background charge fluctuations (such as random telegraph noise) in quantum dot systems. Thus the stabilizer code satisfying quantum Hamming bound would be applied to some quantum systems as a useful error correction tool.

Appendix: Construction of the Code Words of the [[8, 3, 5]] Stabilizer Code

According to Eq. (3), M_5 has no effect on the generation of the quantum code word of [[8, 3, 5]] stabilizer code, thus the term $(1 + M_5)$ can be eliminated, the code word of the [[8, 3, 5]] stabilizer code can be reduced as

$$|c_1 c_2 c_3\rangle_L = \frac{1}{4} N_1^{c_1} N_2^{c_2} N_3^{c_3} (I + M_1)(I + M_2)(I + M_3)(I + M_4)|00000000\rangle \qquad (7)$$

Then, we combine Eqs. (6) with (3), the detailed code words of the $[[8, 3, 5]]$ can be listed as following

$$|000\rangle_L = \frac{1}{4}\big[|00000000\rangle + |11111111\rangle + |10100101\rangle + |10101010\rangle + |10010110\rangle$$
$$+ |01011010\rangle + |01010101\rangle + |01101001\rangle + |00001111\rangle + |00110011\rangle$$
$$+ |00111100\rangle + |11110000\rangle + |11001100\rangle + |11000011\rangle + |10011001\rangle$$
$$+ |01100110\rangle\big]$$

$$|001\rangle_L = \frac{1}{4}\big[|11000000\rangle + |00111111\rangle + |01100101\rangle + |01101010\rangle - |01010110\rangle$$
$$+ |10011010\rangle + |10010101\rangle - |10101001\rangle + |11001111\rangle - |11110011\rangle$$
$$- |11111100\rangle + |00110000\rangle - |00001100\rangle - |00000011\rangle - |01011001\rangle$$
$$- |10100110\rangle\big]$$

$$|010\rangle_L = \frac{1}{4}\big[|10100000\rangle + |01011111\rangle + |00000101\rangle - |00001010\rangle + |00110110\rangle$$
$$+ |11111010\rangle - |11110101\rangle + |11001001\rangle - |10101111\rangle + |10010011\rangle$$
$$- |10011100\rangle - |01010000\rangle + |01101100\rangle - |01100011\rangle - |00111001\rangle$$
$$- |11000110\rangle\big]$$

$$|011\rangle_L = \frac{1}{4}\big[|01100000\rangle + |10011111\rangle + |11000101\rangle - |11001010\rangle - |11110110\rangle$$
$$+ |00111010\rangle - |00110101\rangle - |00001001\rangle - |01101111\rangle - |01010011\rangle$$
$$+ |01011100\rangle - |10010000\rangle - |10101100\rangle + |10100011\rangle + |11111001\rangle$$
$$+ |00000110\rangle\big]$$

$$|100\rangle_L = \frac{1}{4}\big[|10001000\rangle + |01110111\rangle - |00101101\rangle + |00100010\rangle + |00011110\rangle$$
$$- |11010010\rangle + |11011101\rangle + |11100001\rangle - |10000111\rangle - |10111011\rangle$$
$$+ |10110100\rangle - |01111000\rangle - |01000100\rangle + |01001011\rangle - |00010001\rangle$$
$$- |11101110\rangle\big]$$

$$|101\rangle_L = \frac{1}{4}\big[|01001000\rangle + |10110111\rangle - |11101101\rangle + |11100010\rangle - |11011110\rangle$$
$$- |00010010\rangle + |00011101\rangle - |00100001\rangle - |01000111\rangle + |01111011\rangle$$
$$- |01110100\rangle - |10111000\rangle + |10000100\rangle - |10001011\rangle + |11010001\rangle$$
$$+ |00101110\rangle\big]$$

$$|110\rangle_L = \frac{1}{4}\big[|00101000\rangle + |11010111\rangle - |10001101\rangle - |10000010\rangle + |10111110\rangle$$
$$- |01110010\rangle - |01111101\rangle + |01000001\rangle + |00100111\rangle - |00011011\rangle$$
$$- |00010100\rangle + |11011000\rangle - |11100100\rangle - |11101011\rangle + |10110001\rangle$$
$$+ |01001110\rangle\big]$$

$$|111\rangle_L = \frac{1}{4}\big[|11101000\rangle + |10010111\rangle - |01001101\rangle - |01000010\rangle - |01111110\rangle$$
$$- |10110010\rangle - |10111101\rangle - |10000001\rangle + |11100111\rangle + |11011011\rangle$$
$$+ |11010100\rangle + |00011000\rangle + |00100100\rangle + |00101011\rangle - |01110001\rangle$$
$$- |10001110\rangle\big]$$

According to the above equations, the encoding rule of the [[8, 3, 5]] stabilizer code can be reduced to

$$|c_1 c_2 c_3\rangle_L = \frac{1}{4}\big[(-1)^{c_1 \cdot m} \cdot (-1)^{c_2 \cdot m}|c_1 c_2 c_3 m c_1 000\rangle + (-1)^{\overline{c}_1 \cdot n} \cdot (-1)^{\overline{c}_2 \cdot n}|\overline{c}_1 \overline{c}_2 \overline{c}_3 n c_1 000\rangle$$
$$+ (-1)^{c_1 \cdot n} \cdot (-1)^{c_2 \cdot n}|c_1 c_2 c_3 m \overline{c}_1 111\rangle + (-1)^{\overline{c}_1 \cdot m} \cdot (-1)^{\overline{c}_2 \cdot m}|\overline{c}_1 \overline{c}_2 \overline{c}_3 n \overline{c}_1 111\rangle$$
$$+ a \cdot (-1)^{\overline{c}_1 \cdot n} \cdot (-1)^{\overline{c}_2 \cdot n}|\overline{c}_1 c_2 \overline{c}_3 m \overline{c}_1 010\rangle + a \cdot (-1)^{c_1 \cdot m} \cdot (-1)^{c_2 \cdot m}|c_1 \overline{c}_2 c_3 n \overline{c}_1 010\rangle$$
$$+ a \cdot (-1)^{\overline{c}_1 \cdot m} \cdot (-1)^{\overline{c}_2 \cdot m}|\overline{c}_1 c_2 \overline{c}_3 m c_1 101\rangle + a \cdot (-1)^{c_1 \cdot n} \cdot (-1)^{c_2 \cdot n}|c_1 \overline{c}_2 c_3 n c_1 101\rangle$$
$$+ a \cdot (-1)^{\overline{c}_1 \cdot m} \cdot (-1)^{c_2 \cdot m}|\overline{c}_1 c_2 c_3 n \overline{c}_1 001\rangle + b \cdot (-1)^{c_1 \cdot n} \cdot (-1)^{\overline{c}_2 \cdot n}|c_1 \overline{c}_2 \overline{c}_3 m \overline{c}_1 001\rangle$$
$$+ b \cdot (-1)^{\overline{c}_1 \cdot n} \cdot (-1)^{c_2 \cdot n}|\overline{c}_1 c_2 c_3 n c_1 110\rangle + a \cdot (-1)^{c_1 \cdot m} \cdot (-1)^{\overline{c}_2 \cdot m}|c_1 \overline{c}_2 \overline{c}_3 m c_1 110\rangle$$
$$+ (-1)^{c_1 \cdot m} \cdot (-1)^{\overline{c}_2 \cdot m}|c_1 c_2 \overline{c}_3 n \overline{c}_1 100\rangle - (-1)^{\overline{c}_1 \cdot n} \cdot (-1)^{c_2 \cdot n}|\overline{c}_1 \overline{c}_2 c_3 m \overline{c}_1 100\rangle$$
$$- (-1)^{c_1 \cdot n} \cdot (-1)^{\overline{c}_2 \cdot n}|c_1 c_2 \overline{c}_3 n c_1 011\rangle + (-1)^{\overline{c}_1 \cdot m} \cdot (-1)^{c_2 \cdot m}|\overline{c}_1 \overline{c}_2 c_3 m c_1 011\rangle\big]$$

$$(8)$$

where $m = c_2 \oplus c_3$, $\overline{m} = n = \overline{c_2 \oplus c_3}$, $a = (-1)^{c_1}$ and $b = (-1)^{\overline{c}_1}$.

References

1. Lidar, D.A., Chuang, I.L., Whaley, K.B.: Phys. Rev. Lett. **81**, 2594 (1998)
2. Zhang, Y., Zhou, Z.W., Yu, B., Guo, G.C.: Phys. Rev. A **69**, 042315 (2004)
3. Wu, L.-A., Zanardi, P., Lidar, D.A.: Phys. Rev. Lett. **95**, 130501 (2005)
4. Brooke, P.G.: Phys. Rev. A **75**, 022320 (2007)
5. Duan, L.M., Guo, G.C.: Phys. Rev. Lett. **79**, 1953 (1997)
6. Duan, L.M., Guo, G.C.: Phys. Rev. A **57**, 737 (1998)
7. Nielsen, M.A.: Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000)
8. Shor, P.W.: Phys. Rev. A **52**, 2493 (1995)
9. Calderbank, A.R., Shor, P.W.: Phys. Rev. A **54**, 1098 (1996)
10. Steane, A.M.: Phys. Rev. Lett. **77**, 793 (1996)
11. Cleve, R.: Phys. Rev. A **55**, 4054 (1997)
12. Gottesman, D.: Phys. Rev. A **54**, 1862 (1996)
13. Cleve, R., Gottesman, D.: Phys. Rev. A **56**, 76 (1997)
14. Calderbank, A.R., Rains, E.M., Shor, P.W., Sloane, N.J.A.: IEEE Trans. Inf. Theory **44**, 1369 (1998)

15. Ketkar, A., Klappenecker, A., Kumar, S., Sarvepalli, P.K.: IEEE Trans. Inf. Theory **52**, 4892 (2006)
16. Wu, C.H., Tsai, Y.C., Tsai, H.L.: IEEE Int. Symp. Circuits Syst. Proc. **3**, 2333 (2005)
17. Laflamme, R., Miquel, C., Paz, J.P., Zurek, W.H.: Phys. Rev. Lett. **77**, 198 (1996)
18. Bennett, C.H., DiVincenzo, D.P., Smolin, J.A., Wootters, W.K.: Phys. Rev. A **54**, 3824 (1996)
19. Cafaro, C., Mancini, S.: Phys. Rev. A **82**, 012306 (2010)
20. Loss, D., DiVincenzo, D.P.: Phys. Rev. A **57**, 120 (1998)
21. Li, X.Q., Wu, Y.W., Steel, D.C., Gammon, D., et al.: Science **301**, 809 (2003)
22. Li, X.Q., Yan, Y.J.: Phys. Rev. B **65**, 205301 (2002)
23. Hanson, R., Burkard, G.: Phys. Rev. Lett. **98**, 050502 (2007)
24. Stepanenko, D., Burkard, G.: Phys. Rev. B **75**, 085324 (2007)
25. Hao, X., Zhu, S.Q.: Phys. Rev. A **76**, 044306 (2007)
26. Zhang, H., Guo, G.P., Tu, T., Guo, G.C.: Phys. Rev. A **76**, 012335 (2007)
27. Dong, P., Cao, Z.L.: Phys. Lett. A **373**, 1527 (2009)
28. Wu, Y., Payne, M.G., Hagley, E.W., Deng, L.: Phys. Rev. A **69**, 063803 (2004)
29. Xue, Z.Y.: Europhys. Lett. **93**, 20007 (2011)
30. Xue, Z.Y., Wang, Z.D., Zhu, S.L.: Phys. Rev. A **77**, 024301 (2008)
31. Xue, Z.Y., Wang, Z.D.: Phys. Rev. A **75**, 064303 (2007)
32. Chen, C.Y., Feng, M., Gao, K.L.: Phys. Rev. A **73**, 064304 (2006)
33. Chen, C.Y., Li, S.H.: Eur. Phys. J. D **41**, 557 (2007)
34. Ralph, T.C., Resch, K.J., Gilchrist, A.: Phys. Rev. A **75**, 022313 (2007)
35. Shao, X.Q., Zhu, A.D., Zhang, S., Chung, J.S., Yeon, K.H.: Phys. Rev. A **75**, 034307 (2007)
36. Ionicioiu, R., Spiller, T.P., Munro, W.J.: Phys. Rev. A **80**, 012312 (2009)
37. Lin, Q., Li, J.: Phys. Rev. A **79**, 022301 (2009)
38. Borrelli, M., Mazzola, L., Paternostro, M., Maniscalco, S.: Phys. Rev. A **84**, 012314 (2011)
39. Yang, C.P., Han, S.: Phys. Rev. A **72**, 032311 (2005)
40. Ottaviani, C., Vitali, D.: Phys. Rev. A **82**, 012319 (2010)