

Join GitHub today

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)

Branch: master ▾

[Find file](#)[Copy path](#)[krotov](#) / [tests](#) / **test_mu.py** **goerz** Fix `derivative_wrt_pulse` (#47, #48)

cb0d333 Jul 9, 2019

[1 contributor](#)

Raw

Blame

History



131 lines (118 sloc) | 4.03 KB

```
1  """Tests for krotov.mu"""
2  import pytest
3  from qutip import ket, sigmam, sigmap, sigmax, sigmaz
4
5  import krotov
6
7
8  @pytest.fixture
9  def tls_control_system():
10     """Non-trivial control system defined on a TLS"""
11     eps1 = lambda t, args: 0.5
12     eps2 = lambda t, args: 1
13     H1 = [0.5 * sigmaz(), [sigmap(), eps1], [sigmam(), eps1]]
14     H2 = [0.5 * sigmaz(), [sigmaz(), eps2]]
15     c_ops = [0.1 * sigmap()]
16     objectives = [
17         krotov.Objective(
18             initial_state=ket('0'), target=ket('1'), H=H1, c_ops=c_ops
19         ),
20         krotov.Objective(
21             initial_state=ket('0'), target=ket('1'), H=H2, c_ops=c_ops
22         ),
23     ]
24     controls = [eps1, eps2]
25     controls_mapping = krotov.structural_conversions.extract_controls_mapping(
26         objectives, controls
27     )
28     return objectives, controls, controls_mapping
29
30
31  @pytest.fixture
32  def tls_control_system_tdcops(tls_control_system):
33     """Control system with time-dependent collapse operators"""
34     objectives, controls, _ = tls_control_system
35     c_op = [[0.1 * sigmap(), controls[0]]]
36     c_ops = [c_op]
37     H1 = objectives[0].H
38     H2 = objectives[1].H
39     objectives = [
40         krotov.Objective(
41             initial_state=ket('0'), target=ket('1'), H=H1, c_ops=c_ops
42         ),
```

```

43         krotov.Objective(
44             initial_state=ket('0'), target=ket('1'), H=H2, c_ops=c_ops
45         ),
46     ]
47     controls_mapping = krotov.structural_conversions.extract_controls_mapping(
48         objectives, controls
49     )
50     return objectives, controls, controls_mapping
51
52
53 def test_derivative_wrt_pulse_multiple_terms(tls_control_system):
54     """Test the calculation of  $\mu$  if the same control appears more than once"""
55     objectives, pulses, pulses_mapping = tls_control_system
56     # distinction between controls and pulses doesn't matter here, we're only
57     # considering linear controls and don't plug in any time_index
58     i_objective = 0
59     mu = krotov.mu.derivative_wrt_pulse(
60         objectives,
61         i_objective,
62         pulses,
63         pulses_mapping,
64         i_pulse=0,
65         time_index=0,
66     )
67     #  $0.5 * (\sigma_+ + \sigma_-) = \sigma_x$ 
68     for state in (ket('0'), ket('1')):
69         assert (mu(state) - (sigmax()(state)).norm('max')) == 0
70         assert (mu(state)).dims == state.dims
71
72
73 def test_derivative_wrt_pulse_zero(tls_control_system):
74     """Test that  $\mu=0$  if taking derivative wrt pulse not in objective"""
75     objectives, pulses, pulses_mapping = tls_control_system
76     # distinction between controls and pulses doesn't matter here, we're only
77     # considering linear controls and don't plug in any time_index
78     i_objective = 0
79     mu = krotov.mu.derivative_wrt_pulse(
80         objectives,
81         i_objective,
82         pulses,
83         pulses_mapping,
84         i_pulse=1,
85         time_index=0,
86     )
87     for state in (ket('0'), ket('1')):
88         assert mu(state).norm('max') == 0
89         assert (mu(state)).dims == state.dims
90
91     i_objective = 1
92     mu = krotov.mu.derivative_wrt_pulse(
93         objectives,
94         i_objective,
95         pulses,
96         pulses_mapping,
97         i_pulse=0,
98         time_index=0,
99     )
100     for state in (ket('0'), ket('1')):
101         assert mu(state).norm('max') == 0
102         assert (mu(state)).dims == state.dims
103
104
105 def test_derivative_wrt_pulse_no_timedependent_cops(tls_control_system_tdcops):
106     """Test that time-dependent collapse operators are no allowed"""
107     objectives, pulses, pulses_mapping = tls_control_system_tdcops
108     i_objective = 0
109     with pytest.raises(NotImplementedError):
110         krotov.mu.derivative_wrt_pulse(

```

```
111         objectives,
112         i_objective,
113         pulses,
114         pulses_mapping,
115         i_pulse=0,
116         time_index=0,
117     )
118     # however, we do allow the c_ops to be time-dependent with controls we're
119     # not taking the derivative with respect to
120     mu = krotov.mu.derivative_wrt_pulse(
121         objectives,
122         i_objective,
123         pulses,
124         pulses_mapping,
125         i_pulse=1,
126         time_index=0,
127     )
128     for state in (ket('0'), ket('1')):
129         assert mu(state).norm('max') == 0
130         assert (mu(state)).dims == state.dims
```