UniDyn--Demo-Scratch.nb

John A. Marohn
jam99@cornell.edu
Cornell University

Abstract: This demonstration notebook loads the **UniDyn** package and executes the package's unit tests.

Set the path to the package

Tell *Mathematica* the path to the directory containing the package.

EDIT THE FOLLOWING PATH STRING:

In[19]:=
```
$NCPath = "/Users/jam99/Dropbox";
$UniDynPath =
  "/Users/jam99/Dropbox/MarohnGroup__Software_Library/UniDyn/
    unidyn";
```

YOU SHOULD NOT NEED TO EDIT ANYTHING FROM HERE ONWARDS.

Load the package

Append the package path to the system path. Before trying to load the package, ask *Mathematica* to find it. This is a test that we directed *Mathematica* to the correct directory. The output of this command should be the full system path to the UniDyn.m file.

In[21]:=
```
$Path = AppendTo[$Path, $NCPath];
$Path = AppendTo[$Path, $UniDynPath];
FindFile["UniDyn`"]
FindFile["NC`"]
```

Out[23]=
```
/Users/jam99/Dropbox/MarohnGroup__Software_Library/UniDyn/unidyn/UniDyn.m
```

Out[24]=
```
/Users/jam99/Dropbox/NC/init.m
```

Now that we are confident that the path is set correctly, load the package. Setting the global $VerboseLoad variable to True will print out the help strings for key commands in the package.

In[25]:=
```
$VerboseLoad = False; (* Set to load quietly *)
Needs["UniDyn`"]
```

Execute the units tests in batch

Included with the package are a number of files, ending in “-tests.m”, that contain tests of the package’s functions -- so-called unit tests. Set the working directory to the package directory and pretty-print the directory name.

In[27]:=
```
SetDirectory[$UniDynPath];
TableForm[{{$UniDynPath}}, TableHeadings → {None, {"Directory"}}]
```

Out[28]//TableForm=

Directory
/Users/jam99/Dropbox/MarohnGroup__Software_Library/UniDyn/unidyn

Get the names of all the unit-testing files included with the package (following my convention that the unit testing file end in “-tests.m”). Pretty-print the names of the unit-test files included with the package.

In[29]:=
```
fn = FileNames["*-tests.m"];
TableForm[{{fn}}, TableHeadings → {None, {"Test files found"}}]
```

Out[30]//TableForm=

Test files found
Comm-tests.m
Evolve-tests.m
Mult-tests.m
OpCreate-tests.m
Osc-tests.m
Spins-tests.m

Finally, carry out the unit tests and make a report.

In[31]:=
```
tr = TestReport /@ fn;
TableForm[Table[tr [[k]], {k, 1, Length[tr]}]]
```

Out[32]//TableForm=
```
TestReportObject[ Title: Test Report: Comm–tests.m
                  Success rate: 100%   Tests run: 12 ]
TestReportObject[ Title: Test Report: Evolve–tests.m
                  Success rate: 100%   Tests run: 27 ]
TestReportObject[ Title: Test Report: Mult–tests.m
                  Success rate: 100%   Tests run: 20 ]
TestReportObject[ Title: Test Report: OpCreate–tests.m
                  Success rate: 100%   Tests run: 23 ]
TestReportObject[ Title: Test Report: Osc–tests.m
                  Success rate: 100%   Tests run: 20 ]
TestReportObject[ Title: Test Report: Spins–tests.m
                  Success rate: 100%   Tests run: 14 ]
```

Make a report.

In[150]:=
```
tests$passed$total =
  Plus @@ (tr⟦#⟧["TestsSucceededCount"] & /@ List @@ Table[k, {k, 1, Length[tr]}]);
tests$failed$total = Plus @@
   (tr⟦#⟧["TestsFailedCount"] & /@ List @@ Table[k, {k, 1, Length[tr]}]);
Print[Style[ToString[tests$passed$total] <> " tests passed",
  FontWeight → Bold, FontSize → 18, FontColor → Blue]]
Print[Style[ToString[tests$failed$total] <> " tests failed",
  FontWeight → Bold, FontSize → 18, FontColor → Red]]
```

116 tests passed

0 tests failed

Single spin: Taking repeated commutators

Create a single-spin system to play with.

In[37]:=
```
Needs["UniDyn`"];
Clear[H, Ix, Iy, Iz, Δ, ω]
CreateScalar[{Δ, ω}];
SpinSingle$CreateOperators[Ix, Iy, Iz];
```

SpinSingle$CreateOperators: Creating spin operators.

SpinSingle$CreateOperators: Adding spin commutations relations.

SpinSingle$CreateOperators: No angular momentum L defined.

Define a Hamiltonian

In[41]:=
```
H = Δ Iz + ω Ix;
```

Repeated commutators

Define a function to take *n* commutators of an operator *Op* with the -i times the Hamiltonian *H*.

In[42]:=
```
Clear[RepeatedComm];
RepeatedComm[1, H_, Op_] := List[Op];
RepeatedComm[n_, H_, Op_] := Prepend[RepeatedComm[n - 1, H, Op],
   -I Comm[H, RepeatedComm[n - 1, H, Op][[1]]]];
```

Example calculation of repeated commutators

In[45]:=
```
σ = RepeatedComm[5, H, Iz] // Expand // Simplify;
σ // MatrixForm
```

Out[46]//MatrixForm=

$$\begin{pmatrix} \omega\,(-\mathrm{Ix}\,\Delta + \mathrm{Iz}\,\omega)\,(\Delta^2 + \omega^2) \\ \mathrm{Iy}\,\omega\,(\Delta^2 + \omega^2) \\ \omega\,(\mathrm{Ix}\,\Delta - \mathrm{Iz}\,\omega) \\ -\mathrm{Iy}\,\omega \\ \mathrm{Iz} \end{pmatrix}$$

Evolution

Use the Evolver function to calculate the evolution of the I_x operator under the on-resonance Zeeman Hamiltonian first.

```
$Assumptions = {Δ ∈ Reals, Δ > 0};
SetOptions[Evolver, quiet → True];
(* Set this to False when debugging. *)
Evolver[Δ Iz, t, Ix]
```

Out[49]= $\mathrm{Ix}\,\mathrm{Cos}[t\,\Delta] + \mathrm{Iy}\,\mathrm{Sin}[t\,\Delta]$

As a check, use the Evolver function to calculate the evolution of the I_x operator under the off-resonance Zeeman Hamiltonian.

```
$Assumptions = {Δ ∈ Reals, Δ > 0, ω ∈ Reals, ω ≥ 0};
SetOptions[Evolver, quiet → True];
(* Set this to False when debugging. *)
Evolver[H, t, Ix]
```

Out[52]=
$$\frac{1}{2\left(\Delta^2+\omega^2\right)^{3/2}} e^{-i t \sqrt{\Delta^2+\omega^2}} \Big(i\,\mathrm{Iy}\,\Delta^3 - i\,e^{2 i t \sqrt{\Delta^2+\omega^2}}\,\mathrm{Iy}\,\Delta^3 + i\,\mathrm{Iy}\,\Delta\,\omega^2 - i\,e^{2 i t \sqrt{\Delta^2+\omega^2}}\,\mathrm{Iy}\,\Delta\,\omega^2 + \mathrm{Ix}\,\Delta^2\sqrt{\Delta^2+\omega^2} + e^{2 i t \sqrt{\Delta^2+\omega^2}}\,\mathrm{Ix}\,\Delta^2\sqrt{\Delta^2+\omega^2} - \mathrm{Iz}\,\Delta\,\omega\sqrt{\Delta^2+\omega^2} + 2\,e^{i t \sqrt{\Delta^2+\omega^2}}\,\mathrm{Iz}\,\Delta\,\omega\sqrt{\Delta^2+\omega^2} - e^{2 i t \sqrt{\Delta^2+\omega^2}}\,\mathrm{Iz}\,\Delta\,\omega\sqrt{\Delta^2+\omega^2} + 2\,e^{i t \sqrt{\Delta^2+\omega^2}}\,\mathrm{Ix}\,\omega^2\sqrt{\Delta^2+\omega^2} \Big)$$

Some simplification is needed to get a nice-looking answer.

In[55]:=
```
$Assumptions = {Δ ∈ Reals, Δ > 0, ω ∈ Reals, ω ≥ 0};
ρ = Collect[Expand[Simplify[ExpToTrig[Evolver[H, t, Ix]]]],
  {Ix, Iy, Iz}, Simplify]
```

Out[56]=
$$\frac{\mathrm{Ix}\left(\omega^2+\Delta^2\,\mathrm{Cos}\left[t\sqrt{\Delta^2+\omega^2}\right]\right)}{\Delta^2+\omega^2} + \frac{2\,\mathrm{Iz}\,\Delta\,\omega\,\mathrm{Sin}\left[\frac{1}{2}t\sqrt{\Delta^2+\omega^2}\right]^2}{\Delta^2+\omega^2} + \frac{\mathrm{Iy}\,\Delta\,\mathrm{Sin}\left[t\sqrt{\Delta^2+\omega^2}\right]}{\sqrt{\Delta^2+\omega^2}}$$

Quantum optics

Operators

In[57]:=
```
Needs["UniDyn`"];
Clear[H, Ix, Iy, Iz, Δ, ω, g,
 F, aL, aR, Q$sym, P$sym, Q, P, QP$rules]
CreateScalar[{Δ, ω, Δω, g, F, ϕ}];
CreateOperator[{{Ix, Iy, Iz}, {aL, aR}}];
SpinSingle$CreateOperators[Ix, Iy, Iz, 1 / 2];
OscSingle$CreateOperators[aL, aR];

Q$sym = (aR + aL) / Sqrt[2];
P$sym = I (aR - aL) / Sqrt[2];
QP$rules = {aR → (Q - I P) / Sqrt[2], aL → (Q + I P) / Sqrt[2]};
```

SpinSingle$CreateOperators: Spin operators already exist.

SpinSingle$CreateOperators: Adding spin commutations relations.

SpinSingle$CreateOperators: Angular momentum L = 1/2. Adding operator simplification rules.

OscSingle$CreateOperators: Oscillator operators already exist.

OscSingle$CreateOperators: Adding oscillator commutations relations.

In[66]:=
```
Q$sym /. QP$rules // Simplify
P$sym /. QP$rules // Simplify
```

Out[66]= Q

Out[67]= P

Hamiltonians

Free evolution

In[68]:=
```
$Assumptions = {Element[ω, Reals], ω > 0};
```
$$H\$0 = \frac{\omega}{2} (aR ** aL + aL ** aR);$$

In[70]:=
```
Simplify[Evolver[H$0, t, Q$sym] /. QP$rules]
```

Out[70]= $Q \, \mathrm{Cos}[t\,\omega] - P \, \mathrm{Sin}[t\,\omega]$

Position or momentum kick

In[71]:=
```
Clear[δx, δp];
$Assumptions = {Element[δx, Reals], Element[δp, Reals]};
H$0$x$kick = δx P$sym ;
H$0$p$kick = δp Q$sym;
```

In[75]:=
```
Simplify[Evolver[H$0$x$kick , t, Q$sym]  /.
    QP$rules ~Join~ {t → 1}]  // Simplify
Simplify[Evolver[H$0$p$kick , t, P$sym]  /.
    QP$rules ~Join~ {t → 1}]  // Simplify
```

Out[75]= $Q - \delta x$

Out[76]= $P + \delta p$

Phase Kick

In[77]:=
```
$Assumptions = {Element[ω, Reals], ω > 0};
H$0$phase$kick = (ω + Δω)/2 (aR ** aL + aL ** aR) ;
```

In[79]:=
```
Simplify[Evolver[H$0$phase$kick, t, Q$sym]  /. QP$rules] /.
   {Δω → Δϕ / t} // Simplify
```

Out[79]= Q Cos[Δϕ + t ω] – P Sin[Δϕ + t ω]

Force

In[80]:=
```
$Assumptions = {Element[F, Reals], F > 0};
H$1 = - F Q$sym;
```

In[82]:=
```
Simplify[Evolver[H$1, t, Q$sym] /. QP$rules ]
Simplify[Evolver[H$1, t, P$sym] /. QP$rules]
```

Out[82]= Q

Out[83]= P – F t

In[84]:=
```
Simplify[Evolver[H$1, t, aR]]
Simplify[Evolver[H$1, t, aL]]
```

Out[84]= $aR + \frac{i\,F\,t}{\sqrt{2}}$

Out[85]= $aL - \frac{i\,F\,t}{\sqrt{2}}$

Squeezing

In[86]:=
```
$Assumptions =
  {Element[Δ, Reals], Δ > 0, Element[ω, Reals], ω > 0};
```

In[87]:= $\text{Simplify}\left[\text{Evolver}\left[-\frac{\Delta}{2}\,\text{I (aR ** aR - aL ** aL), t, \#}\right]\ \text{/. QP\$rules}\right]\ \text{\& /@}$
 {Qsym, Psym}

Out[87]= $\{e^{t\Delta}\,Q,\ e^{-t\Delta}\,P\}$

In[88]:= $\text{Simplify}\left[\text{Evolver}\left[\frac{\Delta}{2}\,\text{I (aR ** aR - aL ** aL), t, \#}\right]\ \text{/. QP\$rules}\right]\ \text{\& /@}$
 {Qsym, Psym}

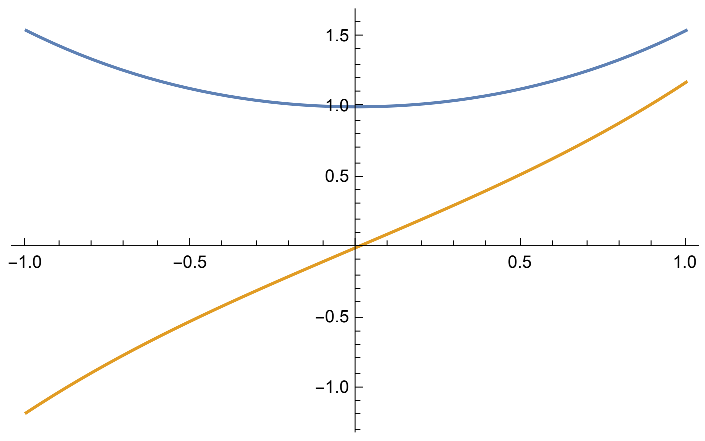
Out[88]= $\{e^{-t\Delta}\,Q,\ e^{t\Delta}\,P\}$

In[89]:= $\text{Expand}\left[\text{Simplify}\left[\text{ExpToTrig}\left[\text{Evolver}\left[\frac{\Delta}{2}\,\text{(aR ** aR + aL ** aL), t, \#}\right]\ \text{/.}\right.\right.\right.$
 $\left.\left.\left.\text{QP\$rules}\right]\right]\right]\ \text{\& /@ \{Q\$sym, P\$sym\}}$

Out[89]= {Q Cosh[t Δ] + P Sinh[t Δ], P Cosh[t Δ] + Q Sinh[t Δ]}

Let's remind ourselves what the hyperbolic functions look like

In[90]:=
```
Plot[{Cosh[T], Sinh[T]}, {T, -1, 1}]
```

Out[90]=


In[91]:=
```
Collect[Simplify[ExpToTrig[
    Evolver[ω/2 (aR ** aL + aL ** aR) - Δ/2 I (aR ** aR - aL ** aL),
     t, #] /. QP$rules]], {Q, P}] & /@ {Q$sym, P$sym}
```

Out[91]=
$$\left\{-\frac{P\,\omega\,\mathrm{Sinh}\left[t\sqrt{\Delta^2-\omega^2}\right]}{\sqrt{\Delta^2-\omega^2}}+Q\left(\mathrm{Cosh}\left[t\sqrt{\Delta^2-\omega^2}\right]+\frac{\Delta\,\mathrm{Sinh}\left[t\sqrt{\Delta^2-\omega^2}\right]}{\sqrt{\Delta^2-\omega^2}}\right),\right.$$
$$\left.\frac{Q\,\omega\,\mathrm{Sinh}\left[t\sqrt{\Delta^2-\omega^2}\right]}{\sqrt{\Delta^2-\omega^2}}+P\left(\mathrm{Cosh}\left[t\sqrt{\Delta^2-\omega^2}\right]-\frac{\Delta\,\mathrm{Sinh}\left[t\sqrt{\Delta^2-\omega^2}\right]}{\sqrt{\Delta^2-\omega^2}}\right)\right\}$$

Another quantum optics example

In[92]:=
```
Needs["UniDyn`"];
Clear[λ, t, Ix, Iy, Iz, I$m, I$p, aL, aR, H];
CreateScalar[{λ, t}];
CreateOperator[{{Ix, Iy, Iz, I$m, I$p}, {aL, aR}}];
OscSingle$CreateOperators[aL, aR];
```

OscSingle$CreateOperators: Oscillator operators already exist.

OscSingle$CreateOperators: Adding oscillator commutations relations.

In[97]:=
```
I$p /: Comm[I$p, I$m] = 2 Iz;
I$p /: Comm[I$p, Iz] = -I$p;
I$m /: Comm[I$m, I$p] = -2 Iz;
I$m /: Comm[I$m, Iz] = I$m;
Iz /: Comm[Iz, I$p] = I$p;
Iz /: Comm[Iz, I$m] = -I$m;
```

In[103]:=
```
Iz /: NonCommutativeMultiply[a___, Iz, Iz, b___] :=
   1/4 NonCommutativeMultiply[a, b];
I$p /: NonCommutativeMultiply[a___, I$p, I$p, b___] := 0;
I$m /: NonCommutativeMultiply[a___, I$m, I$m, b___] := 0;

I$p /: NonCommutativeMultiply[a___, I$p, Iz, b___] :=
  - 1/2 NonCommutativeMultiply[a, I$p, b];
I$p /: NonCommutativeMultiply[a___, I$p, I$m, b___] :=
   1/2 NonCommutativeMultiply[a, b] + NonCommutativeMultiply[a, Iz, b];

I$m /: NonCommutativeMultiply[a___, I$m, Iz, b___] :=
   1/2 NonCommutativeMultiply[a, I$m, b];
I$m /: NonCommutativeMultiply[a___, I$m, I$p, b___] :=
   1/2 NonCommutativeMultiply[a, b] - NonCommutativeMultiply[a, Iz, b];

I$z /: NonCommutativeMultiply[a___, I$z, Ip, b___] :=
   1/2 NonCommutativeMultiply[a, I$p, b];
I$z /: NonCommutativeMultiply[a___, I$z, I$m, b___] :=
  - 1/2 NonCommutativeMultiply[a, I$m, b];
```

In[112]:=
```
H = λ (I$p ** aL + I$m ** aR);
Comm[I H, aL]
```

Out[113]= $-i$ I$m λ

In[114]:=
```
Comm[I H, Comm[I H, aL]]
```

Out[114]= $2\lambda^2$ Iz ** aL

In[115]:=
```
Comm[I H, aR]
```

Out[115]= i I$p λ

In[116]:=
```
Comm[I H, Comm[I H, aR]]
```
Out[116]= $2\lambda^2$ Iz ** aR

In[117]:=
```
Comm[I H, Comm[I H, Comm[I H, aL]]] // Simplify
```
Out[117]= $-i\lambda^3$ (I$m - 2 I$m ** aL ** aR + 2 I$p ** aL ** aL)

In[118]:=
```
Comm[I H, Comm[I H, Comm[I H, Comm[I H, aL]]] ] // Simplify
```
Out[118]= $-\lambda^4$ (3 aL - 4 Iz ** aL + 4 Iz ** aL ** aL ** aR + 4 Iz ** aL ** aR ** aL)

In[119]:=
```
Evolver[λ Iz, t, I$p]
Evolver[λ Iz, t, I$m]
```
Out[119]= $e^{-i t \lambda}$ I$p

Out[120]= $e^{i t \lambda}$ I$m

In[121]:=
```
Evolver[λ (I$p ** aL + I$m ** aR), t, aL] // Simplify
```
Evolver: Unrecognized evolution

Out[121]= {aL, i I$m λ, $2\lambda^2$ Iz ** aL, $i\lambda^3$ (I$m - 2 I$m ** aL ** aR + 2 I$p ** aL ** aL),
$-\lambda^4$ (3 aL - 4 Iz ** aL + 4 Iz ** aL ** aL ** aR + 4 Iz ** aL ** aR ** aL)}

In[122]:=
```
Evolver[λ (I$m ** aR), t, aL] // Simplify
```
Out[122]= aL + i I$m t λ

In[123]:=
```
Evolver[λ (I$p ** aL + I$m ** aR), t, (aL - aR)] // Simplify
```
Evolver: Unrecognized evolution

Out[123]= {aL - aR, i (I$m + I$p) λ, $2\lambda^2$ (Iz ** aL - Iz ** aR),
$i\lambda^3$ (I$m - I$p - 2 I$m ** aL ** aR + 2 I$m ** aR ** aR + 2 I$p ** aL ** aL - 2 I$p ** aR ** aL),
$-\lambda^4$ (3 aL - 3 aR - 4 Iz ** aL - 4 Iz ** aR + 4 Iz ** aL ** aL ** aR +
4 Iz ** aL ** aR ** aL - 4 Iz ** aR ** aL ** aR - 4 Iz ** aR ** aR ** aL)}

Failing cases

Off-resonance evolution of Iz is touchy

Force with a phase factor

This fails because Mathematica cannot recognize that Comm[aL $e^{-i\phi}$, aR] → $e^{-i\phi}$ Comm[aL, aR] → $e^{-i\phi}$. Conclude that the Comm[] function is not yet smart enough to factor a *function of scalar* out of a commutator.

In[134]:=
```
CreateScalar[{F, ϕ}];
$Assumptions = {Element[F, Reals], F > 0};
H$1 = F (Exp[-i ϕ] aL + Exp[i ϕ] aL );
```

In[137]:=
```
Evolver[H$1, t, aR]
```

Evolver: Unrecognized evolution

Out[137]=

$$\Big\{aR,\ -i\,F\left(\text{Comm}\left[aL\,e^{i\phi}, aR\right]+\text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right),\ -F^2\Big($$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]+\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]+$$
$$\text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]+$$
$$\text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]\Big),$$
$$i\,F^3\Big(\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]\right]+$$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]\right]+$$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]\right]+$$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]\right]+$$
$$\text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]\right]+$$
$$\text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]\right]+$$
$$\text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]\right]+$$
$$\text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]\right]\Big),$$
$$F^4\Big(\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]\right]\right]+$$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]\right]\right]+$$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]\right]\right]+$$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]\right]\right]+$$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]\right]\right]+$$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]\right]\right]+$$
$$\text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]\right]\right]+$$
$$\text{Comm}\Big[aL\,e^{i\phi},$$
$$\text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, aR\right]\right]\right]\Big]+$$
$$\text{Comm}\left[aL\left(e^{i\phi}\right)^{-1}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, \text{Comm}\left[aL\,e^{i\phi}, aR\right]\right]\right]\right]+$$

$$\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\,e^{i\phi},\mathrm{Comm}\left[aL\,e^{i\phi},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},aR\right]\right]\right]\right]+$$
$$\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\,e^{i\phi},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\,e^{i\phi},aR\right]\right]\right]\right]+$$
$$\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\right.$$
$$\left.\mathrm{Comm}\left[aL\,e^{i\phi},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},aR\right]\right]\right]\right]+$$
$$\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\,e^{i\phi},\mathrm{Comm}\left[aL\,e^{i\phi},aR\right]\right]\right]\right]+$$
$$\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\right.\right.$$
$$\left.\left.\mathrm{Comm}\left[aL\,e^{i\phi},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},aR\right]\right]\right]\right]+\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[\right.\right.$$
$$\left.\left.aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\,e^{i\phi},aR\right]\right]\right]\right]+\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\right.$$
$$\left.\left.\left.\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},\mathrm{Comm}\left[aL\left(e^{i\phi}\right)^{-1},aR\right]\right]\right]\right]\right)\right\}$$