

UniDyn--Study-04.nb

John A. Marohn
jam99@cornell.edu
Cornell University

Abstract: This demonstration notebook loads the **UniDyn** package runs an **Evolver** rotations involving the harmonic oscillator.

Set the path to the package

Check the Mathematica version number .

```
In[60]:= $VersionNumber
```

```
Out[60]:= 12.3
```

Tell *Mathematica* the path to the directory containing the packages and the /studies directory.

EDIT THE FOLLOWING PATH STRINGS:

```
In[61]:= $UniDynPath =  
        "/Users/jam99/Dropbox/MarohnGroup__Software_Library/UniDyn/  
        unidyn";
```

YOU SHOULD NOT NEED TO EDIT ANYTHING FROM HERE ONWARDS.

Load and test the packages

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[62]:= $Path = AppendTo[$Path, $UniDynPath];  
FindFile["UniDyn`"]
```

```
Out[63]:= /Users/jam99/Dropbox/MarohnGroup__Software_Library/UniDyn/unidyn/UniDyn.m
```

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

```
In[64]:= $VerboseLoad = True;
```

```
Needs["UniDyn`"]
```

Carry out the unit tests for the **Unidyn** functions.



```
In[66]:= SetDirectory[$UniDynPath];
```

```
In[67]:= fn = FileNames["*-tests.m"];
```

```
test$report = TestReport /@ fn;
```

```
TableForm[Table[test$report[[k]], {k, 1, Length[test$report]}]]
```

Out[69]//TableForm=


TestReportObject	 	Title: Test Report: Comm-tests.m Success rate: 100% Tests run: 23
TestReportObject	 	Title: Test Report: Evolve-tests.m Success rate: 100% Tests run: 23
TestReportObject	 	Title: Test Report: Mult-tests.m Success rate: 100% Tests run: 18
TestReportObject	 	Title: Test Report: OpQ-tests.m Success rate: 100% Tests run: 21
TestReportObject	 	Title: Test Report: Osc-tests.m Success rate: 100% Tests run: 22
TestReportObject	 	Title: Test Report: Spins-tests.m Success rate: 100% Tests run: 14

Harmonic oscillator

```
In[95]:= Clear[aL, aR,  $\omega$ , Q$sym, P$sym, Q, P,  $\omega$ ,  $\delta q$ ,  $\delta p$ , A,  $\phi$ ,  $\Delta$ , H$0, H$1, H$1$ $\phi$ , H$2]
```

```
OscSingle$CreateOperators[aL, aR];
```

```
CreateScalar[ $\omega$ ,  $\delta q$ ,  $\delta p$ , A,  $\phi$ ,  $\Delta$ ];
```

 **OscSingle\$CreateOperators**: Creating oscillator operators.

 **OscSingle\$CreateOperators**: Adding oscillator commutations relations.

Define the number operator.

```
In[98]:= Nop = Mult[aR, aL];
```

Define position and momentum operators.

```
In[99]:= Q$sym = (aR + aL) / Sqrt[2];
```

```
P$sym = I (aR - aL) / Sqrt[2];
```

Define a rule for rewriting raising and lowering operators in terms of position and momentum

operators.

```
In[101]:= CreateOperator[{{Q, P}}]
```

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

Define the harmonic oscillator Hamiltonian.

```
In[103]:= H$0 = ω (Nop + 1 / 2);
```

Evolve the position and momentum operators under the harmonic oscillator Hamiltonian.

```
In[104]:= {Evolver[H$0, t, Q$sym], Evolver[H$0, t, P$sym]} /. QP$rules // Simplify
```

```
Out[104]:= { 1/2 e^{-i t ω} (i (-1 + e^{2 i t ω}) P + (1 + e^{2 i t ω}) Q), P Cos[t ω] + Q Sin[t ω] }
```

A position kick. (*Change the sign here from Study-03.*)

```
In[105]:= {Evolver[δq P$sym, 1, Q$sym], Evolver[δq P$sym, 1, P$sym]} /. QP$rules // Simplify
```

```
Out[105]:= {Q - δq, P}
```

Examine this evolution more closely.

```
In[120]:= Evolver[δq P$sym, τ, Q$sym, quiet → False]
```

$$\rho \text{ matrix} = \begin{pmatrix} \frac{aL+aR}{\sqrt{2}} \\ -\delta q \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\Omega = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\text{system of equations} = \left\{ \begin{aligned} &\{ \text{Private`x4$67145}'[\text{Private`time$67145}] = 0, \text{Private`x3$67145}'[\text{Private`time$67145}] = 0, \text{Private`x2$67145}'[\text{Private`time$67145}] = 0, \text{Private`x1$67145}'[\text{Private`time$67145}] = 0 \} \end{aligned} \right.$$

```
1st solution = Private`x1$67145 →
```

$$\text{Function}\left[\{\text{Private`time$67145}\}, \frac{1}{2} \left(\sqrt{2} aL + \sqrt{2} aR - 2 \text{Private`time$67145} \delta q \right)\right]$$

$$\text{1st solution w/ substitution} = \frac{1}{2} \left(\sqrt{2} aL + \sqrt{2} aR - 2 \delta q \tau \right)$$

```
Out[120]=
```

$$\frac{aL}{\sqrt{2}} + \frac{aR}{\sqrt{2}} - \delta q \tau$$

A momentum kick.

```
In[107]:= {Evolver[δp Q$sym, 1, Q$sym], Evolver[δp Q$sym, 1, P$sym]} /. QP$rules // Simplify
```

```
Out[107]:= {Q, P + δp}
```

Evolution under a force .

```
In[108]:= H$1 = - A Q$sym;
```

```
In[109]:= {Evolver[H$1, t, Q$sym], Evolver[H$1, t, P$sym]} /. QP$rules // Simplify
```

```
Out[109]:= {Q, P - A t}
```

Evolution under a force with a phase factor .

```
In[110]:= H$1$φ = A  $\frac{1}{\sqrt{2}}$  (Exp[-I φ] aR + Exp[I φ] aL);
```

```
{Evolver[H$1$φ, t, Q$sym], Evolver[H$1$φ, t, P$sym]} /. QP$rules // Simplify
```

```
Out[111]:= {Q + A t Sin[φ], P + A t Cos[φ]}
```

Evolution under a squeezing Hamiltonian.

```
In[112]:= H$2 = -  $\frac{\Delta}{2}$  I (Exp[-I φ] Mult[aR, aR] - Exp[I φ] Mult[aL, aL]);
```

```
In[113]:= ans = {Evolver[H$2, t, Q$sym], Evolver[H$2, t, P$sym]} /. QP$rules // Simplify //  
ExpToTrig // Simplify
```

```
Out[113]:= {Q Cosh[t Δ] + (Q Cos[φ] - P Sin[φ]) Sinh[t Δ], P Cosh[t Δ] - (P Cos[φ] + Q Sin[φ]) Sinh[t Δ]}
```

We can see that the squeezing is phase-dependent .

```
In[114]:= ans /. φ → 0 // TrigToExp
```

```
Out[114]:= {et Δ Q, e-t Δ P}
```

```
In[115]:= ans /. φ → π // TrigToExp
```

```
Out[115]:= {e-t Δ Q, et Δ P}
```

Evolution under the harmonic oscillator and squeezing Hamiltonians.

```
In[116]:= ans =  
{Evolver[H$0 + H$2, t, Q$sym], Evolver[H$2, t, P$sym]} /. QP$rules // Simplify //  
ExpToTrig // Simplify // Collect[#, {Q, P}] &
```

```
Out[116]:= 
$$\left\{ -\frac{P (\omega + \Delta \sin[\phi]) \sinh\left[t \sqrt{\Delta^2 - \omega^2}\right]}{\sqrt{\Delta^2 - \omega^2}} + Q \left( \cosh\left[t \sqrt{\Delta^2 - \omega^2}\right] + \frac{\Delta \cos[\phi] \sinh\left[t \sqrt{\Delta^2 - \omega^2}\right]}{\sqrt{\Delta^2 - \omega^2}} \right), \right.$$


$$\left. -Q \sin[\phi] \sinh[t \Delta] + P (\cosh[t \Delta] - \cos[\phi] \sinh[t \Delta]) \right\}$$

```

The position and momentum are somewhat simpler when $\phi = 0$ and $\phi = \pi/2$.

In[117]:= **ans /. $\phi \rightarrow 0$ // Simplify // Collect[#, {Q, P}] &**

$$\text{Out[117]} = \left\{ -\frac{P \omega \sinh\left[t \sqrt{\Delta^2 - \omega^2}\right]}{\sqrt{\Delta^2 - \omega^2}} + Q \left(\cosh\left[t \sqrt{\Delta^2 - \omega^2}\right] + \frac{\Delta \sinh\left[t \sqrt{\Delta^2 - \omega^2}\right]}{\sqrt{\Delta^2 - \omega^2}} \right), \right. \\ \left. P (\cosh[t \Delta] - \sinh[t \Delta]) \right\}$$

In[118]:= **ans /. $\phi \rightarrow \pi/2$ // Simplify // Collect[#, {Q, P}] &**

$$\text{Out[118]} = \left\{ Q \cosh\left[t \sqrt{\Delta^2 - \omega^2}\right] + \frac{P (-\Delta - \omega) \sinh\left[t \sqrt{\Delta^2 - \omega^2}\right]}{\sqrt{\Delta^2 - \omega^2}}, P \cosh[t \Delta] - Q \sinh[t \Delta] \right\}$$

Clean up

In[93]:= **Clear[aL, aR, ω , Q\$sym, P\$sym, Q, P, ω , δq , δp , A, ϕ , Δ , H\$0, H\$1, H\$1\$ ϕ , H\$2]**