tested 190817 using *SpinDynamica* 3.0.1 under *Mathematica* 11.0

```
Needs["SpinDynamica`"]
```

```
SpinDynamica version 3.0.1 loaded
```

ModifyBuiltIn: The following built-in routines have been modified in SpinDynamica:
{Chop, Dot, Duration, Exp, Expand, ExpandAll, NumericQ, Plus, Power, Simplify, Times, WignerD}.
Evaluate ??*symbol* to generate the additional definitions for *symbol*.

```
SetSpinSystem[1]
```

SetSpinSystem: the spin system has been set to $\{\{1, \frac{1}{2}\}\}$

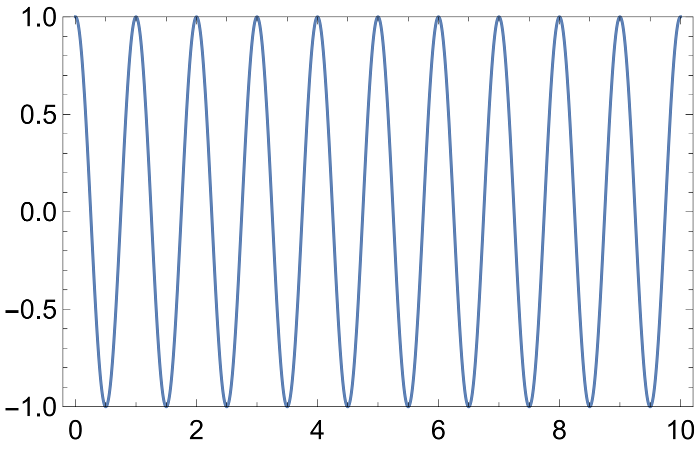
SetBasis: the state basis has been set to ZeemanBasis$[\{\{1, \frac{1}{2}\}\}$, BasisLabels → Automatic$]$.

```
SetOptions[Plot, PlotRange → {-1, 1}, Frame → True];
```

relaxation

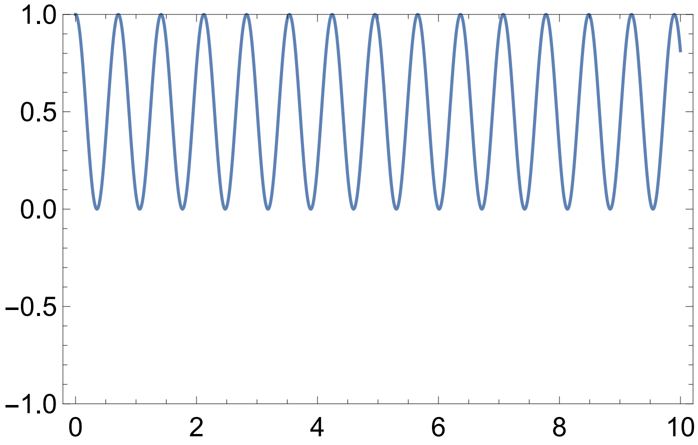
A simple trajectory of z-magnetization under a rf field along the (rotating-frame) x-axis:

```
Plot[
 Evaluate[Trajectory[opI["z"] → opI["z"], {2 π opI["x"], 10}][t]],
 {t, 0, 10}
]
```



An evolution background is used to incorporate a resonance offset

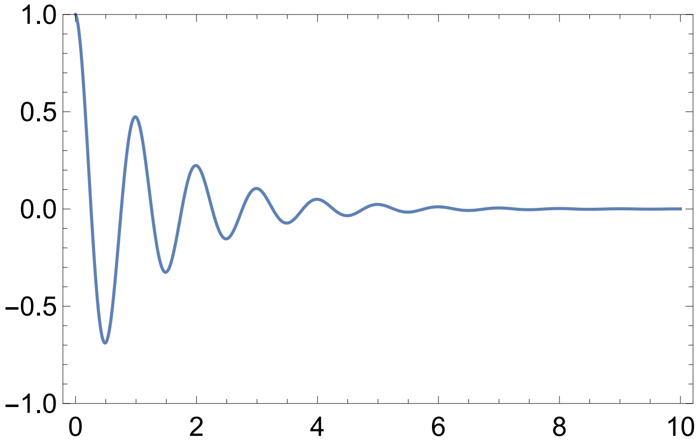
```
Plot[
 Evaluate[
  Trajectory[
    opI["z"] → opI["z"],
    {2 π opI["x"], 10},
    BackgroundGenerator → 2 π opI["z"]
   ][t]
 ],
 {t, 0, 10}
]
```



A different evolution background is used to incorporate T1 and T2 relaxation

note: The T1 relaxation is not “thermalized” so corresponds to an infinite sample temperature.

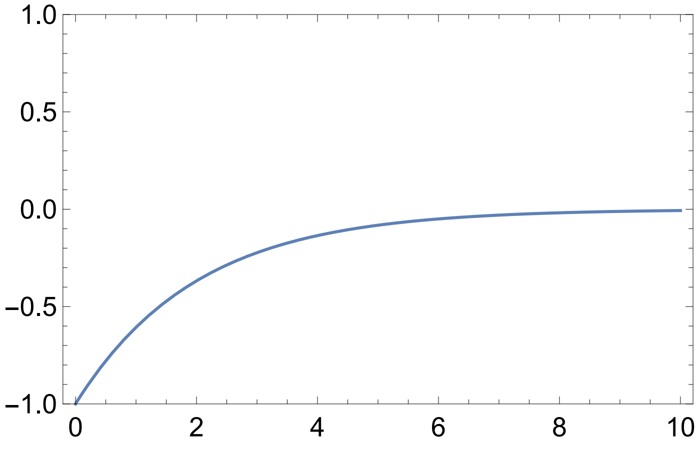
```
T1 = 2; T2 = 1;
Plot[
 Evaluate[
  Trajectory[
    opI["z"] → opI["z"],
    {2 π opI["x"], 10},
    BackgroundGenerator → - (
        (1 / T1) ProjectionSuperoperator[opI["z"]] +
         (1 / T2) ProjectionSuperoperator[opI["x"]] +
         (1 / T2) ProjectionSuperoperator[opI["y"]]
      )
    ][t]
 ],
 {t, 0, 10}
]
```



A similar calculation showing the effect of a short pulse, and then a delay

this shows the relaxation of the spins to the infinite temperature state (no Magnetization)

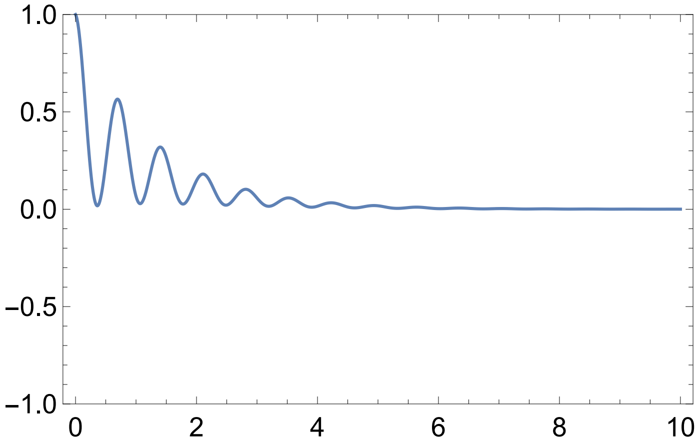
```
T1 = 2; T2 = 1;
Plot[
 Evaluate[
  Trajectory[
    opI["z"] → opI["z"],
    {RotationSuperoperator[{π, "x"}], {None, 10}},
    BackgroundGenerator → - (
        (1 / T1) ProjectionSuperoperator[opI["z"]] +
         (1 / T2) ProjectionSuperoperator[opI["x"]] +
         (1 / T2) ProjectionSuperoperator[opI["y"]]
      )
    ][t]
 ],
 {t, 0, 10}
]
```



A resonance offset and relaxation at the same time

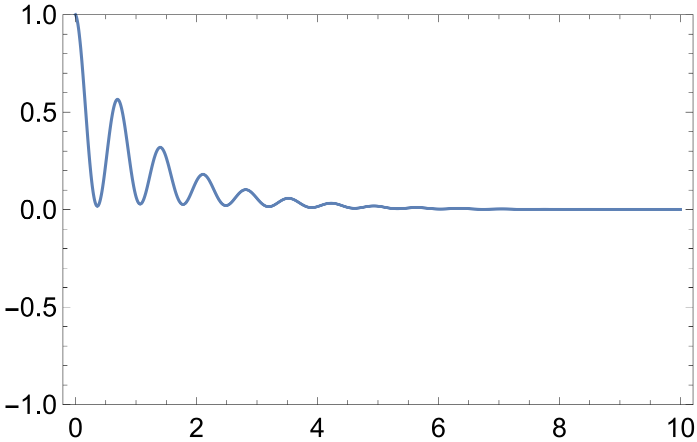
this uses CombineEvolutionGenerators to combine operators and superoperators

```
T1 = 2; T2 = 1;
Plot[
 Evaluate[
  Trajectory[
    opI["z"] → opI["z"],
    {2 π opI["x"], 10},
    BackgroundGenerator →
     CombineGenerators[
      - (
         (1 / T1) ProjectionSuperoperator[opI["z"]] +
          (1 / T2) ProjectionSuperoperator[opI["x"]] +
          (1 / T2) ProjectionSuperoperator[opI["y"]]
        ),
      2 π opI["z"]
     ]
    ][t]
  ],
 {t, 0, 10}
]
```



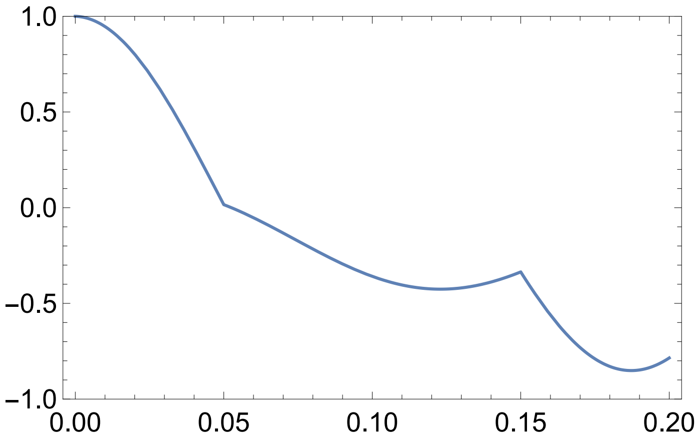
an equivalent formulation in which CombineEvolutionGenerators combines many different sorts of terms

```
T1 = 2; T2 = 1;
Plot[
 Evaluate[
  Trajectory[
    opI["z"] → opI["z"],
    {2 π opI["x"], 10},
    BackgroundGenerator →
     CombineGenerators[
      - (1 / T1) ProjectionSuperoperator[opI["z"]],
      - (1 / T2) ProjectionSuperoperator[opI["x"]],
      - (1 / T2) ProjectionSuperoperator[opI["y"]],
      2 π opI["z"]
     ]
   ][t]
 ],
 {t, 0, 10}
]
```



A composite pulse including strong relaxation

```
T1 = 2; T2 = 1;
ωnut = 2 π 5;
τ360 = (2 π / ωnut);
τ90 = τ360 / 4; τ180 = τ360 / 2;
Plot[
 Evaluate[
  Trajectory[
    opI["z"] → opI["z"],
    {{ωnut opI["x"], τ90}, {ωnut opI["y"], τ180}, {ωnut opI["x"], τ90}},
    BackgroundGenerator →
     CombineGenerators[
      - (1 / T1) ProjectionSuperoperator[opI["z"]],
      - (1 / T2) ProjectionSuperoperator[opI["x"]],
      - (1 / T2) ProjectionSuperoperator[opI["y"]],
      2 π opI["z"]
     ]
   ][t]
 ],
 {t, 0, τ360}
]
```



Thermalizing a relaxation superoperator

use a temperature of 300K at a Larmor frequency of 400MHz

note the use of the thermal equilibrium density operator *ρ*eq and the thermal equilibrium magnetization Meq

```
Γ = ThermalizeSuperoperator[
  -(
    (1/T1) ProjectionSuperoperator[opI["z"]] +
     (1/T2) ProjectionSuperoperator[opI["x"]] +
     (1/T2) ProjectionSuperoperator[opI["y"]]
   ),
  2 π 400 × 10^6 opI["z"],
  300
 ]
```

SetOperatorBasis: the operator basis has been set to ShiftAndZOperatorBasis[{{1, $\frac{1}{2}$}}, Sorted → CoherenceOrder].

```
Superoperator[<<..>>, SuperoperatorType→ None ]
```

```
ρeq = ThermalEquilibriumDensityOperator[2 π 400 × 10^6 opI["z"], 300]
```

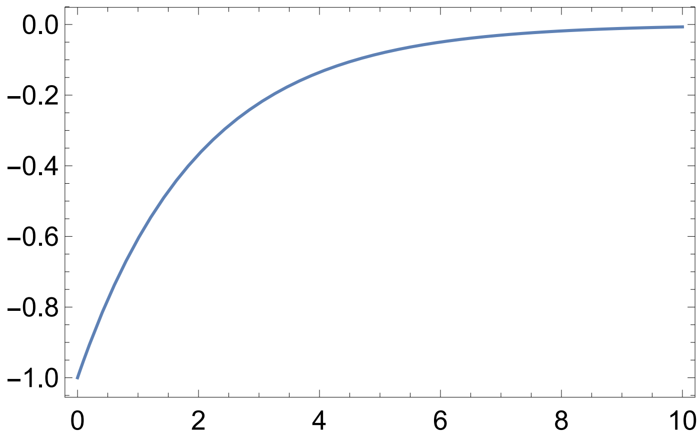
```
Operator[<< .. >> , OperatorType → Hermitian ]
```

```
Meq = OperatorAmplitude[ρeq, opI["z"]]
```

```
-0.000031995
```

inversion recovery:

```
T1 = 2; T2 = 1;
Plot[
 Evaluate[
  Trajectory[
    ρeq → opI["z"],
    {RotationSuperoperator[{π, "x"}], {None, 10}},
    BackgroundGenerator → Γ,
    NormalizationFactor → Meq
   ][t]
 ],
 {t, 0, 10},
 PlotRange → All
]
```



A composite pulse followed by relaxation

```
T1 = 2; T2 = 1;
ωnut = 2 π 5;
τ360 = (2 π / ωnut);
τ90 = τ360 / 4; τ180 = τ360 / 2;
Plot[
 Evaluate[
  Trajectory[
    ρeq → opI["z"],
    {{ωnut opI["x"], τ90}, {ωnut opI["y"], τ180}, {ωnut opI["x"], τ90}, {None, 4}},
    BackgroundGenerator →
     CombineGenerators[Γ, 2 π opI["z"]],
    NormalizationFactor → Meq
   ][t]
 ],
 {t, 0, τ360 + 4}
]
```