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*.

INADEQUATE in a 2-spin-1/2 system

```
SetSpinSystem[2]
```

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

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

chemical shift offsets, and J-coupling

```
ω0I = 2 π 1000;
ω0S = 2 π (-500);
JIS = 100;
```

define spin Hamiltonian

```
H0 = ω0I opI[1, "z"] + ω0S opI[2, "z"] + 2 π JIS opI[1].opI[2]
```

$200\,\pi\,(I_{1x}\bullet I_{2x} + I_{1y}\bullet I_{2y} + I_{1z}\bullet I_{2z}) + 2000\,\pi\,I_{1z} - 1000\,\pi\,I_{2z}$

```
MatrixRepresentation[H0] // MatrixForm
```

$$\begin{pmatrix} 550\pi & 0 & 0 & 0 \\ 0 & -1550\pi & 100\pi & 0 \\ 0 & 100\pi & 1450\pi & 0 \\ 0 & 0 & 0 & -450\pi \end{pmatrix}$$

set up INADEQUATE pulse sequence, using RotationSuperoperator to generate ideal pulses.

Note use of CoherenceOrderFiltrationSuperoperator to perform the double-quantum filtration

```
τJ = 1/JIS;
```

```
INADEQUATESequence = {
  RotationSuperoperator[{π/2, "x"}],
  {None, τJ / 4},
  RotationSuperoperator[{π, "x"}],
  {None, τJ / 4},
  RotationSuperoperator[{π/2, "x"}],
  CoherenceOrderFiltrationSuperoperator[{-2, 2}],
  RotationSuperoperator[{π/2, "x"}],
  {None, τJ / 4},
  RotationSuperoperator[{π, "x"}],
  {None, τJ / 4}
 }
```

```
{RotationSuperoperator[{1, 2}, {π/2, x}],
 {None, 1/400}, RotationSuperoperator[{1, 2}, {π, x}],
 {None, 1/400}, RotationSuperoperator[{1, 2}, {π/2, x}],
 CoherenceOrderFiltrationSuperoperator[{1, 2}, {-2, 2}],
 RotationSuperoperator[{1, 2}, {π/2, x}], {None, 1/400},
 RotationSuperoperator[{1, 2}, {π, x}], {None, 1/400}}
```

```
T = EventDuration[INADEQUATESequence]
```

$$\frac{1}{100}$$

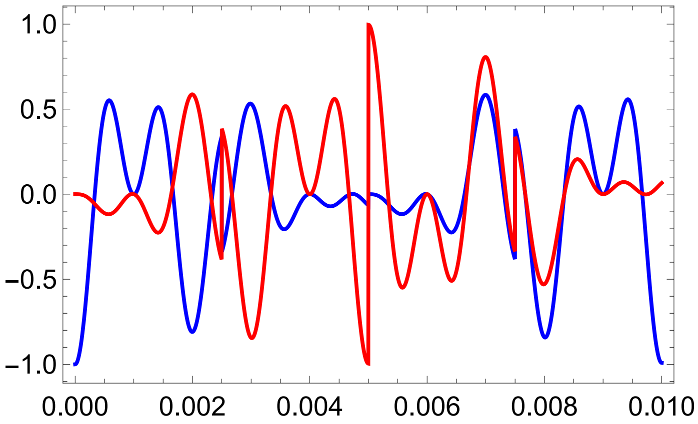
simulate trajectories of y-magnetization, and xz-antiphase terms

Note use of BackgroundGenerator to implement the H0 Hamiltonian acting continuously through the pulse sequence

```
{Iytraj, Ixztraj} = Trajectory[
  opI["z"] → {opI["y"], 2 opI[1, "x"].opI[2, "z"] + 2 opI[1, "z"].opI[2, "x"]},
  INADEQUATESequence,
  InitialTimePoint -> 0,
  BackgroundGenerator → H0
 ]
```

```
{TrajectoryFunction[ {{0, 10.×10^-3}} , <>], TrajectoryFunction[ {{0, 10.×10^-3}} , <>]}
```

```
Plot[{Iytraj[t], Ixztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Blue}, {Thick, Red}}]
```



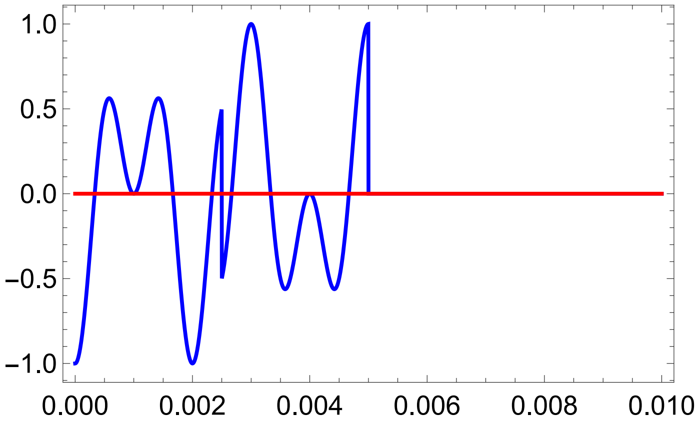
note how the y-magnetization is recovered at the end of the sequence

repeat simulation, but this time with no J-coupling

```
{Iytraj, Ixztraj} = Trajectory[
  opI["z"] → {opI["y"], 2 opI[1, "x"].opI[2, "z"] + 2 opI[1, "z"].opI[2, "x"]},
  INADEQUATESequence, InitialTimePoint -> 0,
  BackgroundGenerator → ω0I opI[1, "z"] + ω0S opI[2, "z"]]
```

```
{TrajectoryFunction[ {{0, 10. × 10^-3}} , <>], TrajectoryFunction[ {{0, 10. × 10^-3}} , <>]}
```

```
Plot[{Iytraj[t], Ixztraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Blue}, {Thick, Red}}]
```



note that the signal is now killed by the double-quantum filter