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*.

non-refocussed INEPT (with idealized pulses)

```
SetSpinSystem[{{"I", 1/2}, {"S", 1/2}}]
```

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

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

chemical shift offsets, and J-coupling

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

define spin Hamiltonian

weakly-coupled form for the heteronuclear system

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

$200\pi\,(I_z \cdot S_z) + 2000\pi\, I_z - 1000\pi\, S_z$

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

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

define INEPT sequence

written in **LeftToRight** chronological order

RotationSuperoperator is used to implement ideal pulses of zero duration

```
τJ = 1/JIS;
```

```
INEPT = {
  RotationSuperoperator["I", {π / 2, "x"}],
  {None, τJ / 4},
  RotationSuperoperator[{π, "x"}],
  {None, τJ / 4},
  RotationSuperoperator["I", {π / 2, "y"}],
  RotationSuperoperator["S", {π / 2, "x"}]}
```

$\{\text{RotationSuperoperator}[\{\text{I}\}, \{\frac{\pi}{2}, \text{x}\}], \{\text{None}, \frac{1}{400}\},$
$\text{RotationSuperoperator}[\{\text{I}, \text{S}\}, \{\pi, \text{x}\}], \{\text{None}, \frac{1}{400}\},$
$\text{RotationSuperoperator}[\{\text{I}\}, \{\frac{\pi}{2}, \text{y}\}], \text{RotationSuperoperator}[\{\text{S}\}, \{\frac{\pi}{2}, \text{x}\}]\}$

```
T = EventDuration[INEPT]
```

$\frac{1}{200}$

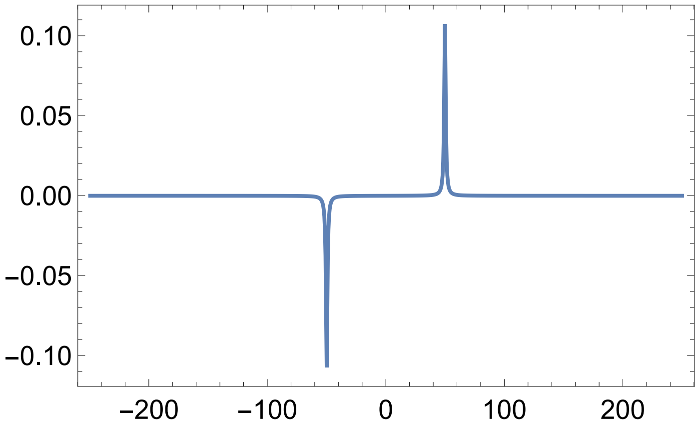
calculate S-spin spectrum induced at end of sequence

```
ListPlot[
 Re@FT@Signal1D[{0, 1, 2 × 10^-3},
    BackgroundGenerator → H0,
    InitialDensityOperator → opI["I", "z"],
    Preparation -> INEPT,
    Observable → - (I / 2) opI["S", "-"],
    Chronology → "LeftToRight"
   ],
 Joined → True, PlotRange → All, Frame → True, PlotStyle → Thick
]
```

Signal1D: Using SignalCalculationMethod → Diagonalization

Signal1D: the last sampling point has been dropped in order to get an even number of points.

Signal1D: Using LineBroadening → $2\pi \times 1.46587$ rad s^{-1}.



refocussed INEPT (with idealized pulses)

define INEPT sequence

```
RefocussedINEPT = {
   RotationSuperoperator["I", {π/2, "x"}],
   {None, τJ / 4},
   RotationSuperoperator[{π, "x"}],
   {None, τJ / 4},
   RotationSuperoperator["I", {π/2, "y"}],
   RotationSuperoperator["S", {π/2, "y"}],
   {None, τJ / 4},
   RotationSuperoperator[{π, "x"}],
   {None, τJ / 4}
  };
```

```
T = EventDuration[RefocussedINEPT]
```

$\frac{1}{100}$

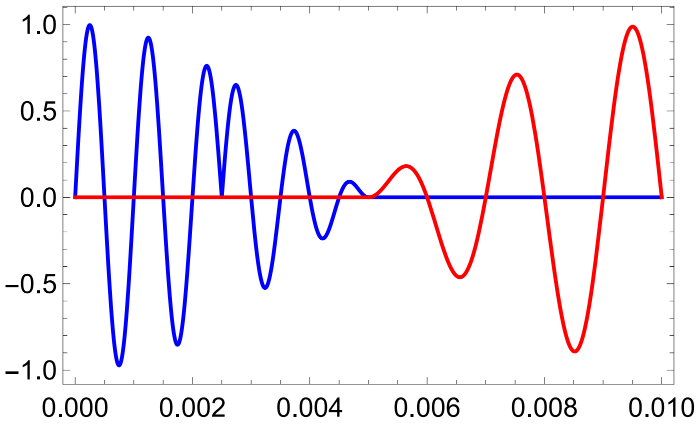
calculate and plot trajectories of the in-phase magnetization components

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

```
{Ixtraj, Sxtraj} = Trajectory[opI["I", "z"] → {opI["I", "x"], opI["S", "x"]},
  RefocussedINEPT, InitialTimePoint -> 0, BackgroundGenerator → H0,
  Chronology → "LeftToRight"]
```

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

```
Plot[{Re@Ixtraj[t], Re@Sxtraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Blue}, {Thick, Red}}]
```



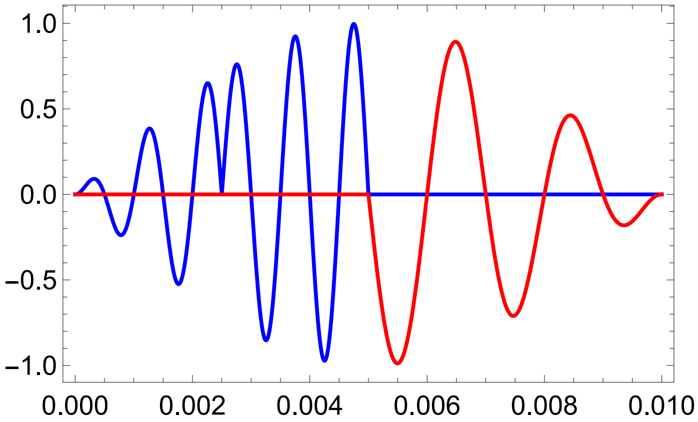
note how the magnetization is transformed from one spin to the other spin

calculate and plot trajectories of the out-of-phase magnetization components

```
{IySztraj, IzSytraj} = Trajectory[
  opI["I", "z"] → {2 opI["I", "y"].opI["S", "z"], 2 opI["I", "z"].opI["S", "y"]},
  RefocussedINEPT, InitialTimePoint -> 0, BackgroundGenerator → H0]
```

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

```
Plot[{Re@IySztraj[t], Re@IzSytraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Blue}, {Thick, Red}}]
```



this shows how the antiphase terms build up and are transformed in the middle of the sequence

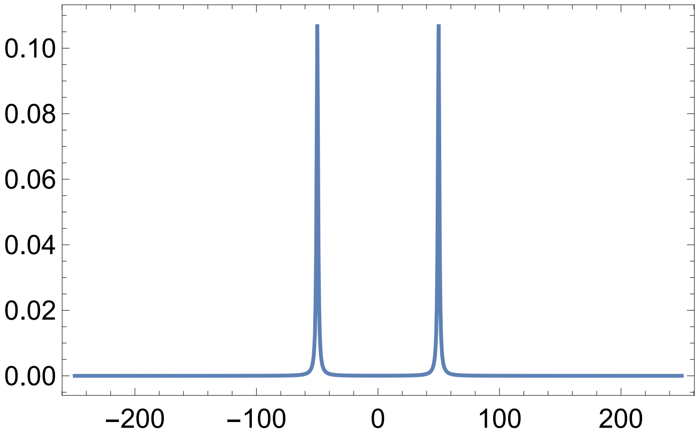
calculate S-spin spectrum induced at end of sequence

```
ListPlot[
 Re@FT[Signal1D[{0, 1, 2 × 10^-3},
    BackgroundGenerator → H0,
    InitialDensityOperator → opI["I", "z"],
    Preparation -> RefocussedINEPT,
    Observable → - (I / 2) opI["S", "-"]
   ]],
 Joined → True, PlotRange → All, Frame → True, PlotStyle → Thick
]
```

Signal1D: Using SignalCalculationMethod → Diagonalization

Signal1D: the last sampling point has been dropped in order to get an even number of points.

Signal1D: Using LineBroadening → $2\pi \times 1.46587$ rad s^{-1}.



refocussed INEPT (with realistic pulses)

chemical shift offsets, and J-coupling

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

$2000\,\pi$

$-1000\,\pi$

100

define spin Hamiltonian

weakly-coupled form for the heteronuclear system

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

$200\,\pi\,(I_z \cdot S_z) + 2000\,\pi\,I_z - 1000\,\pi\,S_z$

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

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

define rf fields and pulse durations

```
ωnutI = 2 π 30 × 10^3
```

60 000 π

```
ωnutS = 2 π 20 × 10^3
```

40 000 π

```
τ90I = (π / 2) / ωnutI;
τ180I = π / ωnutI;
```

```
τ90S = (π / 2) / ωnutS;
τ180S = π / ωnutS;
```

```
τJ = 1 / JIS;
```

implement short routine for aligning simultaneous events

the events are centred with respect to each other, with the longer event overlapping on both sides.

```
AlignEvents[{HA_, τA_}, {HB_, τB_}] /; (τA > τB) :=
  Sequence[{HA, (τA - τB) / 2}, {HA + HB, τB}, {HA, (τA - τB) / 2}];
```

```
AlignEvents[{HA_, τA_}, {HB_, τB_}] /; (τA < τB) :=
  Sequence[{HB, (τB - τA) / 2}, {HA + HB, τA}, {HB, (τB - τA) / 2}];
```

```
AlignEvents[{HA_, τA_}, {HB_, τB_}] /; (τA == τB) := {HA + HB, τA};
```

define INEPT sequence

The AlignEvents routine is used to handle simultaneous pulses of different durations.

```
FinitePulseRefocussedINEPT = {
  {ωnutI opI["I", "x"], τ90I},
  {None, τJ / 4},
  AlignEvents[
   {ωnutS opI["S", "x"], τ180S},
   {ωnutI opI["I", "x"], τ180I}
  ],
  {None, τJ / 4},
  AlignEvents[
   {ωnutS opI["S", "y"], τ90S},
   {ωnutI opI["I", "y"], τ90I}
  ],
  {None, τJ / 4},
  AlignEvents[
   {ωnutS opI["S", "x"], τ180S},
   {ωnutI opI["I", "x"], τ180I}
  ],
  {None, τJ / 4}
 }
```

$$\left\{\left\{60\,000\,\pi\,I_x,\ \frac{1}{120\,000}\right\},\ \left\{\text{None},\ \frac{1}{400}\right\},\ \left\{40\,000\,\pi\,S_x,\ \frac{1}{240\,000}\right\},\ \left\{60\,000\,\pi\,I_x + 40\,000\,\pi\,S_x,\ \frac{1}{60\,000}\right\},\ \left\{40\,000\,\pi\,S_x,\ \frac{1}{240\,000}\right\},\ \left\{\text{None},\ \frac{1}{400}\right\},\ \left\{40\,000\,\pi\,S_y,\ \frac{1}{480\,000}\right\},\ \left\{60\,000\,\pi\,I_y + 40\,000\,\pi\,S_y,\ \frac{1}{120\,000}\right\},\ \left\{40\,000\,\pi\,S_y,\ \frac{1}{480\,000}\right\},\ \left\{\text{None},\ \frac{1}{400}\right\},\ \left\{40\,000\,\pi\,S_x,\ \frac{1}{240\,000}\right\},\ \left\{60\,000\,\pi\,I_x + 40\,000\,\pi\,S_x,\ \frac{1}{60\,000}\right\},\ \left\{40\,000\,\pi\,S_x,\ \frac{1}{240\,000}\right\},\ \left\{\text{None},\ \frac{1}{400}\right\}\right\}$$

```
T = EventDuration[FinitePulseRefocussedINEPT]
```

$$\frac{2417}{240\,000}$$

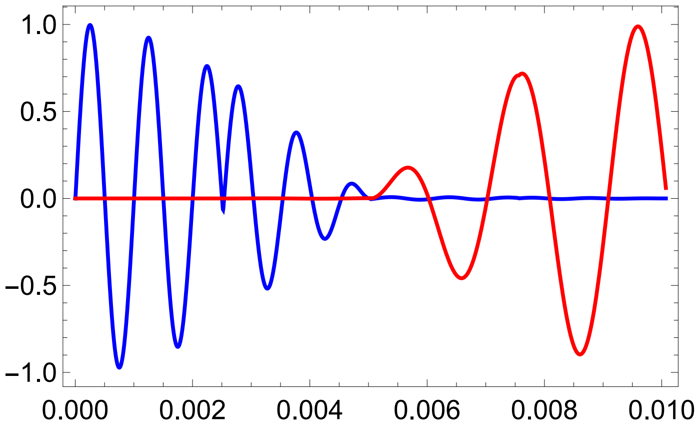
calculate and plot trajectories of the in-phase magnetization components

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

```
{Ixtraj, Sxtraj} =
 Trajectory[opI["I", "z"] → {opI["I", "x"], opI["S", "x"]}, FinitePulseRefocussedINEPT,
  InitialTimePoint -> 0, BackgroundGenerator → H0, Chronology → "LeftToRight"]
```

$\{\text{TrajectoryFunction}[\ \{\{0,\ 10.0708\times 10^{-3}\}\}\ ,\ \text{<>}],$
$\ \text{TrajectoryFunction}[\ \{\{0,\ 10.0708\times 10^{-3}\}\}\ ,\ \text{<>}]\}$

```
Plot[{Re@Ixtraj[t], Re@Sxtraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Blue}, {Thick, Red}}]
```

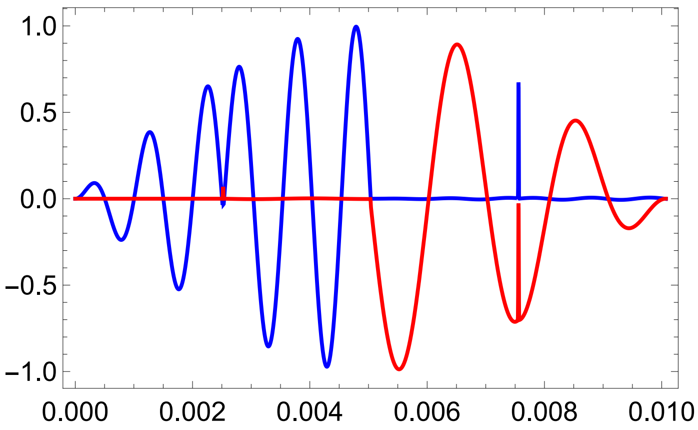


calculate and plot trajectories of the out-of-phase magnetization components

```
{IySztraj, IzSytraj} = Trajectory[
  opI["I", "z"] → {2 opI["I", "y"].opI["S", "z"], 2 opI["I", "z"].opI["S", "y"]},
  FinitePulseRefocussedINEPT, InitialTimePoint -> 0, BackgroundGenerator → H0]
```

```
{TrajectoryFunction[ {{0, 10.0708 × 10^-3}} , <>],
 TrajectoryFunction[ {{0, 10.0708 × 10^-3}} , <>]}
```

```
Plot[{Re@IySztraj[t], Re@IzSytraj[t]}, {t, 0, T}, Frame → True,
 PlotRange → All, PlotStyle → {{Thick, Blue}, {Thick, Red}}]
```



calculate S-spin spectrum induced at end of sequence

note the small phase shift associated with the finite pulse durations

```
ListPlot[
 Re@FT[Signal1D[{0, 1, 2 × 10^-3},
     BackgroundGenerator → H0,
     InitialDensityOperator → opI["I", "z"],
     Preparation -> FinitePulseRefocussedINEPT,
     Observable → - (I / 2) opI["S", "-"]
    ]],
 Joined → True, PlotRange → All, Frame → True, PlotStyle → Thick
]
```

Signal1D: Using SignalCalculationMethod → Diagonalization

Signal1D: the last sampling point has been dropped in order to get an even number of points.

Signal1D: Using LineBroadening → 2π × 1.46587 rad s^{-1}.