Important function calls:

Creates Hamiltonian and initial state from input:

```
[system, state, options]=setup(system, options)
```

Set up pulses:

```
[experiment,options] = triple(sequence, options)
```

Propagates system:

```
[state, detected_signal, experiment, options, tables] = homerun(state,
system, experiment, options, tables)
```

Down conversion of signal:

```
[signal_dc,filter] = strike(signal_sf, t, down, options)
```

Input

The structure `system`.

Name	Description	Example
`.nu01`	Resonance frequency of the 1st spin in GHz, given in the AWG frame.	`system.nu01=1.2;`
`.nu02`	Resonance frequency of the 2nd spin in GHz, given in the AWG frame	`system.nu02=-0.014;`
`.ns1`	Spin quantum number of 1st spin.	`system.ns1=0.5;`
`.ns2`	Spin quantum number of 2nd spin.	`system.ns2=3/2;`
`.T1`	Longitudinal relaxation times in ns. Can be a matrix or a scalar.	`system.T1=5000;` `system.T1=[0 5000; 0 0]`
`.T2`	Transversal relaxation times in ns. Can be a matrix or a scalar.	`system.T2=1000;` `system.T2=[0 1000; 0 0]`
`.default_T1`	Default longitudinal relaxation time in ns. Scalar, used for all non-specified transitions if `.T1` is a matrix.	`system.default_T1=8000;`
`.default_T2`	Default transversal relaxation time in ns. Scalar, used for all non-specified transitions if `.T2` is a matrix.	`system.default_T2=500;`
`.A`	Secular hyperfine coupling term in GHz.	`system.A=0.045;`
`.B`	Pseudo-secular hyperfine coupling term in GHz,	`system.B=0.015;`
`.sops`	Structure containing spin operators, is created by `setup`.	
`.spins`	Number of spins in the system, can be given else created by `setup`.	`system.spins=2;`
`.ham`	Free evolution Hamiltonian of the system in matrix form. Usually calculated by `setup`.	
`.eq`	Equilibrium state as density matrix, created by `setup`.	
`.R`	Relaxation matrix in Hilbert space, created by `setup` for two-level system only.	
`.gamma`	Relaxation super operator, created by `setup`.	

The structure `sequence`.

Name	Description	Example
`.tp`	Vector containing the lengths of all events (pulses, delays and detection) in ns.	`sequence.tp= [50 10 25 100];`
`.nu1`	Vector with pulse strengths in MHz. Zero for no pulse. Position corresponds to events in `.tp`.	`sequence.nu1=[15 0 10];`
`.beta`	The flip angles of the pulses, ignored if zero or `.nu1` for the same event available.	`sequence.beta= [pi/2 0 pi];`
`.t_rise`	Vector or scalar with rise times for the pulses in ns.	`sequence.t_rise=[15 0];`
`.frq`	Excitation bandwidth in GHz. Elements correspond to initial and final frequency of sweep. Has to be a cell-array to assign different sweeps to different pulses.	`sequence.frq=[1.0 1.5];` `sequence.frq= {[1 2] [] [2 1]};` `sequence.frq=[1.0 1.0];`
`.phase`	Vector with phase of the pulse in rad.	`sequence.phase= [pi/2 0 pi];`
`.excite`	Boolean vector/matrix or cell array containing Boolean vector, position in vector corresponds to the index of spin as defined in system. Can also be a custom excitation operator in matrix form.	`sequence.excite=[1 0];` `sequence.excite= {[1 0] [] [1 1]};`
`.pcycle`	Cell array with matrices with rows corresponding to phase cycle step. Firs value is the phase, second value is the weighting.	`sequence.pcycle{3}= [0, 1; pi, -1];`

stephan.pribitzer@phys.chem.ethz.ch

The structure options.

Name	Description	Example
.det_op	Cell array with detection operators.	options.det_op={'-Sz'}; options.det_op={'S1p'};
.relaxation	Switches relaxation on and off, 1 or 0.	options.relaxation=1;
.propagator	Tells SPIDYAN to use propagator method with relaxation on. This is only possible for a two-level system.	options.propagator=1;
.direct_evolutio	Can be used to speed up simulation. Vector which contains position of free evolution events in .tp which can be propagated in a single step.	options.direct_evolution=[2];
.no_detection	Also used to speed up simulation. Vector which contains position of free evolution events in .tp to be excluded from detection.	options.no_detection=[1 2 3];
.no_dc	Cell, contains index of detection operator or string. Tells SPIDYAN to skip down conversion for indicated signals.	options.no_dc={2,'Sp'}
.down_conversion	If true, signal is down converted from LO or lab frame.	options.filtering=1;
.cutoff_freq	Cutoff frequency of the filter for down conversion in MHz.	options.cutoff_freq=100;
.display_filter	If true, the response function of the filter used for down conversion is displayed in a figure.	options.display_filter=1;
.ui	If true, relaxation times can be input in a pop-up window in matrix form.	options.ui=1;
.complex_excitation	True or false. The excitation Hamiltonian can be changed to allow for a complex excitation. Computationally expensive.	options.complex_excitation=1;
.LO	Frequency of the local oscillator in GHz. Required for simulations with resonator or ZFS.	options.LO=8;
.labframe	If true, SPIDYAN simulates in the lab frame (system.nu01+options.LO).	options.labframe=0;
.awg	A structure containing arbitrary waveform generator parameters. If not complete, missing values are loaded from the defaults.	
.awg.s_rate	Sampling rate of the AWG in GS/s, default is 12 GS/s.	options.awg.s_rate=12;
.awg.vert_res	Vertical resolution of the AWG in bit, default is 10 bit.	options.awg.vert_res=10;
.resonator	Structure containing resonator parameters	
.resonator.active	Vector with event indices to which resonator shall be applied.	options.resonator.active=[1 3];
.resonator.Ql	Quality factor of the resonator.	options.resonator.Ql=50;
.resonator.f0	Resonance frequency of the resonator in GHz in the lab frame.	options.resonator.f0=9;
.resonator.s_rate	Sampling rate for the creation of the resonator profile. Should be at least three times options.awg.s_rate.	options.resonator.s_rate=96;
.resonator.range	Can be used to use custom resonator profiles. Frequency axis in GHz.	
.resonator.nu1	Resonator profile corresponding to .resonator.range.	

Output

Name	Description
state	Density matrix describing the system after the pulse experiment.
detected_signal	Structure, containing the detected signal.
detected_signal.sf	Signal of the entire simulation as a vector/matrix in the simulation frame (lab or AWG frame) with traces of the different detection operators as rows.
detected_signal.dc	Down converted signal of the entire simulation, only available if options.down_conversion=1.
detected_signal.t	Time axis for detected_signal.rf.
detected_signal.signals.sf	Cell array containing the traces for each individual event in the AWG or lab frame.
detected_signal.signals.dc	Cell array containing the traces for each individual event after down conversion.
detected_signal.signals.t	Cell array containing the time axis for each individual event.

stephan.pribitzer@phys.chem.ethz.ch