Using the SACNA package - Calvin LES model

The purpose of this document is to give a short tutorial for the SACNA package.
Let's start by bringing the package functions to this notebook.

In[ ]:=
```
Quiet[ClearAll["Global`*"]];
Quiet[Remove["Global`*"]];
SetDirectory[NotebookDirectory[]];

Quiet[Get["../SACNA.wl"]]
```

Now let's input the reactions and rates lists of this model. If we input the rates list as an empty list, SACNA will assign rates by default. Reactions must be in terms of D-species, L-species, Z-species (achiral species), and the empty specie N1.

In[ ]:=
```
(*Calvin-LES model*)
reactions = {"L1->D1", "D1->L1", "L1->L2", "L2->L1", "D1->D2",
   "D2->D1", "L1+L2->2L2", "2L2->L1+L2", "D1+D2->2D2", "2D2->D1+D2",
   "L1+L2->L2+D2", "L2+D2->L1+L2", "D1+D2->D2+L2", "D2+L2->D1+D2"};
rates =
  {};
```

Now we can run the semialgebraic analysis of the model by using the `RunSemiAlgebraicAnalysis` function. The first parameter corresponds to the reactions' list, the second parameter corresponds to the rates' list, and the last parameter corresponds to time in seconds (the Collins' algorithm may take so much time to find a solution). The function will ask for the Routh-Hurwitz condition number. Considering the first and last numbers will be faster, because this conditions are shorter than the others. For this particular example we will be choose the first condition.

In[ ]:=
```
time = 60;
cadSolutions = RunSemiAlgebraicAnalysis[reactions, rates, time]
```

Out[ ]=
$$D1 > 0 \&\& 0 < D2 < D1 \&\& k1 > 0 \&\& k2 > 0 \&\& k3 > 0 \&\& k4 > \frac{2\,k1 + k2 + k3}{D1 - D2} \&\& 0 < k5 < \frac{-2\,k1 - k2 - k3 + D1\,k4 - D2\,k4}{2\,D2} \&\& 0 < k6 < \frac{-2\,k1 - k2 - k3 + D1\,k4 - D2\,k4 - 2\,D2\,k5}{D1 + D2} \&\& k7 == \frac{D1\,k2 - D2\,k3 + D1\,D2\,k4 - D2^2\,k5 + D1\,D2\,k6}{D2^2}$$

Since the algorithm found a solution, let's find some particular solutions by using the `FindInstance` command. Note that the solution doesn't contain an expression for the L-species. It's because we are assuming the racemic condition.

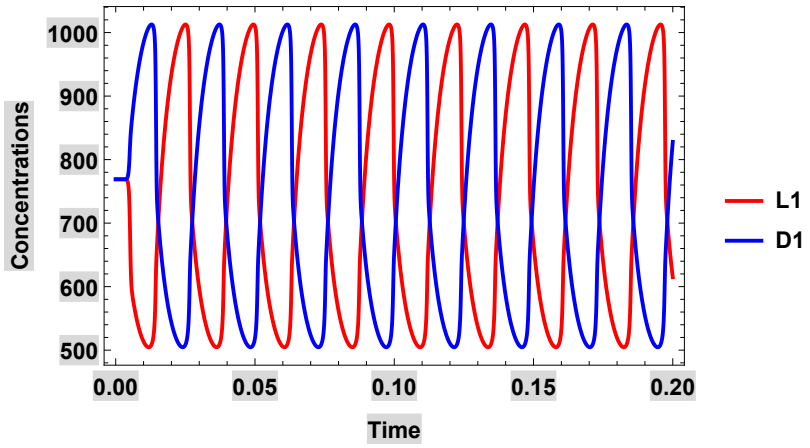
In[•]:=
```
numberOfSamples = 10; (*feel free to change*)
samplesList = FindInstance[cadSolutions,
                encuentra caso
   DeleteCases[DeleteDuplicates@Cases[cadSolutions, _Symbol, Infinity],
   elimina casos  elimina repeticiones  casos                  infinito
    Alternatives @@ {GreaterEqual, Greater, Less, LessEqual}], numberOfSamples];
    alternativas  mayor o igual  mayor  menor  menor o igual
sampleNumber = 8;  (*feel free to change*)
samplesList[[sampleNumber]]
```

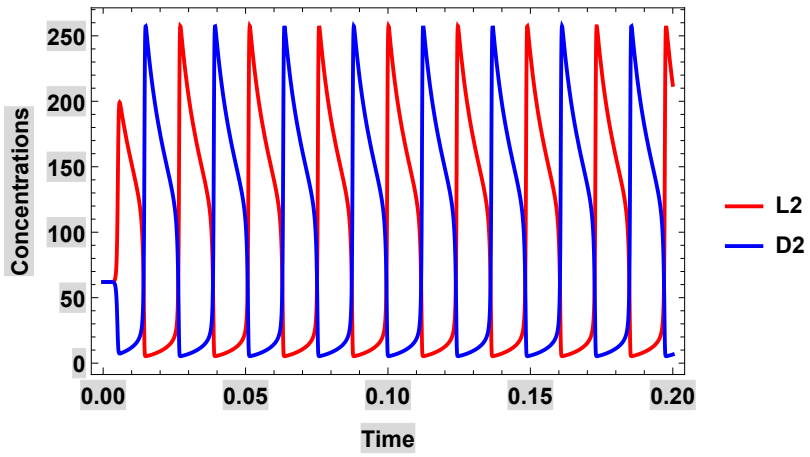
Out[•]=
$$\left\{D1 \to 769,\ D2 \to 62,\ k1 \to 24,\ k2 \to 74,\ k3 \to 63,\ k4 \to 11,\ k5 \to 29,\ k6 \to \frac{5}{11},\ k7 \to \frac{1\,341\,048}{10\,571}\right\}$$

Now we are ready to using the SACNA's system simulator with the `ReactionSystemSimulator` function

In[•]:=
```
simulationTimeMin = 0;
simulationTimeMax = 0.2; (*feel free to change*)
graficss = ReactionSystemSimulator[reactions, rates,
   samplesList[[sampleNumber]], 0.00001, t, simulationTimeMin, simulationTimeMax];
```

Species Concentrations Graphic





SACNA also allows to export a simulation results to ChemKinLator simulator files with the function ExportToChemKinLator.

In[ ]:=
```
content = ExportToChemKinLator[reactions, rates, samplesList[[sampleNumber]],
    1/1 000 000 000 000 000, 1/1 000 000 000 000 000, 0, simulationTimeMax, 1000];
```

Export: First argument $Canceled is not a valid file specification.

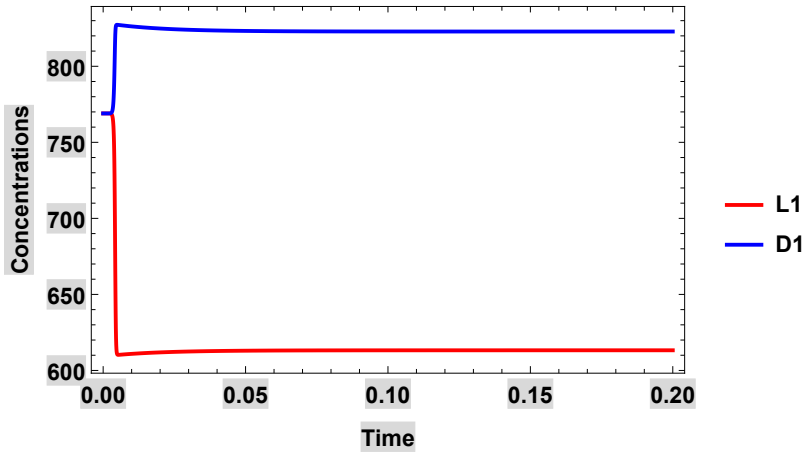
We can do the same analysis with the second Routh-Hurwitz condition.

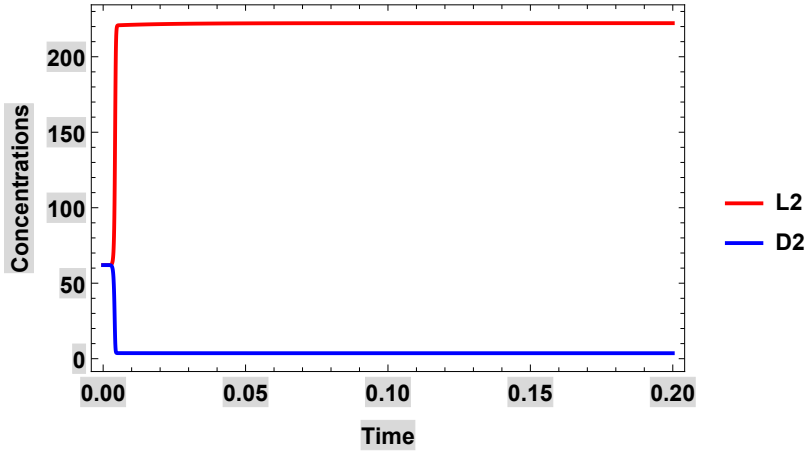
In[ ]:=
```
cadSolutions = RunSemiAlgebraicAnalysis[reactions, rates, time]
numberOfSamples = 10; (*feel free to change*)
samplesList = FindInstance[cadSolutions,
                encuentra caso
   DeleteCases[DeleteDuplicates@Cases[cadSolutions, _Symbol, Infinity],
   elimina casos  elimina repeticiones  casos                  infinito
    Alternatives @@ {GreaterEqual, Greater, Less, LessEqual}], numberOfSamples];
    alternativas        mayor o igual   mayor   menor  menor o igual
sampleNumber = 8;  (*feel free to change*)
samplesList[[sampleNumber]]
simulationTimeMin = 0;
simulationTimeMax = 0.2; (*feel free to change*)
graficss = ReactionSystemSimulator[reactions, rates,
   samplesList[[sampleNumber]], 0.00001, t, simulationTimeMin, simulationTimeMax];
```

Out[ ]=
$$D1 > 0 \&\& D2 > 0 \&\& k1 > 0 \&\& k2 > 0 \&\& k3 > 0 \&\& k4 > \frac{k3}{D1} \&\& 0 < k5 < \frac{-k3 + D1\,k4}{2\,D2} \&\&$$
$$0 < k6 < \frac{-k1\,k3 + D1\,k1\,k4 - 2\,D2\,k1\,k5}{D1\,k1 + D1\,k2 + D2\,k3 + 2\,D2^2\,k5} \&\& k7 == \frac{D1\,k2 - D2\,k3 + D1\,D2\,k4 - D2^2\,k5 + D1\,D2\,k6}{D2^2}$$

Out[ ]=
$$\left\{D1 \to 769,\ D2 \to 62,\ k1 \to 24,\ k2 \to 74,\ k3 \to 63,\ k4 \to 11,\ k5 \to 29,\ k6 \to \frac{2}{53},\ k7 \to \frac{12\,396\,201}{101\,866}\right\}$$

Species Concentrations Graphic





The user would like to know which rate constant corresponds to some reactions. It can be done with the function `GetReactionsAndRates`.

In[ ]:=
```
GetReactionsAndRates[reactions, rates] // MatrixForm
```
forma de matriz

Out[ ]//MatrixForm=

L1->D1	k1
D1->L1	k1
L1->L2	k2
L2->L1	k3
D1->D2	k2
D2->D1	k3
L1+L2->2L2	k4
2L2->L1+L2	k5
D1+D2->2D2	k4
2D2->D1+D2	k5
L1+L2->D2+L2	k6
D2+L2->L1+L2	k7
D1+D2->D2+L2	k6
D2+L2->D1+D2	k7