BPL_CHO_Perfusion_cspr_openloop script with PyFMI

The key library PyFMI is installed.

After the installation a small application BPL_CHO_Pefusion_cspr_openloop is loaded and run. You can continue with this example if you like.

```
!lsb_release -a # Actual VM Ubuntu version used by Google
```

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.3 LTS
Release:        22.04
Codename:       jammy
```

```
%env PYTHONPATH=
```

```
env: PYTHONPATH=
```

Double-click (or enter) to edit

```
!wget https://repo.anaconda.com/miniconda/Miniconda3-py310_23.1.0-1-Linux-x86_64.sh
!chmod +x Miniconda3-py310_23.1.0-1-Linux-x86_64.sh
!bash ./Miniconda3-py310_23.1.0-1-Linux-x86_64.sh -b -f -p /usr/local
import sys
sys.path.append('/usr/local/lib/python3.10/site-packages/')
```

```
--2024-05-24 09:59:44--  https://repo.anaconda.com/miniconda/Miniconda3-py310_23.1.0-1-Li
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.191.158, 104.16.32.241, 2606:47
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.191.158|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 74403966 (71M) [application/x-sh]
Saving to: ‘Miniconda3-py310_23.1.0-1-Linux-x86_64.sh’

Miniconda3-py310_23 100%[===================>]  70.96M   141MB/s    in 0.5s

2024-05-24 09:59:45 (141 MB/s) - ‘Miniconda3-py310_23.1.0-1-Linux-x86_64.sh’ saved [74403

PREFIX=/usr/local
Unpacking payload ...

Installing base environment...


Downloading and Extracting Packages


Downloading and Extracting Packages

Preparing transaction: done
Executing transaction: done
installation finished.
```

```
!conda update -n base -c defaults conda --yes
```

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

```
!conda --version
!python --version
```

```
conda 23.1.0
Python 3.10.14
```

```
!conda install -c conda-forge pyfmi --yes # Install the key package
```

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

BPL_CHO_Perfusion_cspr_openloop setup

Now specific installation and the run simulations. Start with connecting to Github. Then upload the two files:

- FMU - BPL_CHO_Perfusion_cspr_openloop_linux_om_me.fmu
- Setup-file - BPL_CHO_Perfusion_cspr_openloop_explore

```
# Filter out DepracationWarnings for 'np.float as alias' is needed - wish I could make filte
import warnings
warnings.filterwarnings("ignore")
```

```
%%bash
git clone https://github.com/janpeter19/BPL_CHO_Perfusion
```

```
Cloning into 'BPL_CHO_Perfusion'...
```

```
%cd BPL_CHO_Perfusion
```

```
/content/BPL_CHO_Perfusion
```

BPL_CHO_Perfusion_cspr_openloop - demo

Author: Jan Peter Axelson

```
run -i BPL_CHO_perfusion_cspr_openloop_explore.py
```

```
Linux - run FMU pre-comiled OpenModelica 1.23.0-dev

Model for bioreactor has been setup. Key commands:
 - par()       - change of parameters and initial values
 - init()      - change initial values only
 - simu()      - simulate and plot
 - newplot()   - make a new plot
 - show()      - show plot from previous simulation
 - disp()      - display parameters and initial values from the last simulation
 - describe()  - describe culture, broth, parameters, variables with values/units

Note that both disp() and describe() takes values from the last simulation
and the command process_diagram() brings up the main configuration

Brief information about a command by help(), eg help(simu)
Key system information is listed with the command system_info()
```

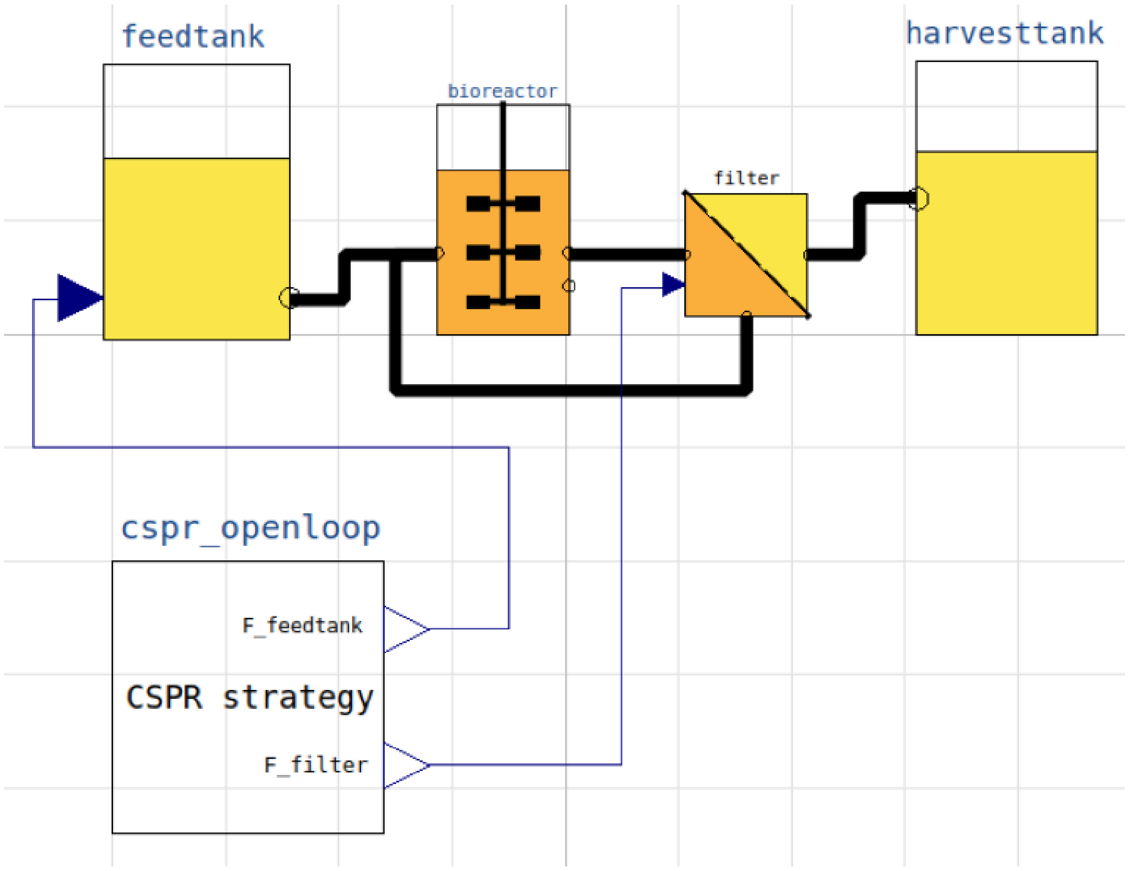
```
%matplotlib inline
plt.rcParams['figure.figsize'] = [25/2.54, 20/2.54]
```

1 About the process model

We can get information about the process and liquid phase by the command describe(). Here is no gas-phase included. This command can also be used to bring up information about a specific variable or parameter. However, you should use describe() after a simulation to get the valued used during the simulation.

```
process_diagram()
```

```
No processDiagram.png file in the FMU, but try the file on disk.
```



```
describe('culture'); print(); #describe('liquidphase')                                    # Pun
```

```
Reactor culture CHO-MAb - cell line HB-58 American Culture Collection ATCC
```

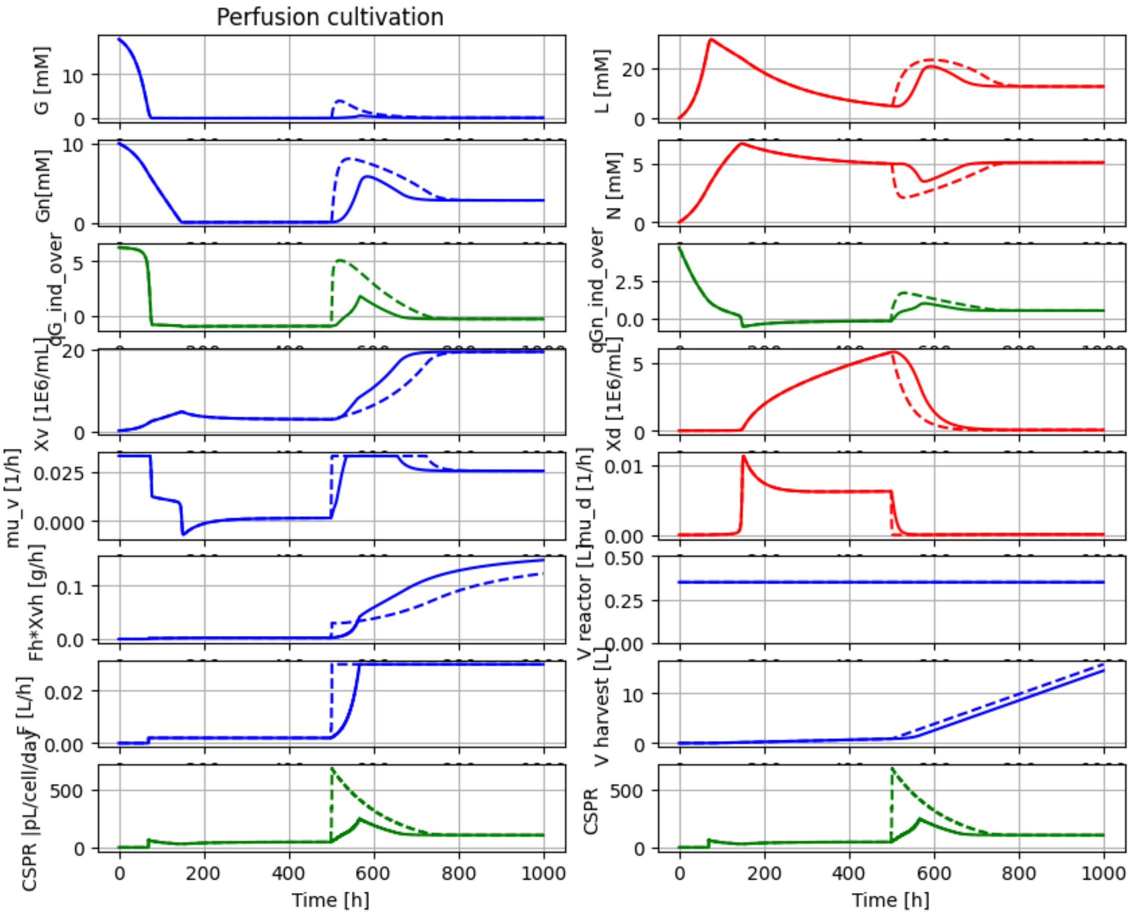
The molecular weight of the recombinant protein (MAb) is somewhat arbitrarly chosen and the value not used in the simulations.

```
describe('MSL')
```

```
MSL: 3.2.3 - used components: RealInput, RealOutput, CombiTimeTable, Types
```

```
newplot('Perfusion cultivation', plotType='Extended')

par(samplePeriod=1); par(F1=0.0020)           # General parameters
par(mu_ref=0.04); simu()                      # First simulation (solid)
par(mu_ref=1); simu()                         # Second simulation (dashed)
```



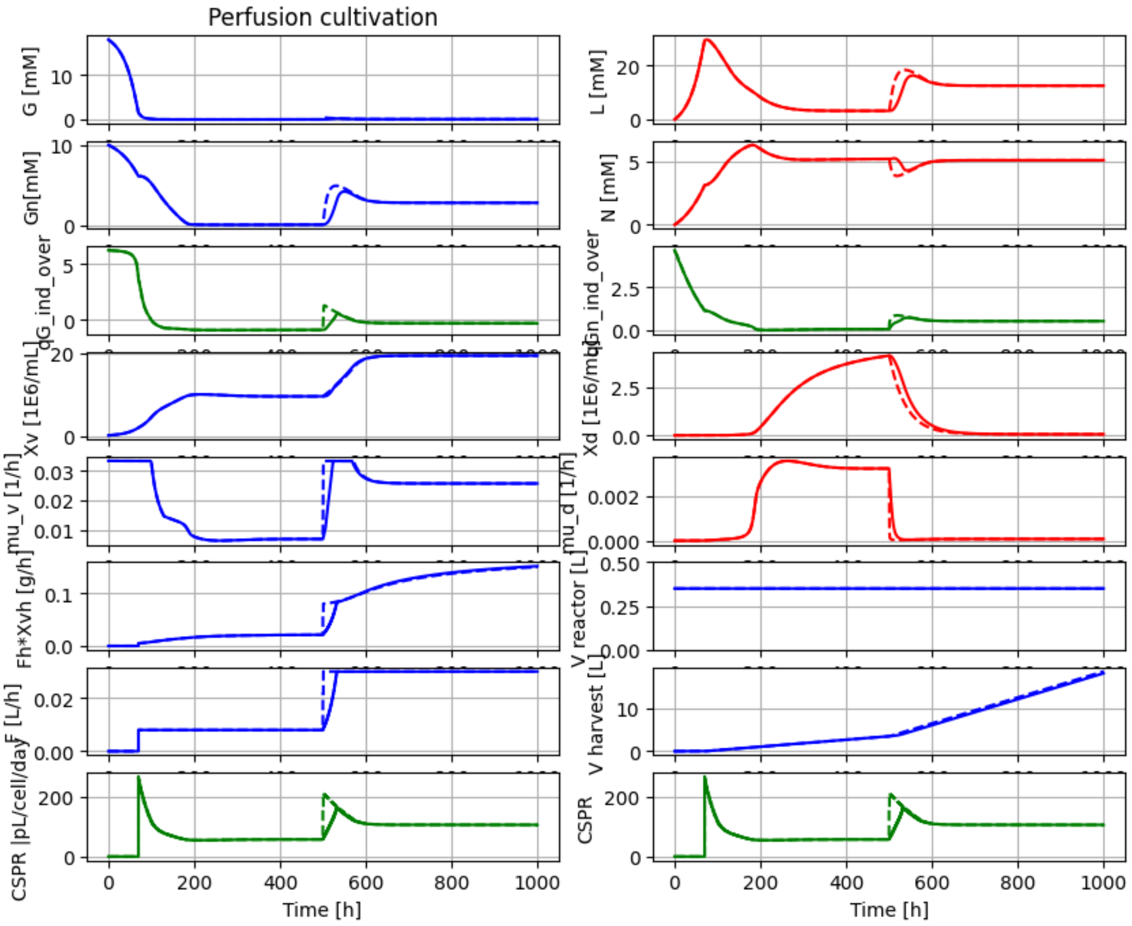
```
model.get('CSPR')
```

```
array([105.41560997])
```

We see here the shorter settling time for cell conc Xv when the slower exponential increase of perfusion rate is used, compared to an abrupt change.

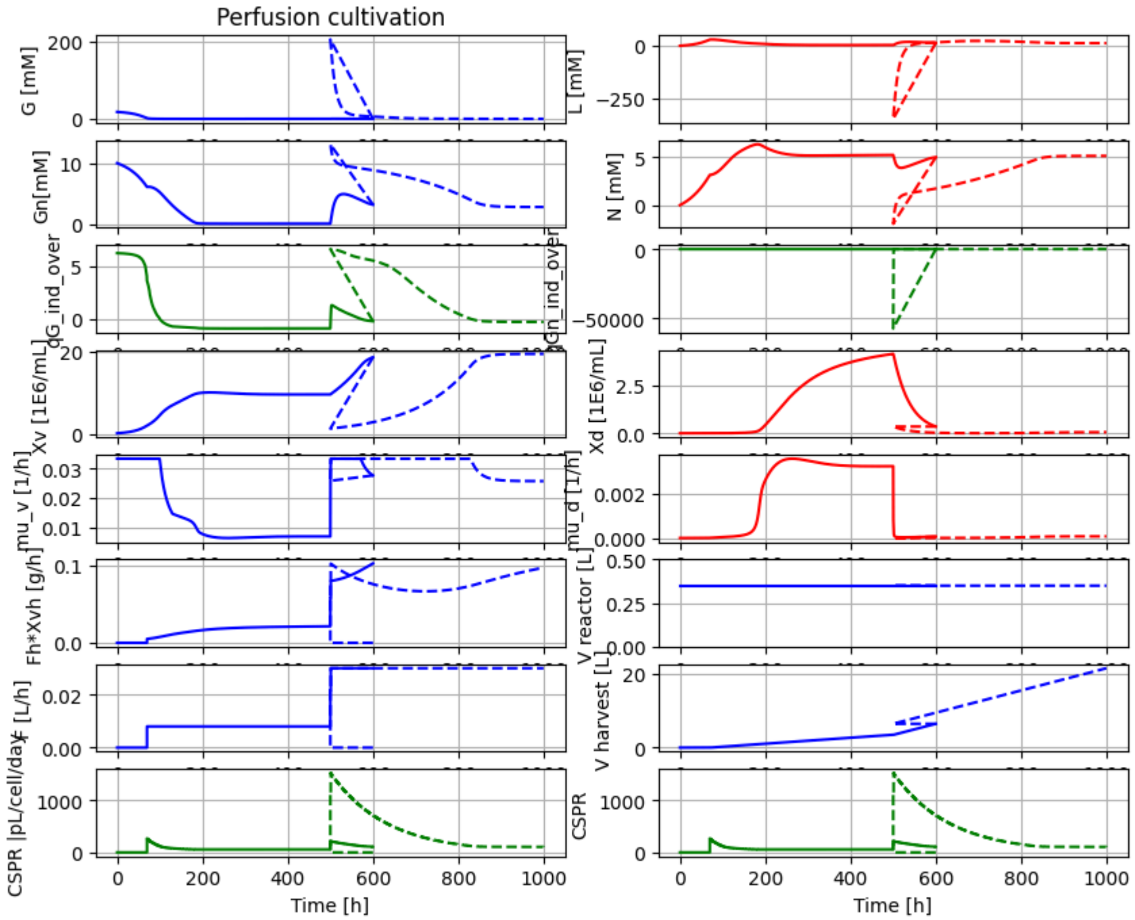
```
newplot('Perfusion cultivation', plotType='Extended')

par(samplePeriod=1); par(F1=0.0080)                 # General parameters
par(mu_ref=0.04); simu()                            # First simulation (solid)
par(mu_ref=1); simu()                               # Second simulation (dashed)
```



We see that a somewhat smaller change in perfusion rate make the differnce in settling time of Xv for the two strategies to almost disappear.

```
newplot('Perfusion cultivation', plotType='Extended')
simu(600)
simu(400,'cont')
```



```
describe('feedtank.W')
```

```
Reactor broth weight : 78.915 [ kg ]
```

```
describe('feedtank.V')
```

```
Feed volume : 78.577 [ L ]
```

```
78.915/78.577
```

```
1.0043015131654303
```

```
describe('bioreactor.broth_decay.k_decay')
```

```
Rate of decay of viable cells : 0.0 [ 1E9 cells/(L*h) ]
```

```
describe('k_lysis')
```

```
Specific rate of lysis of dead cells : 0.0 [ 1/h ]
```

```
disp('culture')
```

```
qG_max1 : 0.297
qG_max2 : 0.038
qGn_max1 : 0.124
qGn_max2 : 0.022
mu_d_max : 0.13
```

```
describe('mu')
```

```
Specific cell growth rate variable : 0.026 [ 1/h ]
```

```
describe('parts')
```

```
['bioreactor', 'bioreactor.broth_decay', 'bioreactor.culture', 'CSPR', 'cspr_openloop', '
```