

BPL_TEST2_Batch_design_space script with PyFMI ver 2.7.4

The key library PyFMI v2.7.4 is installed and downgrading is done Numpy v1.19.1. To simplify this we first install conda.

After the installation a small application BPL_TEST2_Batch_design_space 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 18.04.6 LTS
Release:        18.04
Codename:       bionic
```

```
%env PYTHONPATH=
```

```
env: PYTHONPATH=
```

```
!wget https://repo.anaconda.com/miniconda/Miniconda3-py37_4.12.0-Linux-x86_64.sh
!chmod +x Miniconda3-py37_4.12.0-Linux-x86_64.sh
!bash ./Miniconda3-py37_4.12.0-Linux-x86_64.sh -b -f -p /usr/local
import sys
sys.path.append('/usr/local/lib/python3.7/site-packages/')
- ruamel-yaml==0.15.100=py37h27cfd23_1003
- setuptools==61.2.0=py37h06a4308_0
- six==1.16.0=pyhd3eb1b0_1
- sqlite==3.38.2=hc218d9a_0
- tk==8.6.11=h1ccaba5_0
- tqdm==4.63.0=pyhd3eb1b0_0
- urllib3==1.26.8=pyhd3eb1b0_0
- wheel==0.37.1=pyhd3eb1b0_0
- xz==5.2.5=h7b6447c_0
- yaml==0.2.5=h7b6447c_0
- zlib==1.2.12=h7f8727e_1
```

The following NEW packages will be INSTALLED:

_libgcc_mutex	pkgs/main/linux-64::_libgcc_mutex-0.1-main
_openmp_mutex	pkgs/main/linux-64::_openmp_mutex-4.5-1_gnu
brotlipy	pkgs/main/linux-64::brotlipy-0.7.0-py37h27cfd23_1003
ca-certificates	pkgs/main/linux-64::ca-certificates-2022.3.29-h06a4308_1
certifi	pkgs/main/linux-64::certifi-2021.10.8-py37h06a4308_2
cffi	pkgs/main/linux-64::cffi-1.15.0-py37hd667e15_1
charset-normalizer	pkgs/main/noarch::charset-normalizer-2.0.4-pyhd3eb1b0_0
colorama	pkgs/main/noarch::colorama-0.4.4-pyhd3eb1b0_0
conda	pkgs/main/linux-64::conda-4.12.0-py37h06a4308_0
conda-content-tru~	pkgs/main/noarch::conda-content-trust-0.1.1-pyhd3eb1b0_0
conda-package-hand	pkgs/main/linux-64::conda-package-handling-1.0.1-py37h7766

```

conda-package-nan~ pkgs/main/linux-64::conda-package-nan-1.8.1-py37h9ce1e76_0
cryptography       pkgs/main/linux-64::cryptography-36.0.0-py37h9ce1e76_0
idna                pkgs/main/noarch::idna-3.3-pyhd3eb1b0_0
ld_impl_linux-64   pkgs/main/linux-64::ld_impl_linux-64-2.35.1-h7274673_9
libffi              pkgs/main/linux-64::libffi-3.3-he6710b0_2

```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link.](#)

```

ncurses            pkgs/main/linux-64::ncurses-6.3-h7f8727e_2
openssl            pkgs/main/linux-64::openssl-1.1.1n-h7f8727e_0

pip                pkgs/main/linux-64::pip-21.2.2-py37h06a4308_0
pycosat            pkgs/main/linux-64::pycosat-0.6.3-py37h27cfd23_0
pycparser          pkgs/main/noarch::pycparser-2.21-pyhd3eb1b0_0
pyopenssl          pkgs/main/noarch::pyopenssl-22.0.0-pyhd3eb1b0_0
pysocks            pkgs/main/linux-64::pysocks-1.7.1-py37_1
python             pkgs/main/linux-64::python-3.7.13-h12debd9_0
readline           pkgs/main/linux-64::readline-8.1.2-h7f8727e_1
requests           pkgs/main/noarch::requests-2.27.1-pyhd3eb1b0_0
ruamel_yaml        pkgs/main/linux-64::ruamel_yaml-0.15.100-py37h27cfd23_0
setuptools         pkgs/main/linux-64::setuptools-61.2.0-py37h06a4308_0
six                pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_1
sqlite             pkgs/main/linux-64::sqlite-3.38.2-hc218d9a_0
tk                 pkgs/main/linux-64::tk-8.6.11-h1ccaba5_0
tqdm               pkgs/main/noarch::tqdm-4.63.0-pyhd3eb1b0_0
urllib3            pkgs/main/noarch::urllib3-1.26.8-pyhd3eb1b0_0
wheel              pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
xz                 pkgs/main/linux-64::xz-5.2.5-h7b6447c_0
yaml               pkgs/main/linux-64::yaml-0.2.5-h7b6447c_0
zlib               pkgs/main/linux-64::zlib-1.2.12-h7f8727e_1

```

Preparing transaction: done

Executing transaction: done

installation finished.

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

```

colorama-0.4.4-pyhd3eb1b0_0
conda-content-trust-0.1.1-pyhd3eb1b0_0
six-1.16.0-pyhd3eb1b0_1

```

The following packages will be UPDATED:

```

_openmp_mutex      4.5-1_gnu --> 5.1-1_gnu
ca-certificates    2022.3.29-h06a4308_1 --> 2022.07.19-h06
certifi            2021.10.8-py37h06a4308_2 --> 2022.6.15-py37
cffi               1.15.0-py37hd667e15_1 --> 1.15.1-py37h74
conda              4.12.0-py37h06a4308_0 --> 4.14.0-py37h06
cryptography       36.0.0-py37h9ce1e76_0 --> 37.0.1-py37h9c
ld_impl_linux-64   2.35.1-h7274673_9 --> 2.38-h1181459_
libgcc-ng          9.3.0-h5101ec6_17 --> 11.2.0-h123456
libgomp            9.3.0-h5101ec6_17 --> 11.2.0-h123456
libstdcxx-ng       9.3.0-hd4cf53a_17 --> 11.2.0-h123456
ncurses            6.3-h7f8727e_2 --> 6.3-h5eee18b_3
openssl            1.1.1n-h7f8727e_0 --> 1.1.1q-h7f8727
pip                21.2.2-py37h06a4308_0 --> 22.1.2-py37h06
requests           pkgs/main/noarch::requests-2.27.1-pyh~ --> pkgs/main/linv
setuptools         61.2.0-py37h06a4308_0 --> 63.4.1-py37h06
sqlite             3.38.2-hc218d9a_0 --> 3.39.2-h508229

```

```

tk               8.6.11-niccaba5_0 --> 8.6.12-niccaba5
tqdm             pkgs/main/noarch::tqdm-4.63.0-pyhd3eb~ --> pkgs/main/linux64::tqdm-4.63.0-pyhd3eb~
urllib3         pkgs/main/noarch::urllib3-1.26.8-pyhd~ --> pkgs/main/linux64::urllib3-1.26.8-pyhd~
xz              5.2.5-h7b6447c_0 --> 5.2.5-h7f8727e
zlib            1.2.12-h7f8727e_1 --> 1.2.12-h5eee18

```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link](#).

Downloading and Extracting Packages

```

ncurses-6.3      | 781 KB      | : 100% 1.0/1 [00:00<00:00, 3.25it/s]
pip-22.1.2       | 2.4 MB      | : 100% 1.0/1 [00:00<00:00, 3.98it/s]
toolz-0.11.2     | 49 KB       | : 100% 1.0/1 [00:00<00:00, 16.15it/s]
requests-2.28.1  | 92 KB       | : 100% 1.0/1 [00:00<00:00, 12.37it/s]

cytoolz-0.11.0   | 328 KB      | : 100% 1.0/1 [00:00<00:00, 14.14it/s]
urllib3-1.26.11  | 181 KB      | : 100% 1.0/1 [00:00<00:00, 15.93it/s]
libstdcxx-ng-11.2.0 | 4.7 MB     | : 100% 1.0/1 [00:00<00:00, 5.28it/s]
openssl-1.1.1q   | 2.5 MB      | : 100% 1.0/1 [00:00<00:00, 6.20it/s]
conda-4.14.0     | 909 KB      | : 100% 1.0/1 [00:00<00:00, 7.97it/s]
certifi-2022.6.15 | 153 KB      | : 100% 1.0/1 [00:00<00:00, 17.21it/s]
setuptools-63.4.1 | 1.1 MB      | : 100% 1.0/1 [00:00<00:00, 7.35it/s]
tqdm-4.64.0      | 126 KB      | : 100% 1.0/1 [00:00<00:00, 13.99it/s]
_openmp_mutex-5.1 | 21 KB       | : 100% 1.0/1 [00:00<00:00, 19.07it/s]
libgcc-ng-11.2.0 | 5.3 MB      | : 100% 1.0/1 [00:00<00:00, 4.95it/s]
tk-8.6.12        | 3.0 MB      | : 100% 1.0/1 [00:00<00:00, 4.90it/s]
ld_impl_linux-64-2.3 | 654 KB     | : 100% 1.0/1 [00:00<00:00, 13.01it/s]
ca-certificates-2022 | 124 KB     | : 100% 1.0/1 [00:00<00:00, 18.54it/s]
xz-5.2.5         | 339 KB      | : 100% 1.0/1 [00:00<00:00, 13.48it/s]
zlib-1.2.12      | 103 KB      | : 100% 1.0/1 [00:00<00:00, 17.99it/s]
sqlite-3.39.2    | 1.1 MB      | : 100% 1.0/1 [00:00<00:00, 12.22it/s]
cffi-1.15.1      | 227 KB      | : 100% 1.0/1 [00:00<00:00, 14.80it/s]
libgomp-11.2.0   | 474 KB      | : 100% 1.0/1 [00:00<00:00, 13.99it/s]
cryptography-37.0.1 | 1.3 MB     | : 100% 1.0/1 [00:00<00:00, 6.69it/s]
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

```

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

```
conda 4.14.0
Python 3.7.13
```

```
!conda install -c conda-forge pyfmi==2.7.4 --yes # Install the key package
```

```

liblapack      conda-forge/linux-64::liblapack-3.9.0-15_linux64_openblas
libopenblas    conda-forge/linux-64::libopenblas-0.3.20-pthreads_h78a641
libxml2        conda-forge/linux-64::libxml2-2.9.12-h72842e0_0
libxslt        conda-forge/linux-64::libxslt-1.1.33-h15afd5d_2
lxml           conda-forge/linux-64::lxml-4.8.0-py37h540881e_2
metis         conda-forge/linux-64::metis-5.1.0-h58526e2_1006
mpfr          conda-forge/linux-64::mpfr-4.1.0-h9202a9a_1
numpy         conda-forge/linux-64::numpy-1.21.6-py37h976b520_0
pyfmi         conda-forge/linux-64::pyfmi-2.7.4-py37h161383b_0
python_abi    conda-forge/linux-64::python_abi-3.7-2_cp37m
scipy         conda-forge/linux-64::scipy-1.7.3-py37hf2a6cfl_0
suitesparse   conda-forge/linux-64::suitesparse-5.10.1-h9e50725_1
sundials      conda-forge/linux-64::sundials-5.8.0-h558c624_0
tbb           conda-forge/linux-64::tbb-2021.5.0-h924138e_1

```

The following packages will be UPDATED:

The following packages will be UPDATED:

certifi pkgs/main/linux-64::certifi-2022.6.15~ --> conda-forge/nc

The following packages will be SUPERSEDED by a higher priority channel:

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link.](#)

conda pkgs/main::conda-4.14.0-py37h06a4308_0 --> conda-forge::c

openssl pkgs/main::openssl-1.1.1q-h7f8727e_0 --> conda-forge::c

Downloading and Extracting Packages

icu-68.2	13.1 MB	: 100% 1.0/1 [00:02<00:00, 2.52s/it]
pyfmi-2.7.4	12.4 MB	: 100% 1.0/1 [00:04<00:00, 4.39s/it]
libblas-3.9.0	12 KB	: 100% 1.0/1 [00:00<00:00, 28.51it/s]
python_abi-3.7	4 KB	: 100% 1.0/1 [00:00<00:00, 28.65it/s]
metis-5.1.0	4.1 MB	: 100% 1.0/1 [00:00<00:00, 1.14it/s]
libxslt-1.1.33	522 KB	: 100% 1.0/1 [00:00<00:00, 4.93it/s]
lxml-4.8.0	1.4 MB	: 100% 1.0/1 [00:00<00:00, 2.90it/s]
mpfr-4.1.0	2.6 MB	: 100% 1.0/1 [00:00<00:00, 1.31it/s]
conda-4.14.0	1010 KB	: 100% 1.0/1 [00:00<00:00, 2.60it/s]
fmilib-2.2.3	532 KB	: 100% 1.0/1 [00:00<00:00, 1.42it/s]
openssl-1.1.1o	2.1 MB	: 100% 1.0/1 [00:00<00:00, 2.40it/s]
suitesparse-5.10.1	2.4 MB	: 100% 1.0/1 [00:00<00:00, 1.23it/s]
numpy-1.21.6	6.1 MB	: 100% 1.0/1 [00:01<00:00, 1.65s/it]
assimulo-3.2.9	2.6 MB	: 100% 1.0/1 [00:01<00:00, 1.28s/it]
sundials-5.8.0	1.0 MB	: 100% 1.0/1 [00:00<00:00, 1.27it/s]
tbb-2021.5.0	1.9 MB	: 100% 1.0/1 [00:00<00:00, 2.84it/s]
liblapack-3.9.0	12 KB	: 100% 1.0/1 [00:00<00:00, 26.21it/s]
ca-certificates-2022	149 KB	: 100% 1.0/1 [00:00<00:00, 19.20it/s]
libopenblas-0.3.20	10.1 MB	: 100% 1.0/1 [00:01<00:00, 1.99s/it]
scipy-1.7.3	21.8 MB	: 100% 1.0/1 [00:04<00:00, 4.70s/it]
libgfortran5-12.1.0	1.8 MB	: 100% 1.0/1 [00:00<00:00, 2.47it/s]
libgfortran-ng-12.1.	23 KB	: 100% 1.0/1 [00:00<00:00, 25.95it/s]
gmp-6.2.1	806 KB	: 100% 1.0/1 [00:00<00:00, 4.53it/s]
libiconv-1.16	1.4 MB	: 100% 1.0/1 [00:00<00:00, 4.14it/s]
certifi-2022.6.15	154 KB	: 100% 1.0/1 [00:00<00:00, 16.41it/s]
libcblas-3.9.0	12 KB	: 100% 1.0/1 [00:00<00:00, 26.58it/s]
libxml2-2.9.12	772 KB	: 100% 1.0/1 [00:00<00:00, 5.67it/s]

Preparing transaction: done

Verifying transaction: done

Executing transaction: done

!conda install numpy=1.19.1 --yes # Need to downgrade numpy

Collecting package metadata (current_repodata.json): done

Solving environment: failed with initial frozen solve. Retrying with flexible

Collecting package metadata (repodata.json): done

Solving environment: done

Package Plan

environment location: /usr/local

added / updated specs:

- numpy=1.19.1

The following packages will be downloaded:

package	build
blas-1.0	openblas_0 46 KB
numpy-base-1.19.1	py37h75fe3a5_0 4.1 MB
numpy-1.19.1	py37h976b52_0 21 KB

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link](#).

Total: 4.2 MB

The following NEW packages will be INSTALLED:

```
blas          pkgs/main/linux-64::blas-1.0-openblas
numpy-base    pkgs/main/linux-64::numpy-base-1.19.1-py37h75fe3a5_0
```

The following packages will be UPDATED:

```
ca-certificates  conda-forge::ca-certificates-2022.6.1~ --> pkgs/main::ca-
openssl          conda-forge::openssl-1.1.1o-h166bdaf_0 --> pkgs/main::ope
```

The following packages will be SUPERSEDED by a higher-priority channel:

```
certifi          conda-forge/noarch::certifi-2022.6.15~ --> pkgs/main/linux-64::certifi
conda            conda-forge::conda-4.14.0-py37h89c186~ --> pkgs/main::conda
numpy            conda-forge::numpy-1.21.6-py37h976b52~ --> pkgs/main::numpy
```

Downloading and Extracting Packages

```
blas-1.0          | 46 KB      | : 100% 1.0/1 [00:00<00:00, 10.68it/s]
numpy-base-1.19.1 | 4.1 MB     | : 100% 1.0/1 [00:01<00:00, 1.84s/it]
numpy-1.19.1      | 21 KB      | : 100% 1.0/1 [00:00<00:00, 3.10it/s]
```

Preparing transaction: done

Verifying transaction: done

Executing transaction: done

Retrieving notices: ...working... done

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

- FMU - BPL_TEST2_Batch_design_space_no_noise_linux_jm_cs.fmu
- Setup-file - BPL_TEST2_Batch_design_space_no_noise_explore.py
- FMU - BPL_TEST2_Batch_design_space_with_noise_linux_jm_cs.fmu
- Setup-file - BPL_TEST2_Batch_design_space_with_noise_explore.py

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

```
%%bash
```

```
git clone https://github.com/janpeter19/BPL_TEST2_Batch_design_space
```

Cloning into 'BPL_TEST2_Batch_design_space'...

```
%cd BPL_TEST2_Batch_design_space

/content/BPL_TEST2_Batch_design_space
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link](#).

In this notebook the design space for a batch cultivation process is determined and visualized. The example is kept as simple as possible. The culture grow on a substrate S and the cell concentration X increase until the substrate is consumed. We study the problem first without any measurement noise and then later with measurement noise and use one separate FMU for each. The end criteria for a batch is here when the substrate level has decreased below a certain predefined level and that time is called `time_final`:

- $S < S_{min}$

The evaluation of the batch culture is just in terms of the obtained value of cell concentration at the end in combination with how long time the culture took. The batch is accepted provided the culture fulfill the two requirements:

- $X_{final} > X_{final_min}$
- $Time_{final} < time_{final_max}$

The question is what range of process parameters Y and qS_{max} that can be allowed to still get accepted batches.

Here we simply use brute force and sweep through a number combinations of process parameters and evaluate by simulation the result for each parameter setting. We get rather clear-cut corners in the process parameter space that result in acceptable batches.

In the later part we introduce substrate measurement error and in this way introduce some uncertainty in the determination of end of batch. The impact of this measurement noise is that the design space get more rounded corners.

The practical experimental approach is usually to just use a few parameter combinations and evaluate these and from that information calculate the design space. Usually "process linearity" assumption is used. The combination of this experimental approach with brute force simulation is discussed in reference [1].

▼ 1 Batch end detection - no measurement noise

Here we load a system model without noise. Thus detection of end of batch is an event in continuous time.

```
run -i BPL_TEST2_Batch_no_noise_explore.py

Linux - run FMU pre-compiled JModelica 2.4
```

Model for bioreactor has been setup. Key commands:

- `par()` - change of parameters and initial values
- `init()` - change initial values only
- `simu()` - simulate and plot

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link.](#)

- `disp()` - display parameters and initial values from the last simulation
- `describe()` - describe culture, broth, parameters, variables with values /

Note that both `disp()` and `describe()` takes values from the last simulation

Brief information about a command by `help()`, eg `help(simu)`

Key system information is listed with the command `system_info()`

```
# Adjust the diagram size
%matplotlib inline
plt.rcParams['figure.figsize'] = [30/2.54, 24/2.54]
```

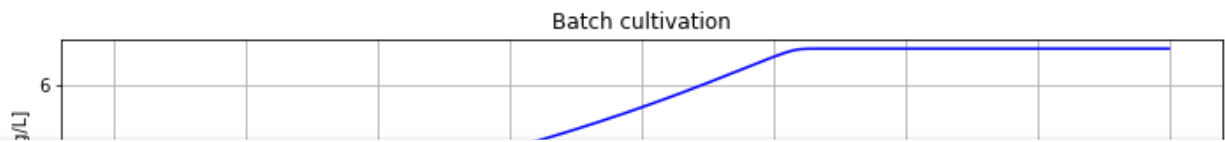
▼ 1.1 Batch evaluation

The first an example of batch that has an end of batch that fulfills the criteria for acceptance. In the following diagram we see examples of impact of variation on the criteria for acceptance.

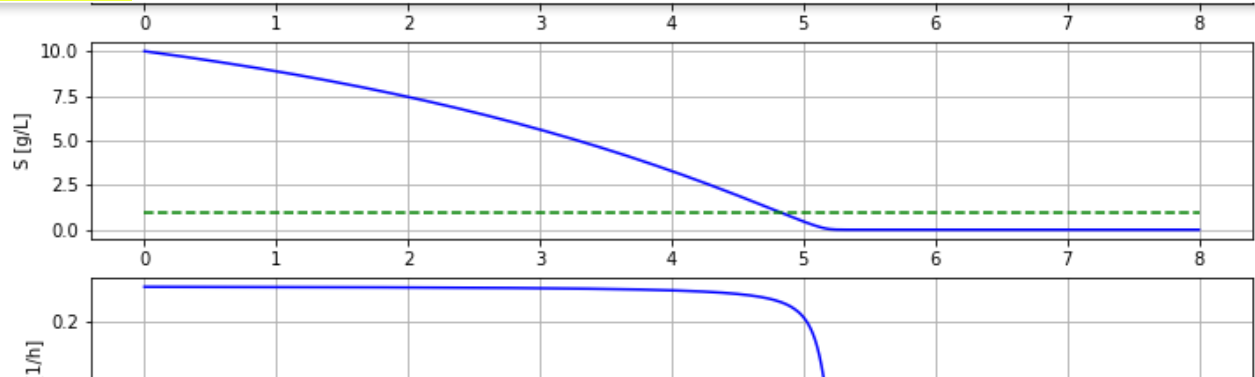
The variable `batch_evaluation` goes from 0 to either 1 or a negative value when end of batch is detected. A positive value 1 means that the acceptance criteria is fulfilled and a negative value -1, -2 or -3 is obtained if one or more criteria for acceptance is not fulfilled.

```
# Nominal parameters
par(S_min=1.0, time_final_max=6.0, X_final_min=5.0)
init(VX_0=2, VS_0=10)
par(Y=0.5, qSmax=0.5, Ks=0.1)

# Simulation of nominal parameters that gives a batch that meed the end criteria
newplot(plotType='TimeSeries_2')
simu(8)
```



Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link](#).



Example of process parameter changes and how they meet the end criteria

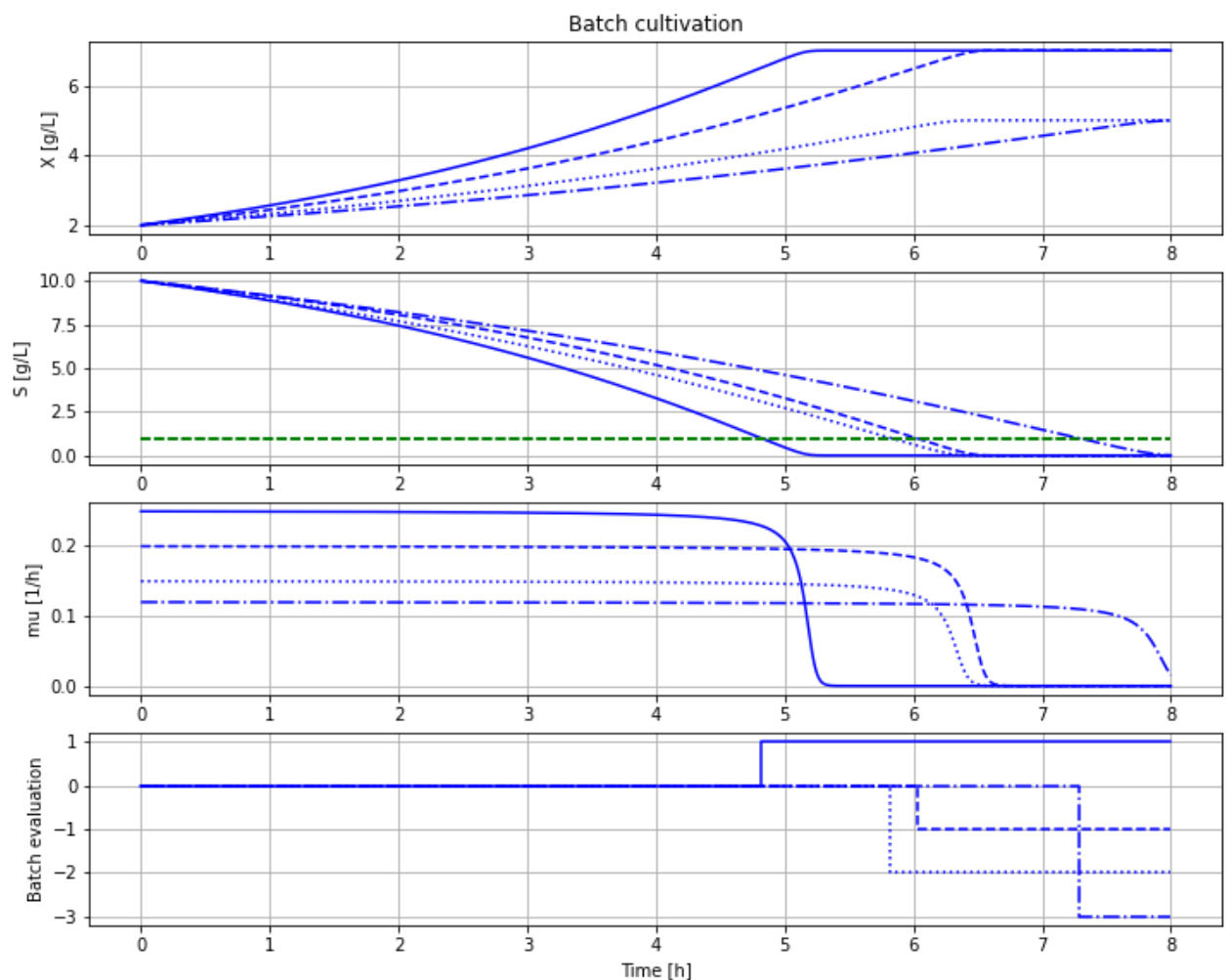
```
newplot(plotType='TimeSeries_2')
```

```
par(Y=0.50, qSmax=0.50); simu(8) # - pass (solid line)
```

```
par(Y=0.50, qSmax=0.40); simu(8) # - fail criteria time_final < 6.0 (dashed line)
```

```
par(Y=0.30, qSmax=0.50); simu(8) # - fail criteria X_final > 5.0 (dotted line)
```

```
par(Y=0.30, qSmax=0.40); simu(8) # - fail both criteria (dash dotted line)
```



We see that the accepted batch (solid line) finish first. The batches that fail take longer time and two of them has also lower cell concentration at the end.

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link.](#)

Now let us systematically sweep through a number of combinations of process parameters Y and q_{Smax} and evaluate the batches and visualise the result.

```
# Define sweep ranges and storage of final data
nY = 20
nqSmax = 20
Y_range = np.linspace(0.3,0.5,nY)
qSmax_range = np.linspace(0.4,0.6,nqSmax)
data = np.zeros([nY,nqSmax,5])

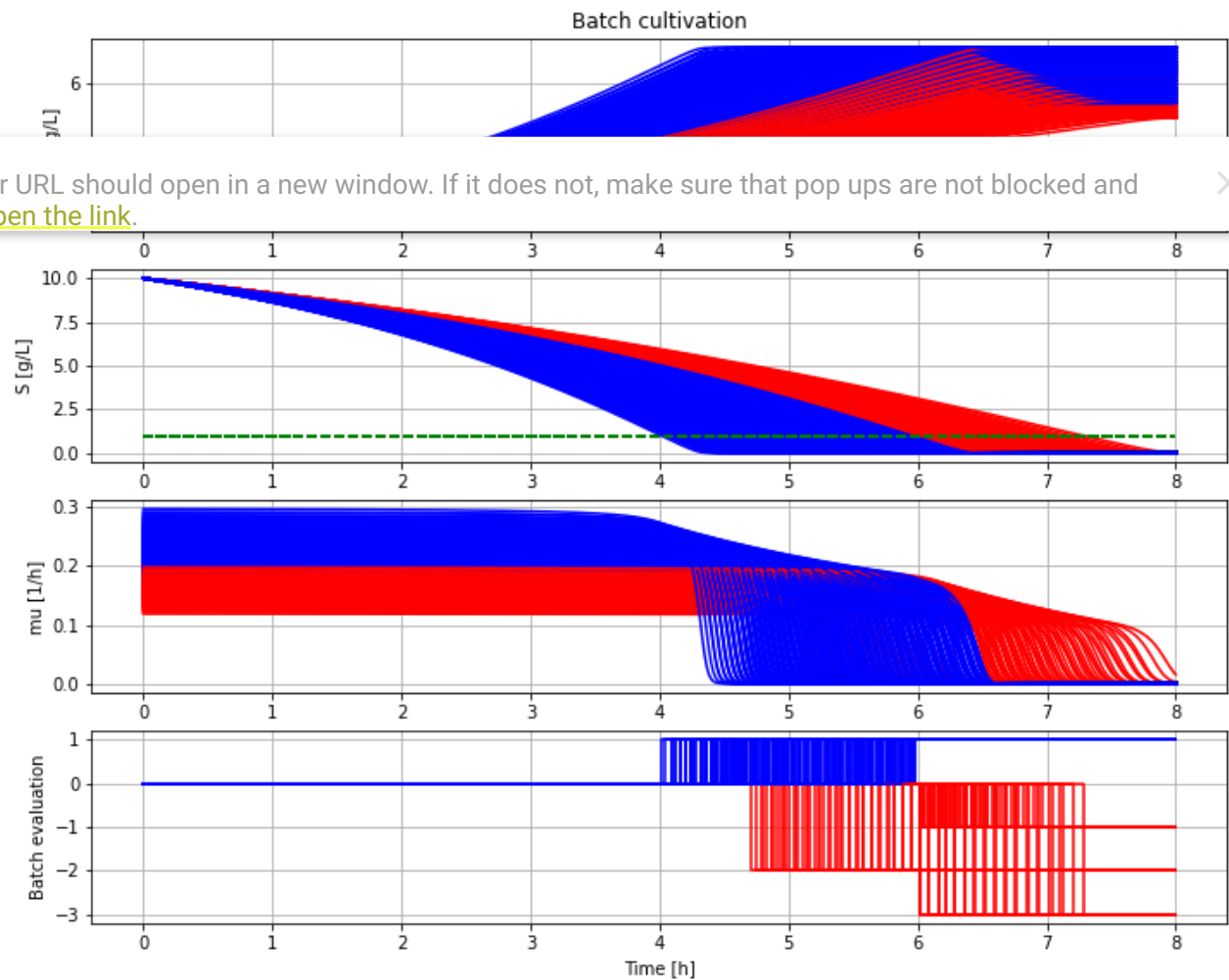
# Run parameter sweep - takes a few minutes
newplot(plotType='TimeSeries_2_diagrams')
init(VX_0=2, VS_0=10)

for j in range(nY):
    for k in range(nqSmax):
        par(Y=Y_range[j])
        par(qSmax=qSmax_range[k])
        simu(8)

# Store final results
data[j,k,0] = Y_range[j]
data[j,k,1] = qSmax_range[k]
data[j,k,2] = sim_res['monitor.time_final'][-1]
data[j,k,3] = sim_res['monitor.X_final'][-1]
data[j,k,4] = sim_res['monitor.batch_evaluation'][-1]

# Plot simulation results
if sim_res['monitor.batch_evaluation'][-1] > 0:
    ax1.plot(sim_res['time'], sim_res['bioreactor.c[1]'], 'b-')
    ax2.plot(sim_res['time'], sim_res['bioreactor.c[2]'], 'b-')
    ax2.plot([0, simulationTime], [model.get('monitor.S_min'), model.get('m
    ax3.plot(sim_res['time'], sim_res['bioreactor.culture.q[1]'], 'b-')
    ax4.step(sim_res['time'], sim_res['monitor.batch_evaluation'], where='pos
else:
    ax1.plot(sim_res['time'], sim_res['bioreactor.c[1]'], 'r-')
    ax2.plot(sim_res['time'], sim_res['bioreactor.c[2]'], 'r-')
    ax2.plot([0, simulationTime], [model.get('monitor.S_min'), model.get('m
    ax3.plot(sim_res['time'], sim_res['bioreactor.culture.q[1]'], 'r-')
    ax4.step(sim_res['time'], sim_res['monitor.batch_evaluation'], where='pos

plt.show()
```



Batches represented by blue lines are those that in the end got accepted. The red ones failed.

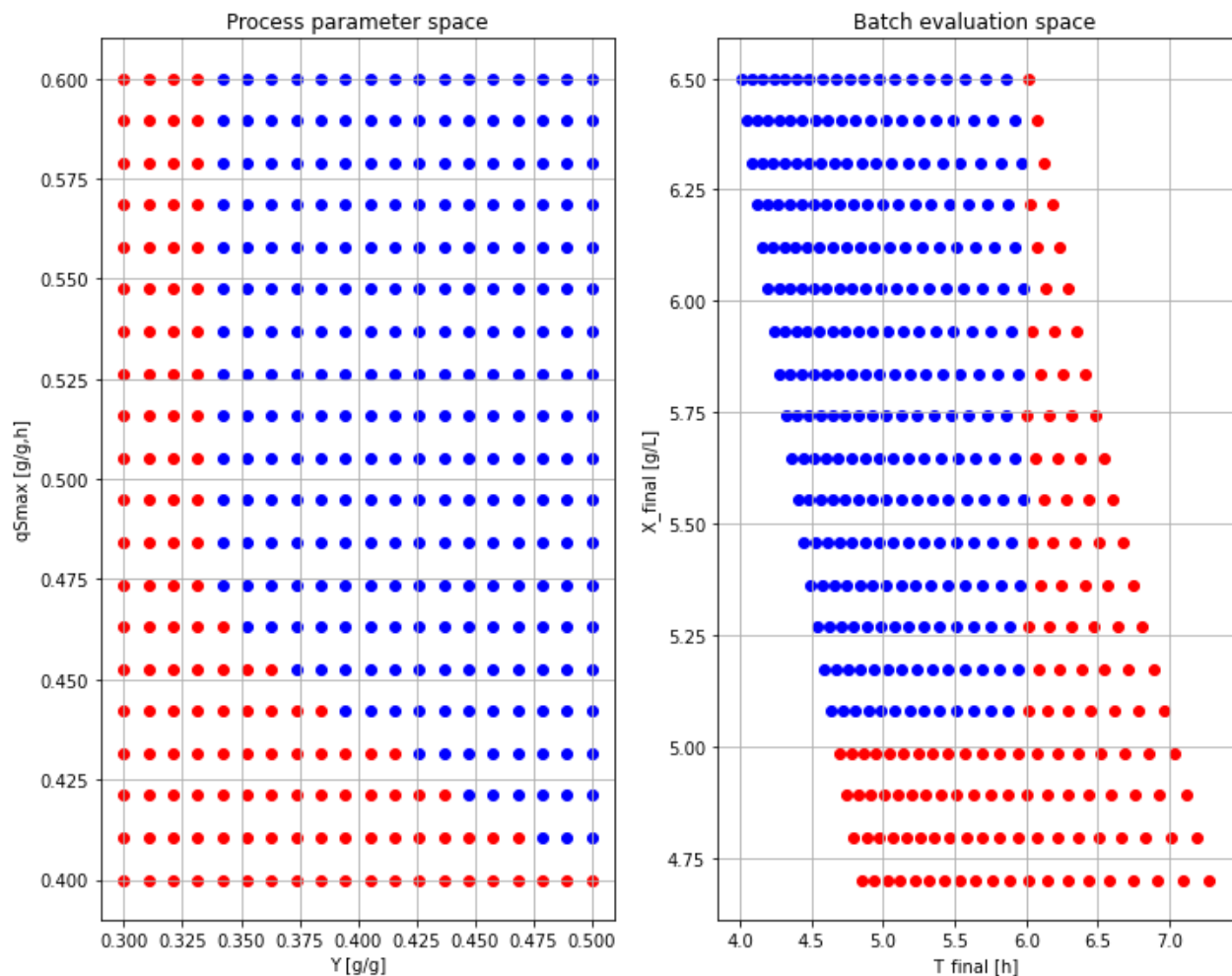
```
# Show end results
plt.figure()
ax1 = plt.subplot(1,2,1)
ax2 = plt.subplot(1,2,2)

for j in range(nY):
    for k in range(nqSmax):
        if data[j,k,4] > 0:
            ax1.scatter(data[j,k,0],data[j,k,1],c='b')
        else:
            ax1.scatter(data[j,k,0],data[j,k,1],c='r')
ax1.grid()
#plt.axis([0, 0.8, 0, 0.8])
ax1.set_ylabel('qSmax [g/g,h]')
ax1.set_xlabel('Y [g/g]')
ax1.set_title('Process parameter space')

for j in range(nY):
    for k in range(nqSmax):
        if data[j,k,4] > 0:
            ax2.scatter(data[j,k,2],data[j,k,3],c='b')
        else:
            ax2.scatter(data[j,k,2],data[j,k,3],c='r')
```

```
ax2.scatter(data[j,k,2],data[j,k,3],c='r')
ax2.grid()
#plt.axis([0, 8, 0, 8])
ax2.set_xlabel('T_final [h]')
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link](#).



Here we visualize the previous simulations results in a different way with focus on the end result. Each dot in the left diagram (process parameter space) represents a simulation that gives a result in the right diagram (batch evaluation space). The blue dots are those batches that were accepted and the red ones those that failed.

The blue dots in the process parameter space show the "design space" for the acceptance criteria we have.

▼ 2 Batch end detection - with measurement noise

Here we load a system model with normal noise added to the sampled value of substrate concentration. The measurement of substrate concentration usually has a higher variation than measurement of cell concentration and therefore we focus here on the impact on substrate

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link.](#)

This discretization also introduces an error in detection of the end point. By changing this sample interval to shorter values you can see the impact of this error but not done here

```
run -i BPL_TEST2_Batch_with_noise_explore.py
```

Linux - run FMU pre-compiled JModelica 2.4

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

Note that both disp() and describe() takes values from the last simulation

Brief information about a command by help(), eg help(simu)

Key system information is listed with the command system_info()

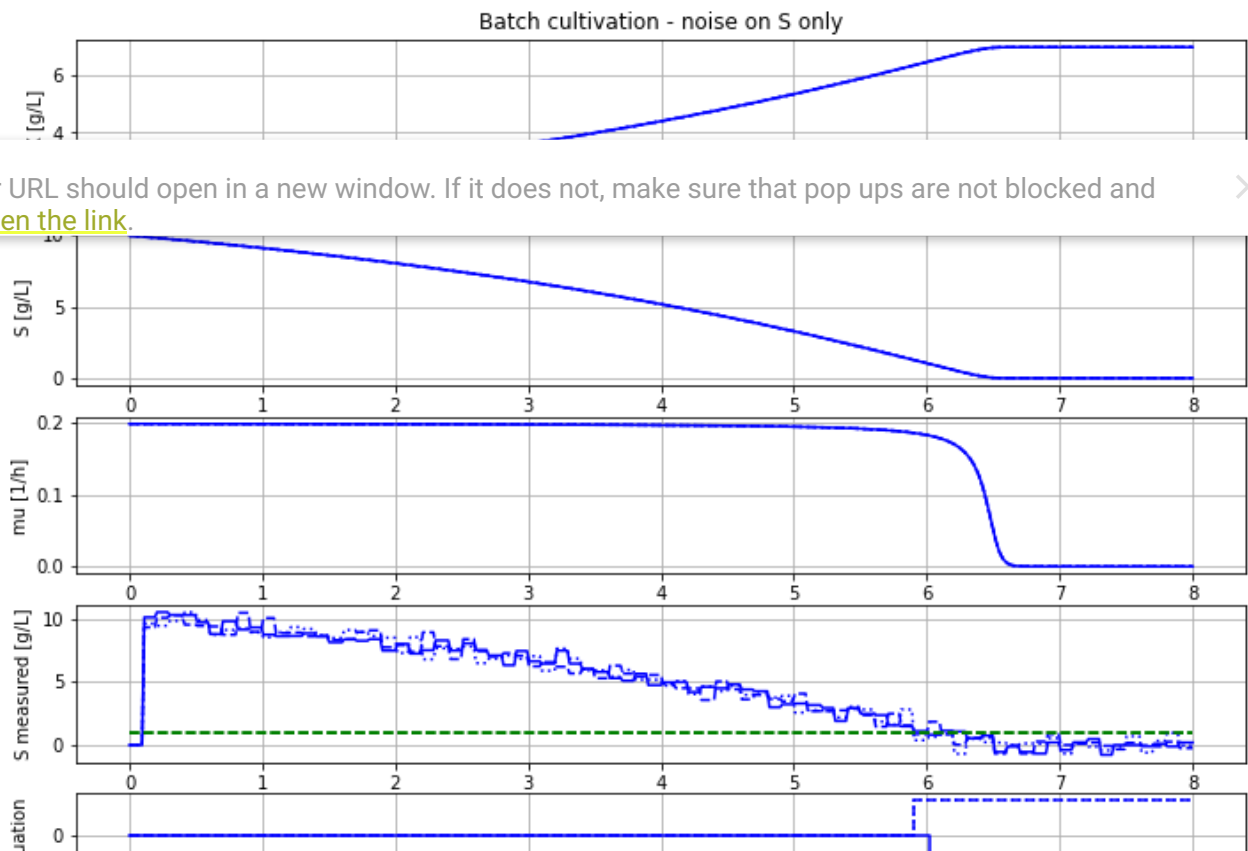
<Figure size 850.394x680.315 with 0 Axes>

▼ 2.1 Batch evaluation under substrate measurement error

Here we see an example of how substrate measurement noise directly affects the evaluation of the batch from acceptable to not acceptable.

```
# Nominal parameters
par(S_min=1.0, time_final_max=6.0, X_final_min=5.0)
init(VX_0=2, VS_0=10)
par(Y=0.5, qSmax=0.5, Ks=0.1)
par(sigma=0.48, samplePeriod=0.1)

# Simulation of nominal parameters that gives a batch that meets the end criteria
newplot(plotType='TimeSeries_2')
par(Y=0.5, qSmax=0.4);
for value in [2,3,5]: par(seed=value); simu(8)
```



2.2 Batch evaluation under process variation and measurement error - parameter sweep

Now let us again systematically sweep through a number of combinations of process parameters Y and qS_{max} and evaluate the batches and visualise their result.

```
# Define sweep ranges and storage of final data
nY = 20
nqSmax = 20
Y_range = np.linspace(0.3,0.5,nY)
qSmax_range = np.linspace(0.4,0.6,nqSmax)
data = np.zeros([nY,nqSmax,5])

# Run parameter sweep - takes a few minutes
newplot(plotType='TimeSeries_2_diagrams')
par(sigma=0.48, seed=1, samplePeriod=0.1)

for j in range(nY):
    for k in range(nqSmax):
        par(Y=Y_range[j])
        par(qSmax=qSmax_range[k])
        simu(8)

# Store final results
data[j,k,0] = Y_range[j]
data[j,k,1] = qSmax_range[k]
data[j,k,2] = sim_res['monitor.time_final'][-1]
data[j,k,3] = sim_res['monitor.X_final'][-1]
```

```
data[j,k,4] = sim_res['monitor.batch_evaluation'][-1]
```

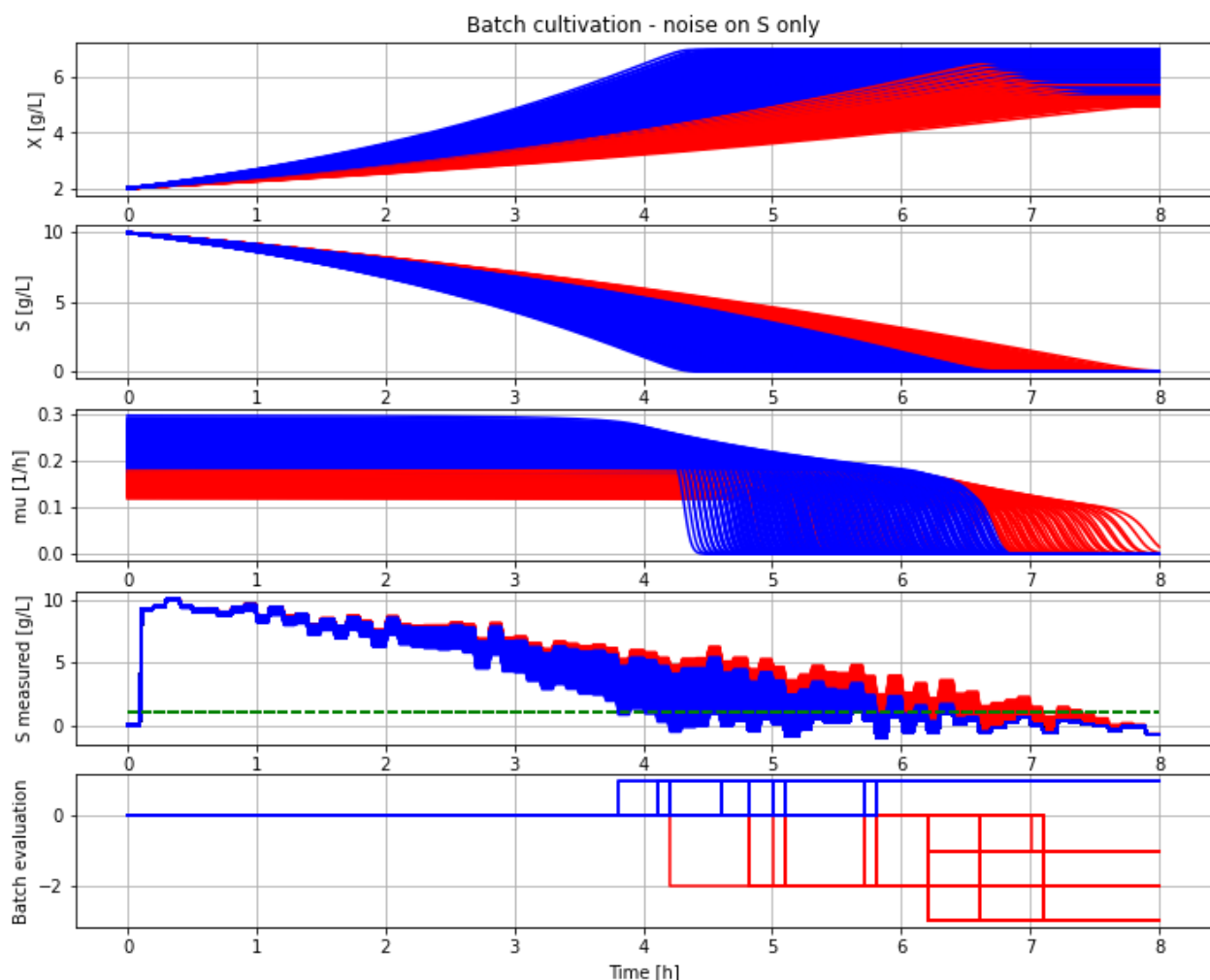
```
# Plot simulation results
```

```
if sim_res['monitor.batch_evaluation'][-1] > 0:
```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link](#).

```
ax5.plot(sim_res['time'], sim_res['bioreactor.culture.q[1]'], 'b-')
ax4.plot(sim_res['time'], sim_res['sensor.out.c[2]'], 'b-')
ax4.plot([0, simulationTime], [model.get('monitor.S_min'), model.get('m
ax5.step(sim_res['time'], sim_res['monitor.batch_evaluation'], where='pos
else:
    ax1.plot(sim_res['time'], sim_res['bioreactor.c[1]'], 'r-')
    ax2.plot(sim_res['time'], sim_res['bioreactor.c[2]'], 'r-')
    ax3.plot(sim_res['time'], sim_res['bioreactor.culture.q[1]'], 'r-')
    ax4.plot(sim_res['time'], sim_res['sensor.out.c[2]'], 'r-')
    ax4.plot([0, simulationTime], [model.get('monitor.S_min'), model.get('m
    ax5.step(sim_res['time'], sim_res['monitor.batch_evaluation'], where='pos

plt.show()
```



```
# Show end results
plt.figure()
```

```

ax1 = plt.subplot(1,2,1)
ax2 = plt.subplot(1,2,2)

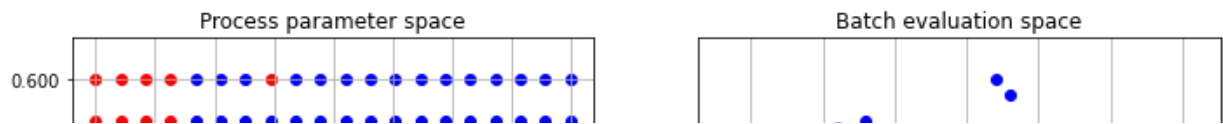
for j in range(nY):
    ax1.scatter(data[j,k,0],data[j,k,1],c='b')
    else:
        ax1.scatter(data[j,k,0],data[j,k,1],c='r')
ax1.grid()
#plt.axis([0, 0.8, 0, 0.8])
ax1.set_ylabel('qSmax [g/g,h]')
ax1.set_xlabel('Y [g/g]')
ax1.set_title('Process parameter space')

for j in range(nY):
    for k in range(nqSmax):
        if data[j,k,4] > 0:
            ax2.scatter(data[j,k,2],data[j,k,3],c='b')
        else:
            ax2.scatter(data[j,k,2],data[j,k,3],c='r')
ax2.grid()
#plt.axis([0, 8, 0, 8])
ax2.set_xlabel('T_final [h]')
ax2.set_ylabel('X_final [g/L]')
ax2.set_title('Batch evaluation space')
plt.show()

```

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link.](#)





Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link](#).

rugged.

With much more simulations we could get a better idea of the probability that a batch is accepted and determine the design space in probabilistic sense.



3 Summary

We have worked through a simple example of evaluation of batch culture with given acceptance criteria and how that criteria can be translated to acceptable variation in process parameters, i.e. the design space.

In the deterministic case we get a rather clear cut design space.

In the more realistic case with substrate measurement noise included we get a more complicated design space, but still similar.

The stochastic model introduces errors both due to the added normal noise in the substrate concentration, and due to the fact that we use a time discrete system for the noise. The impact of the time discrete check when batch has ended can be made smaller by choosing a smaller sample interval. This was not studied here and is left for the interested reader.

Note...

References

[1] Axelsson J.P. and A. Elsheikh: "An example of sensitivity analysis of a bioprocess using Bioprocess Library for Modelica", Proceedings MODPROD, Linköping, Sweden 2019, see presentation [here](#).

▼ Appendix

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

```
['bioreactor', 'bioreactor.culture', 'liquidphase', 'monitor', 'MSL', 'sensor']
```

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

```
MSL: 3.2.2 build 3 - used components: Noise.NormalNoise
```



```
system_info()
```

System information

Your URL should open in a new window. If it does not, make sure that pop ups are not blocked and [reopen the link](#).

```
-FMU by: JModelica.org
-FMI: 2.0
-Type: FMUModelCS2
-Name: BPL_TEST2.BatchWithNoise
-Generated: 2022-09-09T12:20:39
-MSL: 3.2.2 build 3
-Description: Bioprocess Library version 2.1.0 beta
-Interaction: FMU-explore ver 0.9.3
```

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 14:28

● ✕