

BPL_TEST2_Batch_design_space script with PyFMI

The key library PyFMI is installed.

After the installation a small application BPL_TEST2_Batch_design_space is loaded and run.
You can continue with this example if you like.

```
In [1]: !lsb_release -a # Actual VM Ubuntu version used by Google
```

```
No LSB modules are available.  
Distributor ID: Ubuntu  
Description:    Ubuntu 22.04.4 LTS  
Release:        22.04  
Codename:       jammy
```

```
In [2]: %env PYTHONPATH=
```

```
env: PYTHONPATH=
```

```
In [3]: !python --version
```

```
Python 3.11.11
```

```
In [4]: !wget https://repo.anaconda.com/miniconda/Miniconda3-py311_24.11.1-0-Linux-x86_64.s  
!chmod +x Miniconda3-py311_24.11.1-0-Linux-x86_64.sh  
!bash ./Miniconda3-py311_24.11.1-0-Linux-x86_64.sh -b -f -p /usr/local  
import sys  
sys.path.append('/usr/local/lib/python3.11/site-packages/')
```

```
--2025-03-26 10:07:48-- https://repo.anaconda.com/miniconda/Miniconda3-py311_24.11.1-0-Linux-x86_64.sh
Resolving repo.anaconda.com (repo.anaconda.com)... 104.16.191.158, 104.16.32.241, 2606:4700::6810:bf9e, ...
Connecting to repo.anaconda.com (repo.anaconda.com)|104.16.191.158|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 145900576 (139M) [application/octet-stream]
Saving to: 'Miniconda3-py311_24.11.1-0-Linux-x86_64.sh'

Miniconda3-py311_24 100%[=====>] 139.14M  96.5MB/s   in 1.4s

2025-03-26 10:07:50 (96.5 MB/s) - 'Miniconda3-py311_24.11.1-0-Linux-x86_64.sh' saved
[145900576/145900576]

PREFIX=/usr/local
Unpacking payload ...

Installing base environment...

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

```
In [5]: !conda update -n base -c defaults conda --yes
```

```
Channels:
- defaults
Platform: linux-64
Collecting package metadata (repodata.json): - 00\ 00| 00/ 00- 00\ 00| 00/ 00- 00\
00| 00/ 00done
Solving environment: \ 00| 00done
```

Package Plan

environment location: /usr/local

added / updated specs:

- conda

The following packages will be downloaded:

package	build	
-----	-----	
ca-certificates-2025.2.25	h06a4308_0	129 KB
certifi-2025.1.31	py311h06a4308_0	163 KB
openssl-3.0.16	h5eee18b_0	5.2 MB
-----	-----	
Total:		5.5 MB

The following packages will be UPDATED:

ca-certificates	2024.11.26-h06a4308_0 --> 2025.2.25-h06a4308_0
certifi	2024.8.30-py311h06a4308_0 --> 2025.1.31-py311h06a4308_0
openssl	3.0.15-h5eee18b_0 --> 3.0.16-h5eee18b_0

Downloading and Extracting Packages:

openssl-3.0.16	5.2 MB	: 0% 0/1 [00:00<?, ?it/s]
certifi-2025.1.31	163 KB	: 0% 0/1 [00:00<?, ?it/s]
ca-certificates-2025	129 KB	: 0% 0/1 [00:00<?, ?it/s]
ca-certificates-2025	129 KB	: 100% 1.0/1 [00:00<00:00, 30.35it/s]
openssl-3.0.16	5.2 MB	: 100% 1.0/1 [00:00<00:00, 10.10it/s]
certifi-2025.1.31	163 KB	: 10% 0.0984012204057609/1 [00:00<00:01, 1.51s/it]
certifi-2025.1.31	163 KB	: 100% 1.0/1 [00:00<00:00, 1.51s/it]

```
Preparing transaction: - 00done
Verifying transaction: | 00/ 00- 00done
Executing transaction: | 00done
```

```
In [6]: !conda --version
!python --version
```

conda 24.11.1
Python 3.11.11

In [7]: `!conda config --set channel_priority strict`

In [8]: `!conda install -c conda-forge pyfmi --yes # Install the key package`

```
Channels:
- conda-forge
- defaults
Platform: linux-64
Collecting package metadata (repodata.json): - 22\ 22| 22/ 22- 22\ 22| 22/ 22- 22\
22| 22/ 22- 22\ 22| 22/ 22- 22\ 22| 22/ 22- 22\ 22| 22/ 22- 22\ 22| 22/ 22- 22\ 22| 22/ 22- 22\ 22|
22/ 22- 22\ 22| 22/ 22- 22\ 22| 22/ 22done
Solving environment: \ 22| 22/ 22- 22done
```

Package Plan

environment location: /usr/local

added / updated specs:

- pyfmi

The following packages will be downloaded:

package	build		
-----	-----		
_x86_64-microarch-level-3	2_broadwell	8 KB	conda-forge
assimulo-3.6.0	py311h083bc19_0	1.1 MB	conda-forge
certifi-2025.1.31	pyhd8ed1ab_0	159 KB	conda-forge
conda-25.1.1	py311h38be061_1	1.1 MB	conda-forge
fmilib-2.4.1	hac33072_1	383 KB	conda-forge
gmp-6.3.0	hac33072_2	449 KB	conda-forge
libamd-3.3.3	haaf9dc3_7100102	49 KB	conda-forge
libblas-3.9.0	31_h59b9bed_openblas	16 KB	conda-forge
libbtf-2.3.2	h32481e8_7100102	27 KB	conda-forge
libcamd-3.3.3	h32481e8_7100102	46 KB	conda-forge
libcbblas-3.9.0	31_he106b2a_openblas	16 KB	conda-forge
libccolamd-3.3.4	h32481e8_7100102	42 KB	conda-forge
libcholmod-5.3.1	h59ddab4_7100102	1.1 MB	conda-forge
libcolamd-3.3.4	h32481e8_7100102	33 KB	conda-forge
libcxsparse-4.4.1	h32481e8_7100102	118 KB	conda-forge
libgcc-14.2.0	h767d61c_2	828 KB	conda-forge
libgcc-ng-14.2.0	h69a702a_2	52 KB	conda-forge
libgfortran-14.2.0	h69a702a_2	52 KB	conda-forge
libgfortran-ng-14.2.0	h69a702a_2	53 KB	conda-forge
libgfortran5-14.2.0	hf1ad2bd_2	1.4 MB	conda-forge
libgomp-14.2.0	h767d61c_2	449 KB	conda-forge
libklu-2.3.5	hf24d653_7100102	142 KB	conda-forge
liblapack-3.9.0	31_h7ac8fdf_openblas	16 KB	conda-forge
libldl-3.3.2	h32481e8_7100102	24 KB	conda-forge
libopenblas-0.3.29	pthreads_h94d23a6_0	5.6 MB	conda-forge
libparu-1.0.0	h17147ab_7100102	91 KB	conda-forge
librbio-4.3.4	h32481e8_7100102	47 KB	conda-forge
libspex-3.2.3	had10066_7100102	79 KB	conda-forge
libspqr-4.3.4	h852d39f_7100102	213 KB	conda-forge
libstdcxx-14.2.0	h8f9b012_2	3.7 MB	conda-forge
libstdcxx-ng-14.2.0	h4852527_2	53 KB	conda-forge
libsuitesparseconfig-7.10.1	h92d6892_7100102	42 KB	conda-forge
libumfpack-6.3.5	heb53515_7100102	424 KB	conda-forge
metis-5.1.0	hd0bcaf9_1007	3.7 MB	conda-forge
mpfr-4.2.1	h90cbb55_3	620 KB	conda-forge

numpy-2.2.4	py311h5d046bc_0	8.6 MB	conda-forge
openssl-3.4.1	h7b32b05_0	2.8 MB	conda-forge
pyfmi-2.16.3	py311h9f3472d_0	5.2 MB	conda-forge
python_abi-3.11	2_cp311	5 KB	conda-forge
scipy-1.15.2	py311h8f841c2_0	16.4 MB	conda-forge
suitesparse-7.10.1	ha0f6916_7100102	12 KB	conda-forge
sundials-7.1.1	ha52427a_0	907 KB	conda-forge

Total:		56.1 MB	

The following NEW packages will be INSTALLED:

_x86_64-microarch~	conda-forge/noarch::_x86_64-microarch-level-3-2_broadwell
assimulo	conda-forge/linux-64::assimulo-3.6.0-py311h083bc19_0
fmilib	conda-forge/linux-64::fmilib-2.4.1-hac33072_1
gmp	conda-forge/linux-64::gmp-6.3.0-hac33072_2
libamd	conda-forge/linux-64::libamd-3.3.3-haaf9dc3_7100102
libblas	conda-forge/linux-64::libblas-3.9.0-31_h59b9bed_openblas
libbtf	conda-forge/linux-64::libbtf-2.3.2-h32481e8_7100102
libcamd	conda-forge/linux-64::libcamd-3.3.3-h32481e8_7100102
libcbblas	conda-forge/linux-64::libcbblas-3.9.0-31_he106b2a_openblas
libccolamd	conda-forge/linux-64::libccolamd-3.3.4-h32481e8_7100102
libcholmod	conda-forge/linux-64::libcholmod-5.3.1-h59ddab4_7100102
libcolamd	conda-forge/linux-64::libcolamd-3.3.4-h32481e8_7100102
libcxsparse	conda-forge/linux-64::libcxsparse-4.4.1-h32481e8_7100102
libgcc	conda-forge/linux-64::libgcc-14.2.0-h767d61c_2
libgfortran	conda-forge/linux-64::libgfortran-14.2.0-h69a702a_2
libgfortran-ng	conda-forge/linux-64::libgfortran-ng-14.2.0-h69a702a_2
libgfortran5	conda-forge/linux-64::libgfortran5-14.2.0-hf1ad2bd_2
libklu	conda-forge/linux-64::libklu-2.3.5-hf24d653_7100102
liblapack	conda-forge/linux-64::liblapack-3.9.0-31_h7ac8fdf_openblas
libldl	conda-forge/linux-64::libldl-3.3.2-h32481e8_7100102
libopenblas	conda-forge/linux-64::libopenblas-0.3.29-pthreads_h94d23a6_0
libparu	conda-forge/linux-64::libparu-1.0.0-h17147ab_7100102
librbio	conda-forge/linux-64::librbio-4.3.4-h32481e8_7100102
libspex	conda-forge/linux-64::libspex-3.2.3-had10066_7100102
libspqr	conda-forge/linux-64::libspqr-4.3.4-h852d39f_7100102
libstdcxx	conda-forge/linux-64::libstdcxx-14.2.0-h8f9b012_2
libsuitesparsecon~	conda-forge/linux-64::libsuitesparseconfig-7.10.1-h92d6892_7100102
libumfpack	conda-forge/linux-64::libumfpack-6.3.5-heb53515_7100102
metis	conda-forge/linux-64::metis-5.1.0-hd0bc9f9_1007
mpfr	conda-forge/linux-64::mpfr-4.2.1-h90cbb55_3
numpy	conda-forge/linux-64::numpy-2.2.4-py311h5d046bc_0
pyfmi	conda-forge/linux-64::pyfmi-2.16.3-py311h9f3472d_0
python_abi	conda-forge/linux-64::python_abi-3.11-2_cp311
scipy	conda-forge/linux-64::scipy-1.15.2-py311h8f841c2_0
suitesparse	conda-forge/linux-64::suitesparse-7.10.1-ha0f6916_7100102
sundials	conda-forge/linux-64::sundials-7.1.1-ha52427a_0

The following packages will be UPDATED:

conda	pkgs/main::conda-24.11.1-py311h06a430~ --> conda-forge::conda-25.1.1-py311h38be061_1
libgcc-ng	pkgs/main::libgcc-ng-11.2.0-h1234567_1 --> conda-forge::libgcc-ng-14.2.0-h69a702a_2

```
libgomp                pkgs/main::libgomp-11.2.0-h1234567_1 --> conda-forge::libgomp
-14.2.0-h767d61c_2
libstdcxx-ng           pkgs/main::libstdcxx-ng-11.2.0-h12345~ --> conda-forge::libstdc
xx-ng-14.2.0-h4852527_2
openssl                pkgs/main::openssl-3.0.16-h5eee18b_0 --> conda-forge::openssl
-3.4.1-h7b32b05_0
```

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

```
certifi                pkgs/main/linux-64::certifi-2025.1.31~ --> conda-forge/noarch::
certifi-2025.1.31-pyhd8ed1ab_0
```

Downloading and Extracting Packages:

```
scipy-1.15.2           | 16.4 MB | : 0% 0/1 [00:00<?, ?it/s]
numpy-2.2.4            | 8.6 MB  | : 0% 0/1 [00:00<?, ?it/s]
```

```
libopenblas-0.3.29    | 5.6 MB  | : 0% 0/1 [00:00<?, ?it/s]
```

```
pyfmi-2.16.3          | 5.2 MB  | : 0% 0/1 [00:00<?, ?it/s]
```

```
metis-5.1.0           | 3.7 MB  | : 0% 0/1 [00:00<?, ?it/s]
```

```
libstdcxx-14.2.0      | 3.7 MB  | : 0% 0/1 [00:00<?, ?it/s]
```

```
openssl-3.4.1         | 2.8 MB  | : 0% 0/1 [00:00<?, ?it/s]
```

```
libgfortran5-14.2.0   | 1.4 MB  | : 0% 0/1 [00:00<?, ?it/s]
```

```
conda-25.1.1          | 1.1 MB  | : 0% 0/1 [00:00<?, ?it/s]
```

assimulo-3.6.0 | 1.1 MB | : 0% 0/1 [00:00<?, ?it/s]

libcholmod-5.3.1 | 1.1 MB | : 0% 0/1 [00:00<?, ?it/s]

sundials-7.1.1 | 907 KB | : 0% 0/1 [00:00<?, ?it/s]

libgcc-14.2.0 | 828 KB | : 0% 0/1 [00:00<?, ?it/s]

mpfr-4.2.1 | 620 KB | : 0% 0/1 [00:00<?, ?it/s]

gmp-6.3.0 | 449 KB | : 0% 0/1 [00:00<?, ?it/s]

libgomp-14.2.0 | 449 KB | : 0% 0/1 [00:00<?, ?it/s]

libumfpack-6.3.5 | 424 KB | : 0% 0/1 [00:00<?, ?it/s]

fmilib-2.4.1 | 383 KB | : 0% 0/1 [00:00<?, ?it/s]

libspqr-4.3.4 | 213 KB | : 0% 0/1 [00:00<?, ?it/s]

scipy-1.15.2 | 16.4 MB | : 3% 0.032399925412051306/1 [00:00<00:02, 3.09
s/it]

numpy-2.2.4 | 8.6 MB | : 20% 0.20195372604482412/1 [00:00<00:00, 2.02i
t/s]

libopenblas-0.3.29 | 5.6 MB | : 32% 0.32107645378971256/1 [00:00<00:00, 3.20i
t/s]

scipy-1.15.2 | 16.4 MB | : 41% 0.4059520066333487/1 [00:00<00:00, 2.32i
t/s]

numpy-2.2.4 | 8.6 MB | : 98% 0.9751999744146462/1 [00:00<00:00, 5.37i
t/s]

pyfmi-2.16.3 | 5.2 MB | : 0% 0.002983953056648666/1 [00:00<01:08, 68.53
s/it]

libopenblas-0.3.29	5.6 MB	: 100% 1.0/1 [00:00<00:00, 3.86it/s]
libopenblas-0.3.29	5.6 MB	: 100% 1.0/1 [00:00<00:00, 3.86it/s]
scipy-1.15.2	16.4 MB	: 69% 0.6870690065320292/1 [00:00<00:00, 2.54i t/s]
libstdcxx-14.2.0	3.7 MB	: 0% 0.0042177278432850495/1 [00:00<01:09, 69.4 2s/it]
pyfmi-2.16.3	5.2 MB	: 61% 0.6117103766129766/1 [00:00<00:00, 2.51i t/s]
numpy-2.2.4	8.6 MB	: 100% 1.0/1 [00:00<00:00, 5.37it/s]
metis-5.1.0	3.7 MB	: 100% 1.0/1 [00:00<00:00, 3.29it/s]
scipy-1.15.2	16.4 MB	: 94% 0.9415037149149026/1 [00:00<00:00, 2.19i t/s]
libgfortran5-14.2.0	1.4 MB	: 1% 0.011206734985068174/1 [00:00<00:39, 39.45 s/it]
pyfmi-2.16.3	5.2 MB	: 100% 1.0/1 [00:00<00:00, 2.36it/s]
pyfmi-2.16.3	5.2 MB	: 100% 1.0/1 [00:00<00:00, 2.36it/s]
libstdcxx-14.2.0	3.7 MB	: 100% 1.0/1 [00:00<00:00, 2.56it/s]
libstdcxx-14.2.0	3.7 MB	: 100% 1.0/1 [00:00<00:00, 2.56it/s]

libgfortran5-14.2.0 | 1.4 MB | : 100% 1.0/1 [00:00<00:00, 39.45s/it]

openssl-3.4.1 | 2.8 MB | : 100% 1.0/1 [00:00<00:00, 2.36it/s]

openssl-3.4.1 | 2.8 MB | : 100% 1.0/1 [00:00<00:00, 2.36it/s]

assimulo-3.6.0 | 1.1 MB | : 1% 0.014703493605362324/1 [00:00<00:36, 37.54
s/it]

assimulo-3.6.0 | 1.1 MB | : 100% 1.0/1 [00:00<00:00, 37.54s/it]

libgcc-14.2.0 | 828 KB | : 2% 0.01932337522187561/1 [00:00<00:31, 32.00
s/it]

sundials-7.1.1 | 907 KB | : 2% 0.01763373830085844/1 [00:00<00:34, 35.20
s/it]

libcholmod-5.3.1 | 1.1 MB | : 1% 0.014870549794649543/1 [00:00<00:41, 41.97
s/it]

conda-25.1.1 | 1.1 MB | : 1% 0.013622478419712683/1 [00:00<00:46, 47.33
s/it]

libgcc-14.2.0 | 828 KB | : 100% 1.0/1 [00:00<00:00, 32.00s/it]

libcholmod-5.3.1 | 1.1 MB | : 100% 1.0/1 [00:00<00:00, 1.84it/s]

libcholmod-5.3.1 | 1.1 MB | : 100% 1.0/1 [00:00<00:00, 1.84it/s]

sundials-7.1.1 | 907 KB | : 100% 1.0/1 [00:00<00:00, 1.80it/s]

sundials-7.1.1 | 907 KB | : 100% 1.0/1 [00:00<00:00, 1.80it/s]

conda-25.1.1 | 1.1 MB | : 100% 1.0/1 [00:00<00:00, 1.80it/s]

conda-25.1.1 | 1.1 MB | : 100% 1.0/1 [00:00<00:00, 1.80it/s]

mpfr-4.2.1 | 620 KB | : 3% 0.025811696239942908/1 [00:00<00:28, 28.90
s/it]

gmp-6.3.0 | 449 KB | : 4% 0.03561313321233331/1 [00:00<00:20, 21.33
s/it]

libgomp-14.2.0 | 449 KB | : 4% 0.03562807972826631/1 [00:00<00:20, 21.40
s/it]

gmp-6.3.0 | 449 KB | : 100% 1.0/1 [00:00<00:00, 21.33s/it]

libgomp-14.2.0 | 449 KB | : 100% 1.0/1 [00:00<00:00, 21.40s/it]

mpfr-4.2.1 | 620 KB | : 100% 1.0/1 [00:00<00:00, 28.90s/it]

... (more hidden) ...

... (more hidden) ...

libspqr-4.3.4 | 213 KB | : 8% 0.07503068271326775/1 [00:00<00:09, 10.79
s/it]

libspqr-4.3.4 | 213 KB | : 100% 1.0/1 [00:00<00:00, 10.79s/it]

scipy-1.15.2 | 16.4 MB | : 100% 1.0/1 [00:00<00:00, 2.19it/s]

fmilib-2.4.1	383 KB	: 4% 0.04180391656566945/1 [00:00<00:19, 19.94s/it]
--------------	--------	---

libumfpack-6.3.5	424 KB	: 100% 1.0/1 [00:00<00:00, 21.95s/it]
------------------	--------	---------------------------------------

fmilib-2.4.1	383 KB	: 100% 1.0/1 [00:00<00:00, 19.94s/it]
--------------	--------	---------------------------------------

libopenblas-0.3.29	5.6 MB	: 100% 1.0/1 [00:01<00:00, 3.86it/s]
--------------------	--------	--------------------------------------

metis-5.1.0	3.7 MB	: 100% 1.0/1 [00:01<00:00, 3.29it/s]
-------------	--------	--------------------------------------

pyfmi-2.16.3	5.2 MB	: 100% 1.0/1 [00:01<00:00, 2.36it/s]
--------------	--------	--------------------------------------

libstdc++-14.2.0	3.7 MB	: 100% 1.0/1 [00:01<00:00, 2.56it/s]
------------------	--------	--------------------------------------

libgfortran5-14.2.0	1.4 MB	: 100% 1.0/1 [00:01<00:00, 1.69s/it]
---------------------	--------	--------------------------------------

libgfortran5-14.2.0	1.4 MB	: 100% 1.0/1 [00:01<00:00, 1.69s/it]
---------------------	--------	--------------------------------------

openssl-3.4.1	2.8 MB	: 100% 1.0/1 [00:02<00:00, 2.36it/s]
numpy-2.2.4	8.6 MB	: 100% 1.0/1 [00:02<00:00, 5.37it/s]

libgcc-14.2.0	828 KB	: 100% 1.0/1 [00:02<00:00, 2.00s/it]
---------------	--------	--------------------------------------

libgcc-14.2.0	828 KB	: 100% 1.0/1 [00:02<00:00, 2.00s/it]
---------------	--------	--------------------------------------

assimulo-3.6.0	1.1 MB	: 100% 1.0/1 [00:02<00:00, 2.05s/it]
----------------	--------	--------------------------------------

assimulo-3.6.0 | 1.1 MB | : 100% 1.0/1 [00:02<00:00, 2.05s/it]

libcholmod-5.3.1 | 1.1 MB | : 100% 1.0/1 [00:02<00:00, 1.84it/s]

sundials-7.1.1 | 907 KB | : 100% 1.0/1 [00:02<00:00, 1.80it/s]

gmp-6.3.0 | 449 KB | : 100% 1.0/1 [00:02<00:00, 2.21s/it]

gmp-6.3.0 | 449 KB | : 100% 1.0/1 [00:02<00:00, 2.21s/it]

mpfr-4.2.1 | 620 KB | : 100% 1.0/1 [00:02<00:00, 2.28s/it]

mpfr-4.2.1 | 620 KB | : 100% 1.0/1 [00:02<00:00, 2.28s/it]

libgomp-14.2.0 | 449 KB | : 100% 1.0/1 [00:02<00:00, 2.36s/it]

libgomp-14.2.0 | 449 KB | : 100% 1.0/1 [00:02<00:00, 2.36s/it]

... (more hidden) ...

... (more hidden) ...

libspqr-4.3.4 | 213 KB | : 100% 1.0/1 [00:02<00:00, 2.43s/it]

libspqr-4.3.4 | 213 KB | : 100% 1.0/1 [00:02<00:00, 2.43s/it]

conda-25.1.1 | 1.1 MB | : 100% 1.0/1 [00:02<00:00, 1.80it/s]

libumfpack-6.3.5 | 424 KB | : 100% 1.0/1 [00:02<00:00, 2.47s/it]

libumfpack-6.3.5 | 424 KB | : 100% 1.0/1 [00:02<00:00, 2.47s/it]

fmilib-2.4.1 | 383 KB | : 100% 1.0/1 [00:02<00:00, 2.56s/it]

scipy-1.15.2 | 16.4 MB | : 100% 1.0/1 [00:03<00:00, 2.19it/s]


```

Preparing transaction: - 00\ 00done
Verifying transaction: / 00- 00\ 00| 00/ 00done
Executing transaction: \ 00| 00/ 00- 00\ 00| 00/ 00- 00\ 00| 00/ 00- 00\ 00| 00/ 00-
00\ 00| 00/ 00- 00\ 00| 00/ 00- 00\ 00| 00done

```

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_om_me.fmu
- Setup-file - BPL_TEST2_Batch_design_space_no_noise_explore.py
- FMU - BPL_TEST2_Batch_design_space_with_noise_linux_om_me.fmu
- Setup-file - BPL_TEST2_Batch_design_space_with_noise_explore.py

```

In [9]: # Filter out DeprecationWarnings for 'np.float as alias' is needed - wish I could m
import warnings
warnings.filterwarnings("ignore")

```

```

In [10]: %%bash
git clone https://github.com/janpeter19/BPL_TEST2_Batch_design_space

```

Cloning into 'BPL_TEST2_Batch_design_space'...

```

In [11]: %cd BPL_TEST2_Batch_design_space
/content/BPL_TEST2_Batch_design_space

```

BPL_TEST2_Batch_design_space - demo

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 subdstrate 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:

- $X_{\text{final}} > X_{\text{final_min}}$
- $\text{Time_final} < \text{time_final_max}$

The question is what range of process parameters Y and q_{Smax} 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.

```
In [12]: run -i BPL_TEST2_Batch_no_noise_explore.py
```

Linux - run FMU pre-compiled OpenModelica

Model for the process 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()`

```
In [13]: # Adjust the diagram size
         %matplotlib inline
```



```
plt.rcParams['figure.figsize'] = [30/2.54, 24/2.54]
```

1.1 Batch evaluation

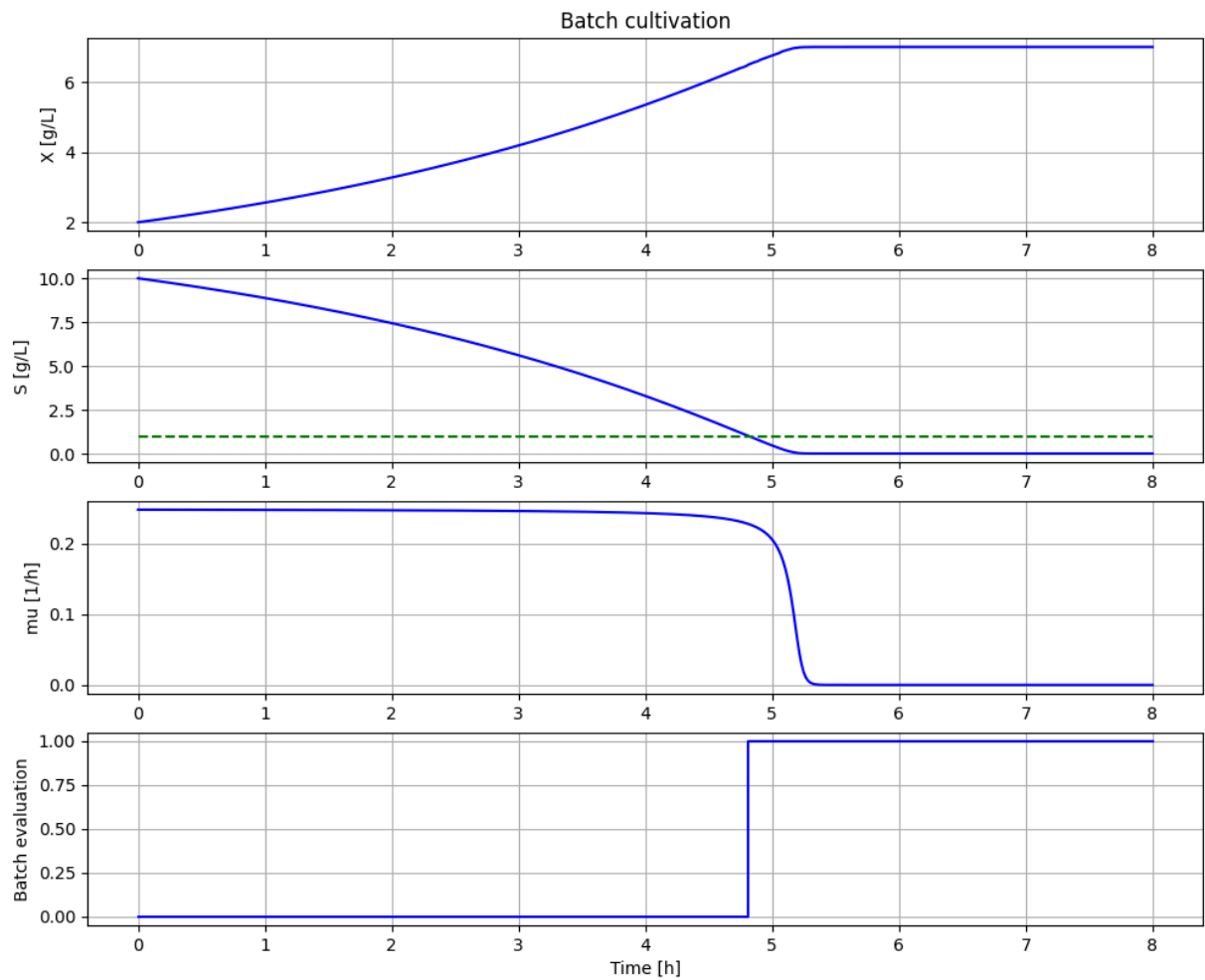
The first an example of batch that has an end of batch that fulfills the criteria for acceptance. In the following diagram we see examples of impact of variation on the criteria for acceptance.

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.

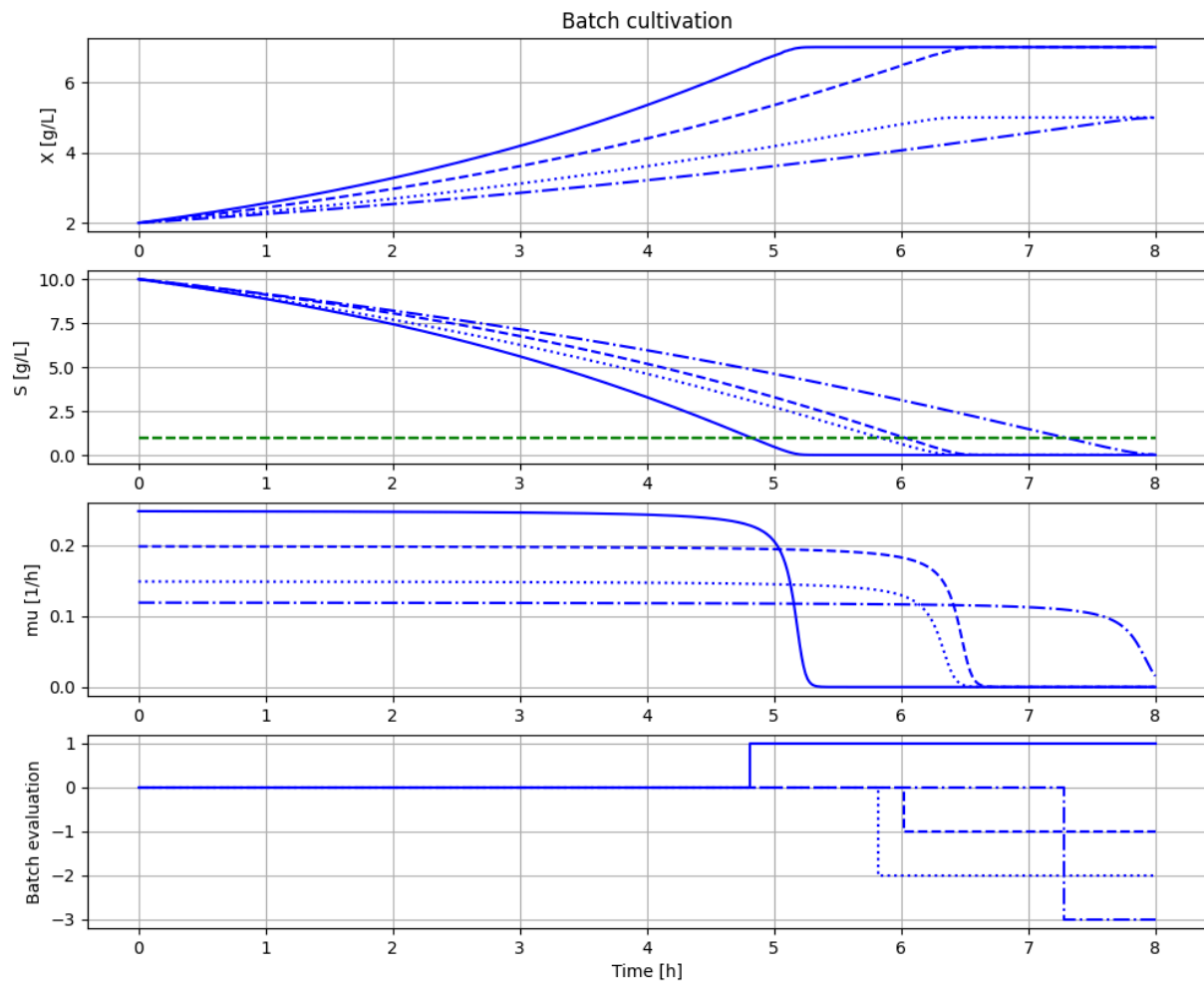
```
In [14]: # Nominal parameters
par(S_min=1.0, time_final_max=6.0, X_final_min=5.0)
init(VX_start=2, VS_start=10)
par(Y=0.5, qSmax=0.5, Ks=0.1)
```

```
In [15]: # Simulation of nominal parameters that gives a batch that meet the end criteria
newplot(plotType='TimeSeries_2')
simu(8)
```

```
Could not find cannot import name 'dopri5' from 'assimulo.lib' (/usr/local/lib/python3.11/site-packages/assimulo/lib/__init__.py)
Could not find cannot import name 'rodas' from 'assimulo.lib' (/usr/local/lib/python3.11/site-packages/assimulo/lib/__init__.py)
Could not find cannot import name 'odassl' from 'assimulo.lib' (/usr/local/lib/python3.11/site-packages/assimulo/lib/__init__.py)
Could not find ODEPACK functions.
Could not find RADAR5
Could not find GLIMDA.
```



```
In [16]: # 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.

1.2 Batch evaluation under process variation - parameter sweep

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

```
In [17]: # 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])
```

```
In [18]: # Run parameter sweep - takes a few minutes
newplot(plotType='TimeSeries_2_diagrams')
init(VX_start=2, VS_start=10)

for j in range(nY):
    for k in range(nqSmax):
```

```

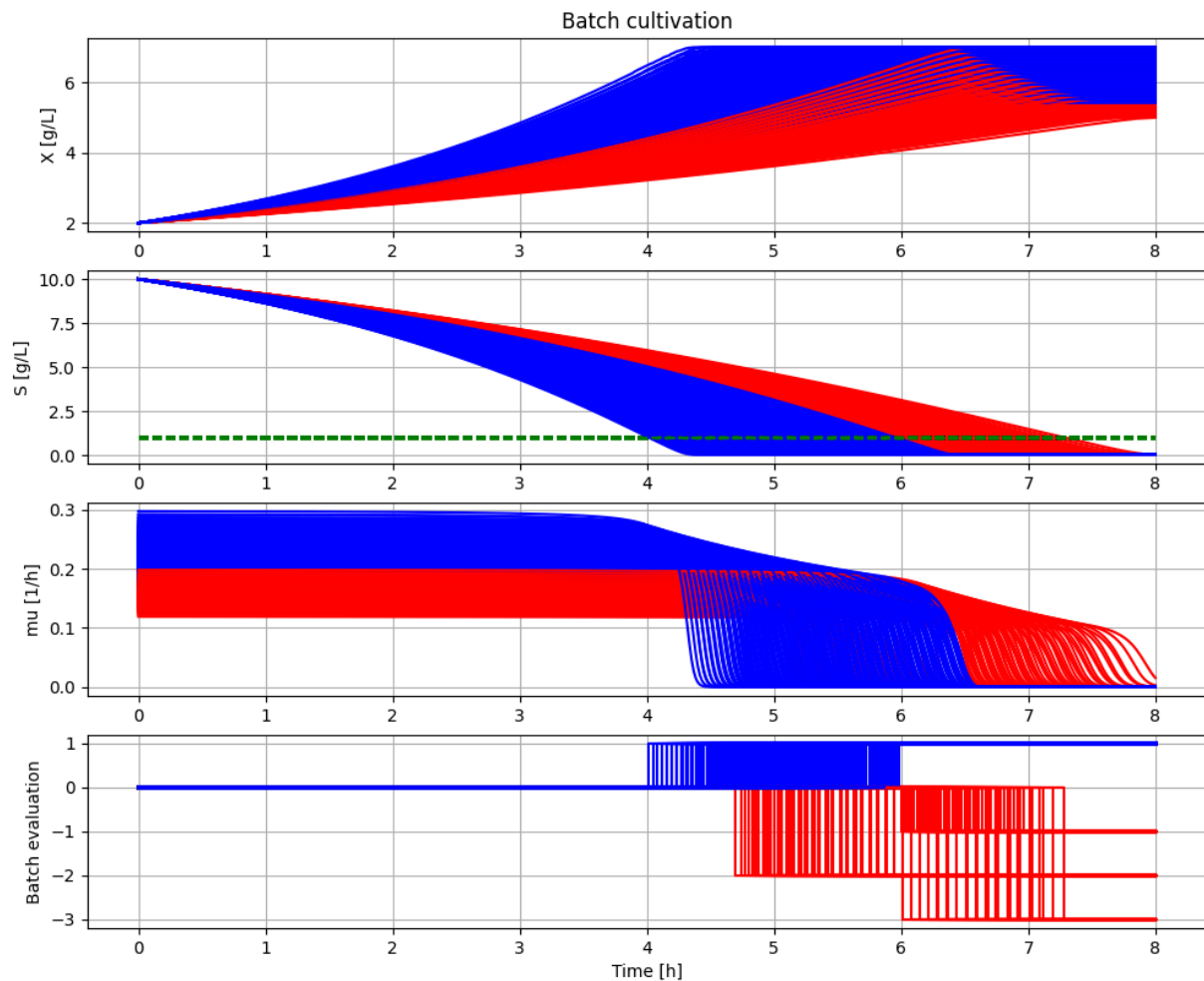
par(Y=Y_range[j])
par(qSmax=qSmax_range[k])
simu(8)

# 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.

```
In [19]: # Show end results
plt.figure()
ax1 = plt.subplot(1,2,1)
ax2 = plt.subplot(1,2,2)

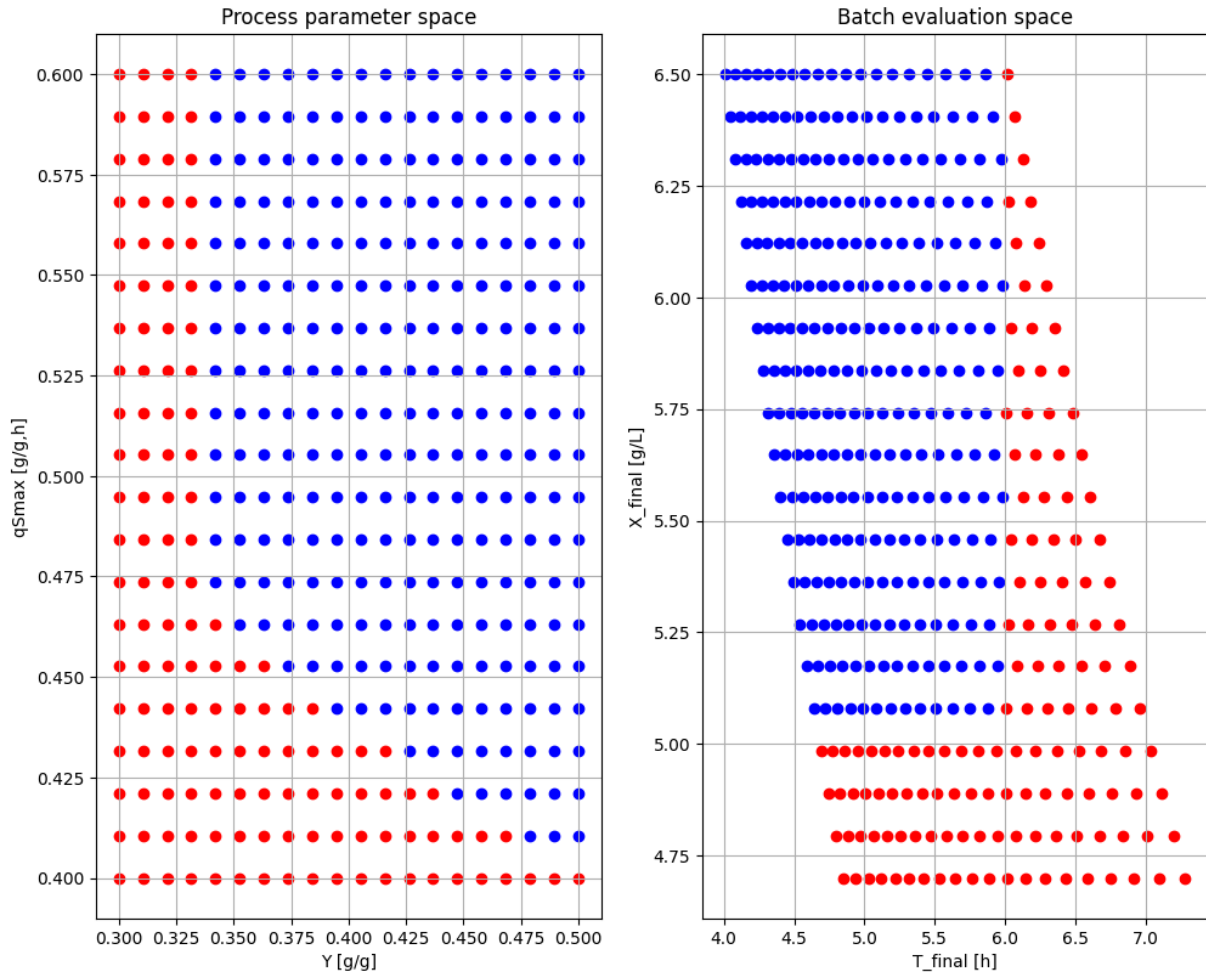
for j in range(nY):
    for k in range(nqSmax):
        if data[j,k,4] > 0:
            ax1.scatter(data[j,k,0],data[j,k,1],c='b')
        else:
            ax1.scatter(data[j,k,0],data[j,k,1],c='r')
ax1.grid()
#plt.axis([0, 0.8, 0, 0.8])
ax1.set_ylabel('qSmax [g/g,h]')
ax1.set_xlabel('Y [g/g]')
ax1.set_title('Process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

```



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) represent a simulation that give 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 concentrations.

Thus detection of end of batch is now in discrete time with a give samplePeriod (default 0.1 hour). This discreteization also introduce an error in detection of the end point. By changing this sample intervall to shorter values you can see the impact of this error but not done here.

```
In [20]: run -i BPL_TEST2_Batch_with_noise_explore.py
```

Linux - run FMU pre-compiled OpenModelica

Model for the process 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()

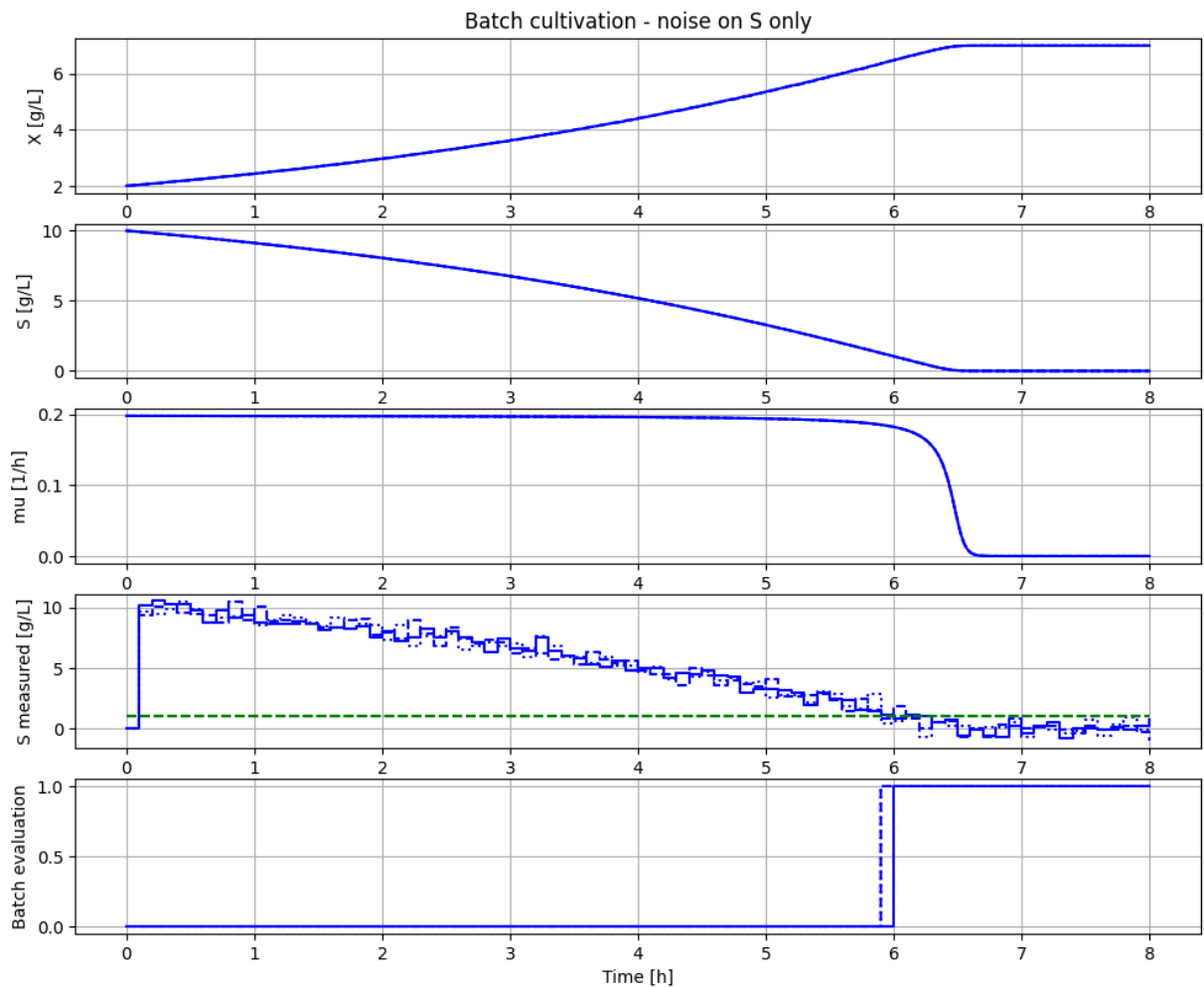
<Figure size 1181.1x944.882 with 0 Axes>

2.1 Batch evaluation under substrate measurement error

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

```
In [21]: # Nominal parameters
par(S_min=1.0, time_final_max=6.0, X_final_min=5.0)
init(VX_start=2, VS_start=10)
par(Y=0.5, qSmax=0.5, Ks=0.1)
par(sigma=0.48, samplePeriod=0.1)
```

```
In [22]: # Simulation of nominal parameters that gives a batch that meed 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.

```
In [23]: # 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])
```

```
In [24]: # 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)
```



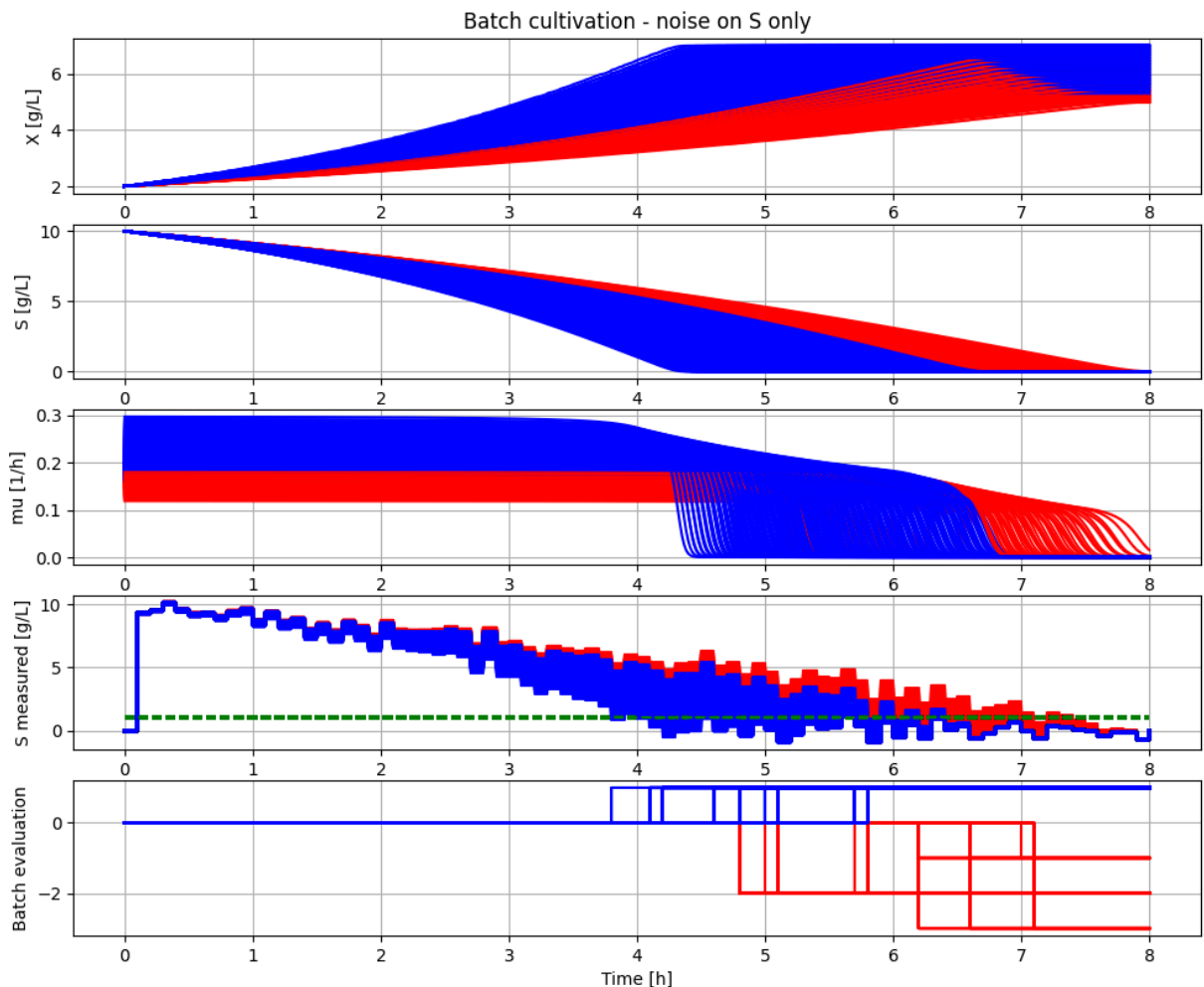
```

# 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:
    ax1.plot(sim_res['time'], sim_res['bioreactor.c[1]'], 'b-')
    ax2.plot(sim_res['time'], sim_res['bioreactor.c[2]'], 'b-')
    ax3.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()

```



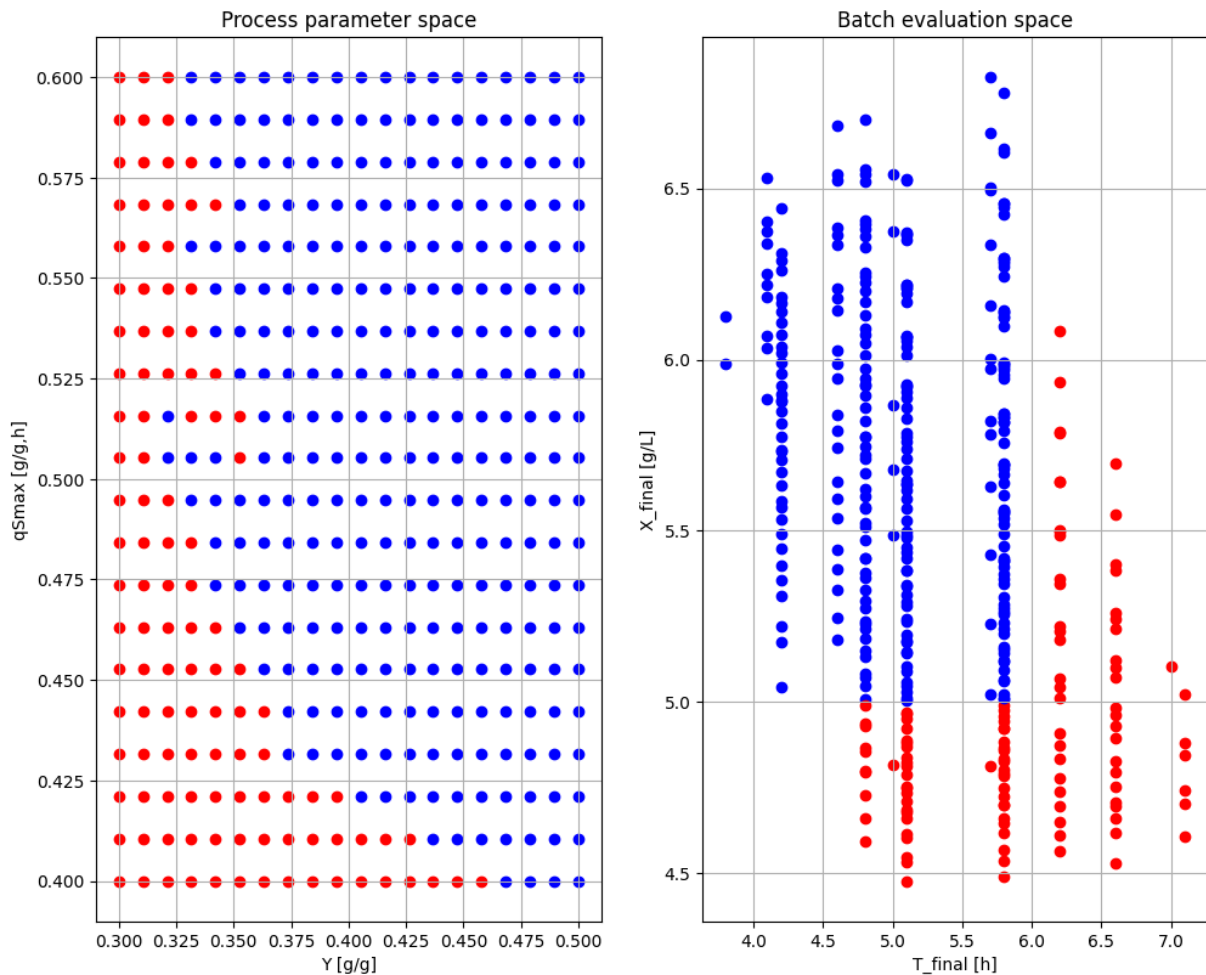
```

In [25]: # 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.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()

```



We see that we get somewhat different results in the parameter space. The acceptable region with blue dots (design space) get a more rounded corner. The vertical left line is also more 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 introduce errors both due to the added normal noise in the substrate concentration, and due to the fact that we use 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

```
In [26]: describe('parts')
```

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

```
In [27]: describe('MSL')
```

```
MSL: 3.2.3 - used components: Noise.NormalNoise
```

```
In [28]: system_info()
```

```
System information
```

```
-OS: Linux
```

```
-Python: 3.11.11
```

```
-Scipy: not installed in the notebook
```

```
-PyFMI: 2.16.3
```

```
-FMU by: OpenModelica Compiler OpenModelica 1.25.0~dev-133-ga5470be
```

```
-FMI: 2.0
```

```
-Type: FMUModelME2
```

```
-Name: BPL_TEST2.BatchWithNoise
```

```
-Generated: 2024-11-08T08:51:32Z
```

```
-MSL: 3.2.3
```

```
-Description: Bioprocess Library version 2.3.0
```

```
-Interaction: FMU-explore version 1.0.0
```