

BPL_TEST2_Batch_design_space - demo

```
In [1]: run -i BPL_TEST2_Batch_explore.py
```

Windows - run FMU pre-compiled JModelica 2.14

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

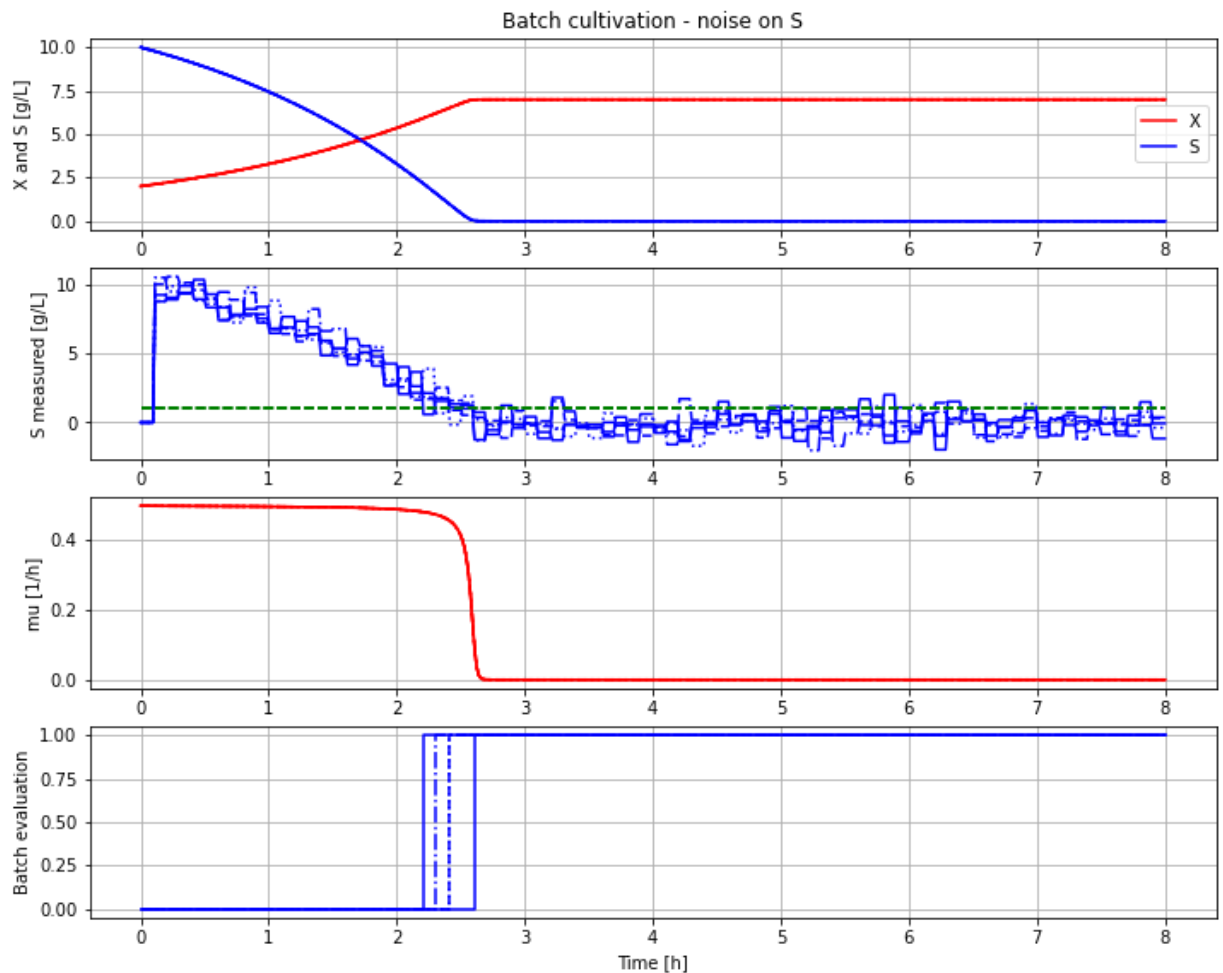
Brief information about a command by help(), eg help(simu)

Key system information is listed with the command system_info()

```
In [2]: plt.rcParams['figure.figsize'] = [30/2.54, 24/2.54]
```

Batch evaluation

```
In [3]: newplot()
init(VX_0=2, VS_0=10)
par(sigma=0.75)
for value in [1,2,3,4,5]: par(seed=value); simu(8)
```



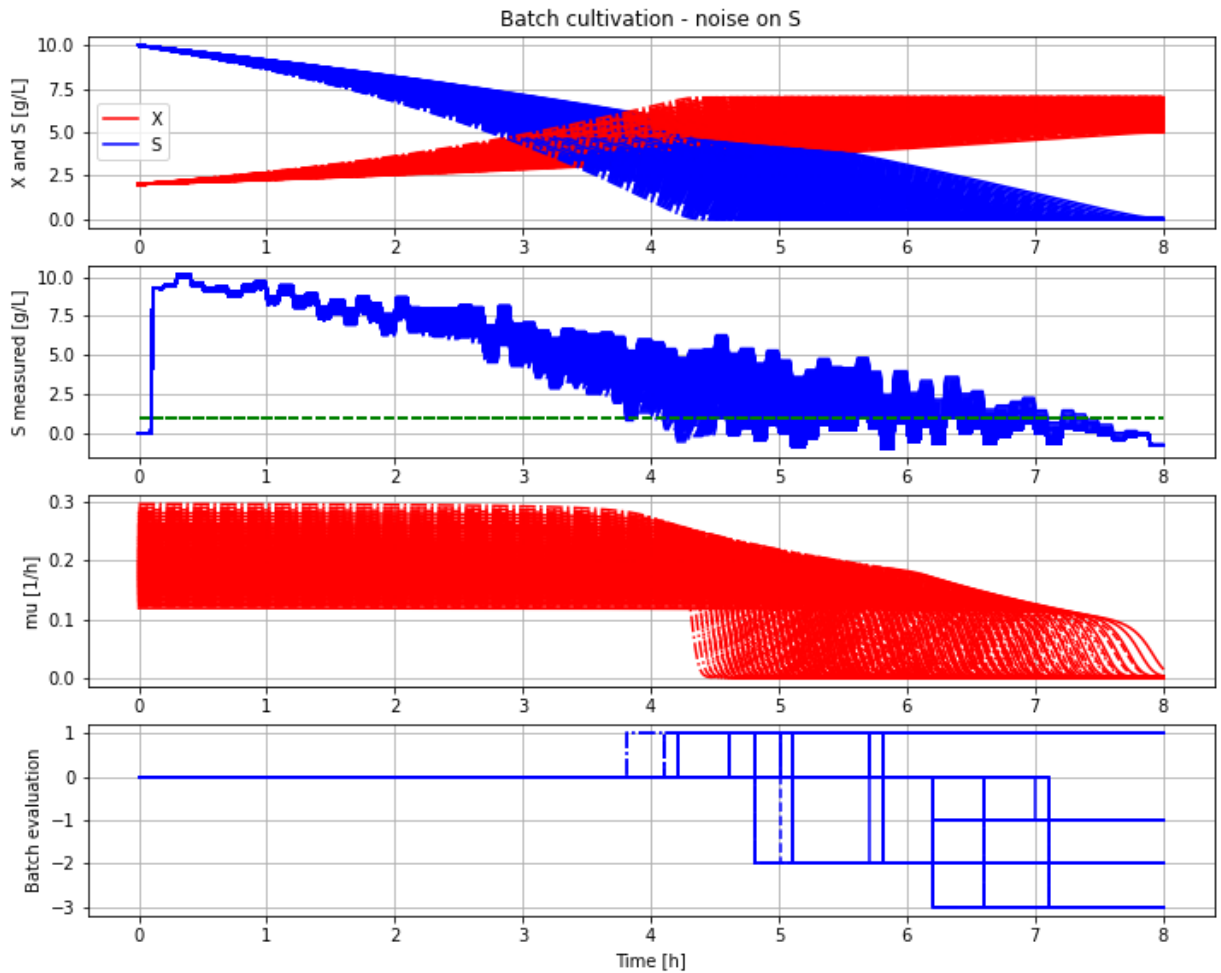
Batch evaluation under process variation - parameter sweep

```
In [4]: # Define sweep ranges and storage of final data
nY = 20
nqSmax = 20
Yxs_range = np.linspace(0.3, 0.5, nY)
qSmax_range = np.linspace(0.4, 0.6, nqSmax)
data = np.zeros([nY, nqSmax, 5])
```

```
In [5]: # Run parameter sweep
newplot()
init(VX_0=2, VS_0=10)
par(sigma=0.48, seed=1)

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

    data[j, k, 0] = Yxs_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]
```

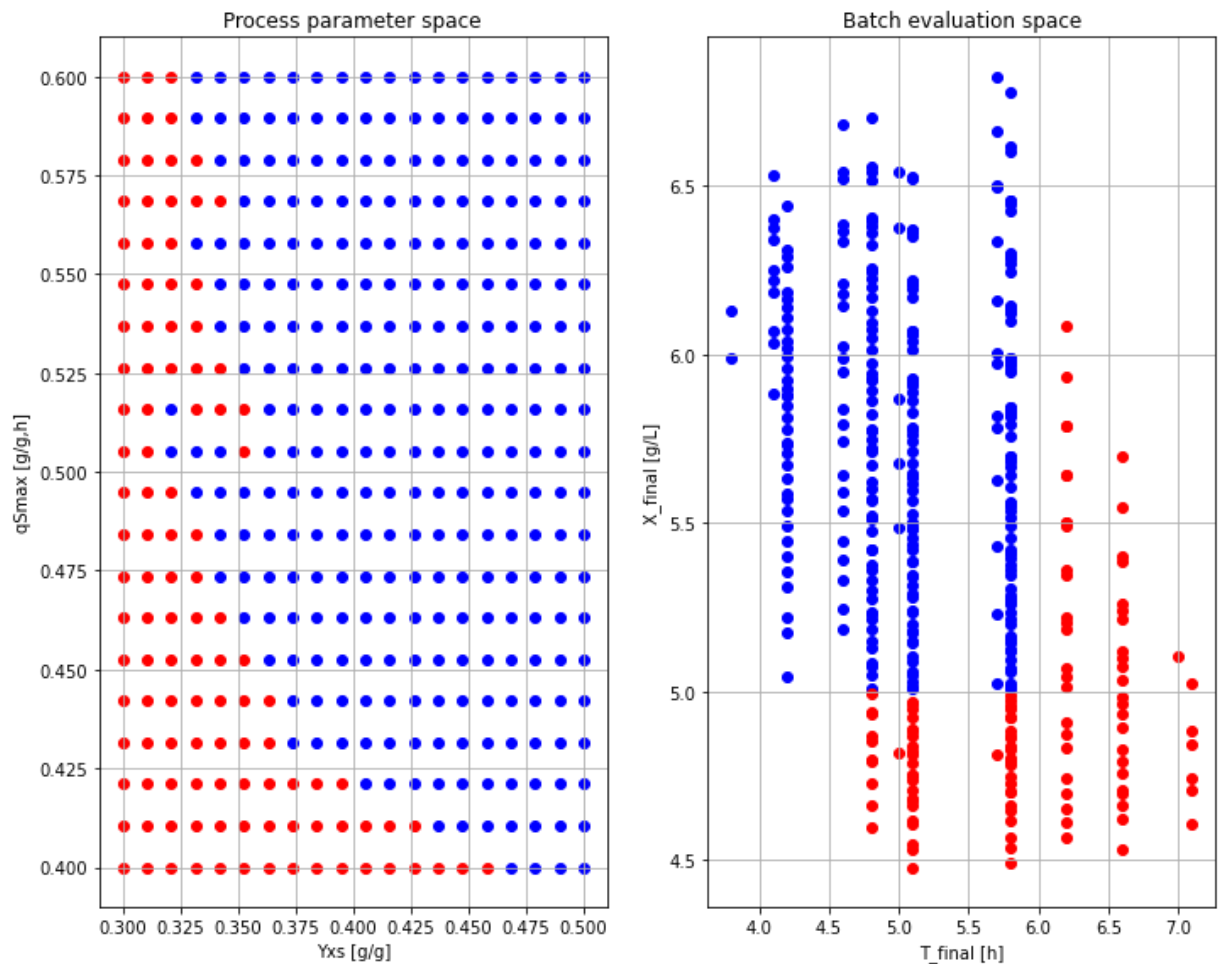


In [6]:

```
# 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('Yxs [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()
```



Summary

Appendix

In [7]: `describe('parts')`

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

In [8]: `model.get('MSL.usage')`

Out[8]: `['Noise.NormalNoise']`

In [9]: `system_info()`

System information

- OS: Windows
- Python: 3.9.5
- PyFMI: 2.9.5
- FMU by: JModelica.org
- FMI: 2.0
- Type: FMUModelCS2
- Name: BPL_TEST2.BatchWithNoise
- Generated: 2022-09-04T14:31:31
- MSL: 3.2.2 build 3
- Description: Bioprocess Library version 2.1.0 beta
- Interaction: FMU-explore ver 0.9.3

In []:

