BPL_YEAST_AIR_Fedbatch script with FMPy

The key library FMPy is installed.

After the installation a small application BPL_YEAST_AIR_Fedbatch 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 22.04.4 LTS
Release:        22.04
Codename:       jammy
```

```
!python --version
```

```
Python 3.11.11
```

```
!pip install fmpy
```

```
Collecting fmpy
  Downloading FMPy-0.3.22-py3-none-any.whl.metadata (1.9 kB)
Requirement already satisfied: attrs in /usr/local/lib/python3.11/dist-packages (from fmpy) (25.3.0)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.11/dist-packages (from fmpy) (3.1.6)
Collecting lark (from fmpy)
  Downloading lark-1.2.2-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: lxml in /usr/local/lib/python3.11/dist-packages (from fmpy) (5.3.1)
Requirement already satisfied: msgpack in /usr/local/lib/python3.11/dist-packages (from fmpy) (1.1.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from fmpy) (2.0.2)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from Jinja2->fmpy) (
Downloading FMPy-0.3.22-py3-none-any.whl (4.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.9/4.9 MB 15.4 MB/s eta 0:00:00
Downloading lark-1.2.2-py3-none-any.whl (111 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 111.0/111.0 kB 7.1 MB/s eta 0:00:00
Installing collected packages: lark, fmpy
Successfully installed fmpy-0.3.22 lark-1.2.2
```

Notes of BPL_YEAST_AIR_Fedbatch

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

- FMU - BPL_YEAST_AIR_Fedbatch_linux_jm_cs.fmu
- Setup-file - BPL_YEAST_AIR_Fedbatch_fmpy_explore

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

```
Cloning into 'BPL_YEAST_AIR_Fedbatch'...
```

```
%cd BPL_YEAST_AIR_Fedbatch
```

```
/content/BPL_YEAST_AIR_Fedbatch
```

BPL_YEAST_AIR_Fedbatch - demo

Author: Jan Peter Axelsson

This notebook demonstrate yeast fedbatch cultivation. We look at impact of changes in the glucose feeding. We also take a look at tuning of the DO-control system. Both liquid- and gasphase are included in the model. The culture growth and metabolism are formulated in relation to to the cells respiratory capacity [1] as well as oxygen limitation in the reactor. The model is exapanded to describe also the gas phase as well as the culture heat production [2]. The model was derived mainly from continuous culture data but proved to capture dynamic aspects well of ethanol production and consumption [3]. Several cultivations where donw with ethnol control that facilatated high cell density cultivations [4].

Interaction with the compiled model as FMU is mainly through the simplified commands: par(), init(), newplot(), simu() etc. The last simulation is always available in the workspace and called 'sim_res'. The command describe() brings mainly up description infomration from the actual Modelica code from the FMU but is complemented with information given in the dedicated Python setup-file.

The idea is to demonstrate how simulations and varyiing conditions can provide some process insight that can support the experimetnal work. I hope that at the end of this session you are ready to formulate your own questions you want to address with simulations - and you can just go on in this notebook! Just press the field "+Code" in the upper left part of notebook interface and you get a new "cell" where you write your own code. You can copy and paste from cells above using ctrol-c and ctrl-p as usual and edit the cell. When your are ready to execute the cell just press the "play button" to the left in the cell or press shift-enter as in "ordinary" Jupyter notebooks.

After a session you may want to save your own notebook. That you can do on your Google Drive account and I refer to Colab instructions for how to do this. It is easy.

Enjoy!

```
run -i BPL_YEAST_AIR_Fedbatch_DOcontrol_fmpy_explore.py
```

```
Linux - run FMU pre-comiled 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()
```

```
%matplotlib inline
plt.rcParams['figure.figsize'] = [36/2.54, 30/2.54]
```

About the process model

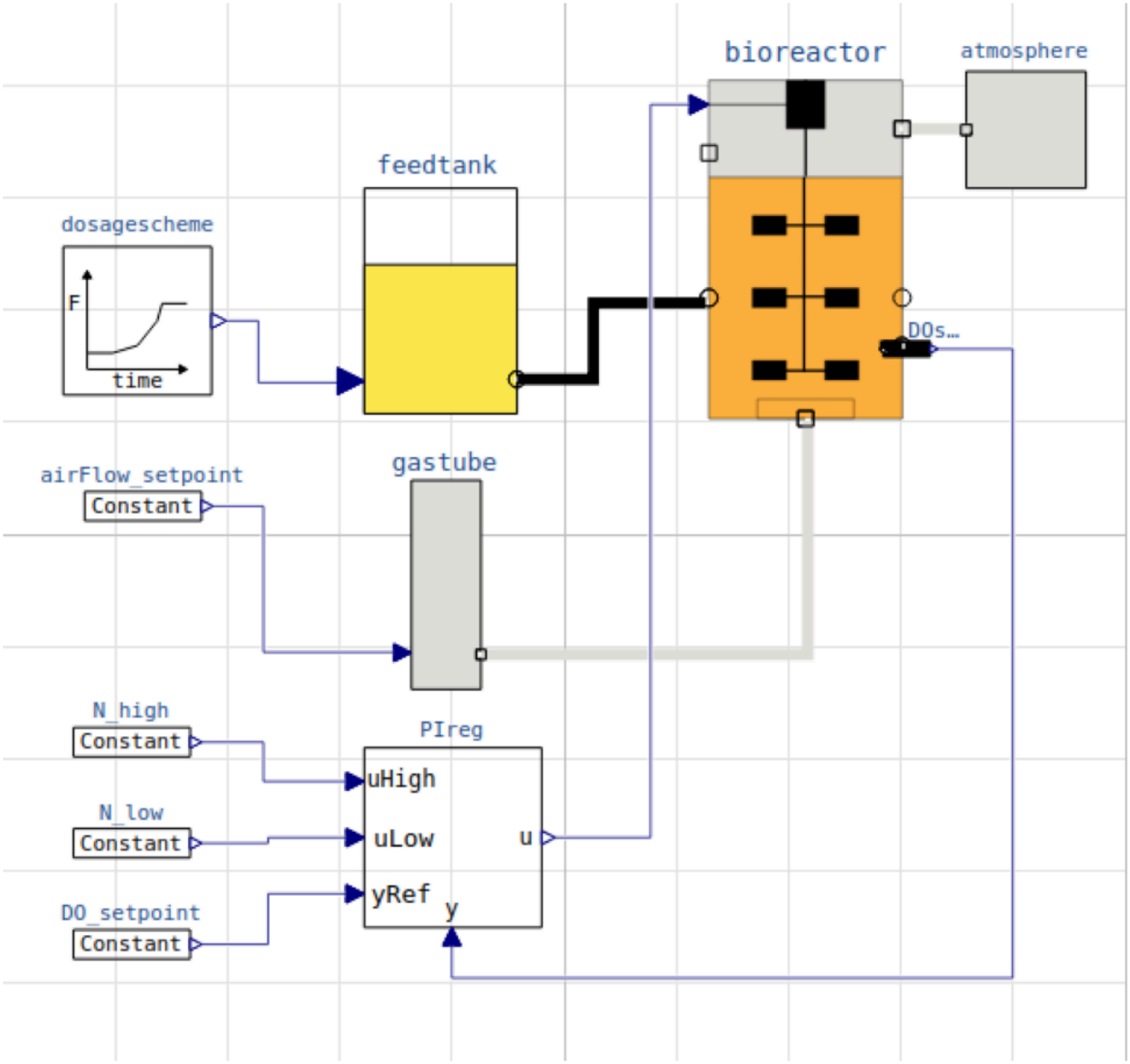
We can get information about the process, liquid- and gas-phase by the command describe(). This command can also be used to bring up information about a specific variable or parameter. However, you should use describe() after a simulation to get the valued used during the simulation.

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

```
Saccharomyces cerevisae - default parameters for strain H1022
```

```
process_diagram()
```

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



The model of the process has parameters both for culture, gas_liquid_transfer, as well as feeding procedure. The paramters that are available for changes you find by the command disp() and you get a long list and you change by them by command par(). The model has even more parameters in the background but not made available for interaction.

First simulations - adjusting start of substrate feeding

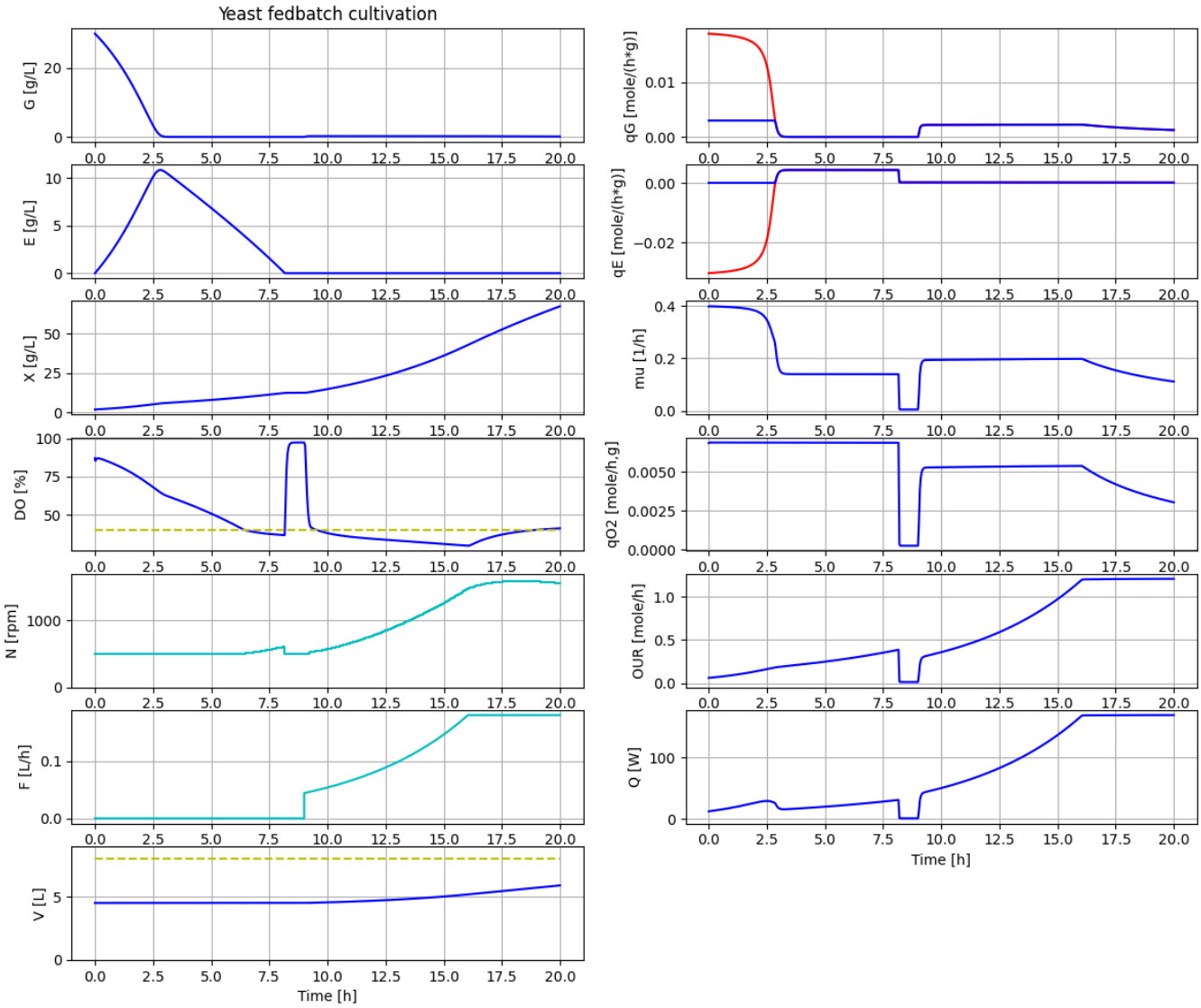
```
# Culture parameters and others at default values
par(qO2max=0.0069)

# Process initial conditions
init(V_start=4.5, VG_start=4.5*30, VX_start=4.5*2, VE_start=4.5*0)

# Feed profile
par(t_startExp=9, F_startExp=0.044, mu_feed=0.20, F_max=0.18)

# DO-control parameters
par(K=10, Ti=0.5)

# Simulate and plot
newplot(title='Yeast fedbatch cultivation', plotType='Overview')
simu(20)
```



Now we can get value of broth volume as well as the headspace and values are the last ones in the simulation

```
describe('bioreactor.V')
```

```
Reactor broth volume : 5.891 [ L ]
```

```
describe('bioreactor.V_gasphase')
```

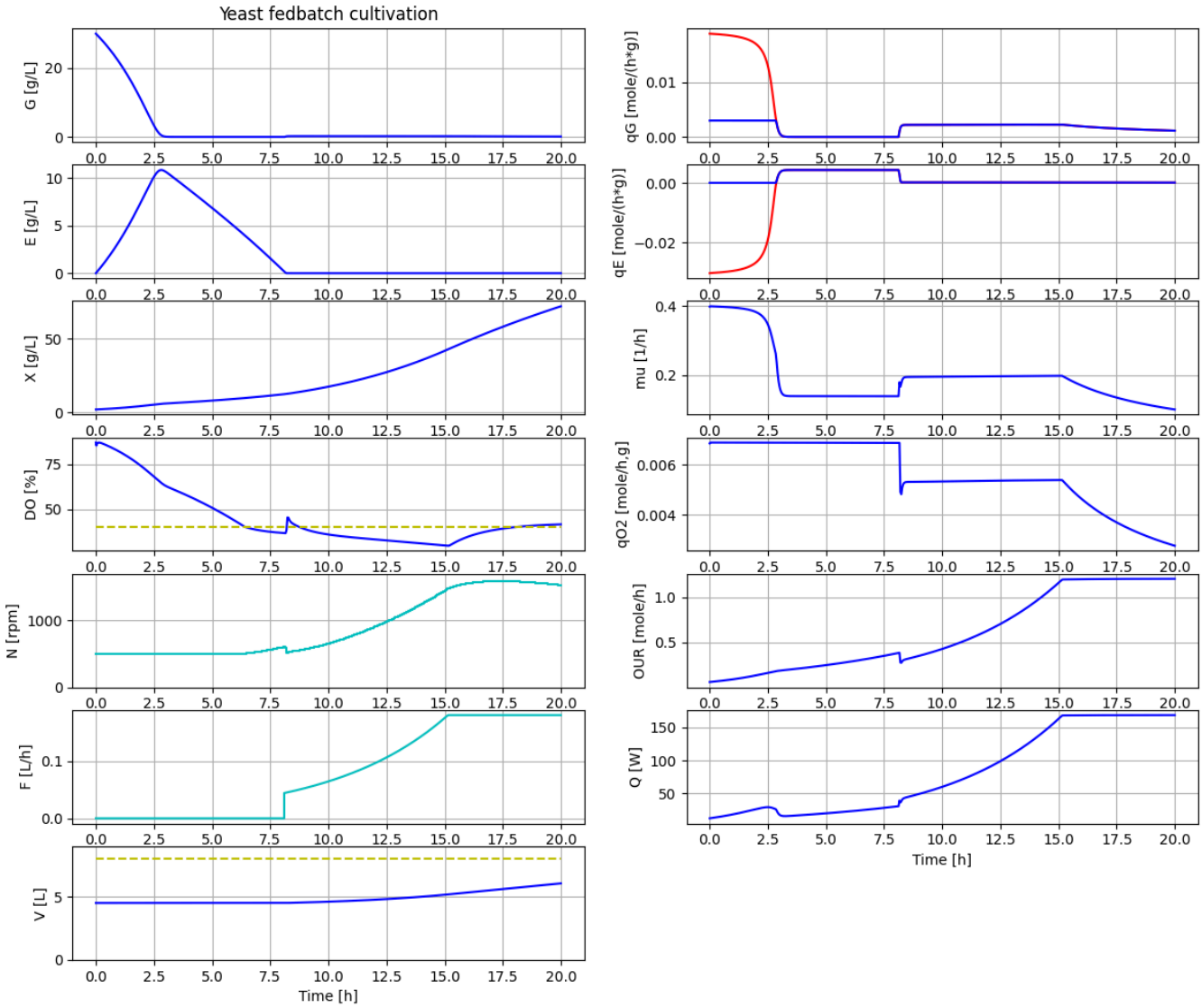
```
Volume of the gas phase : 2.109 [ L ]
```

```
# Take a look at the parameters available to adjust the dosage scheme
disp('dosage', decimals=4)
```

```
F_start : 0.0
mu_feed : 0.2
t_startExp : 2.0
F_startExp : 0.12
F_max : 3.0
```

```
# Let us start the feeding just after the batch phase has ended and keep other parameters the same
par(t_startExp=8.1)

# Simulate and plot
newplot(title='Yeast fedbatch cultivation', plotType='Overview')
simu(20)
```



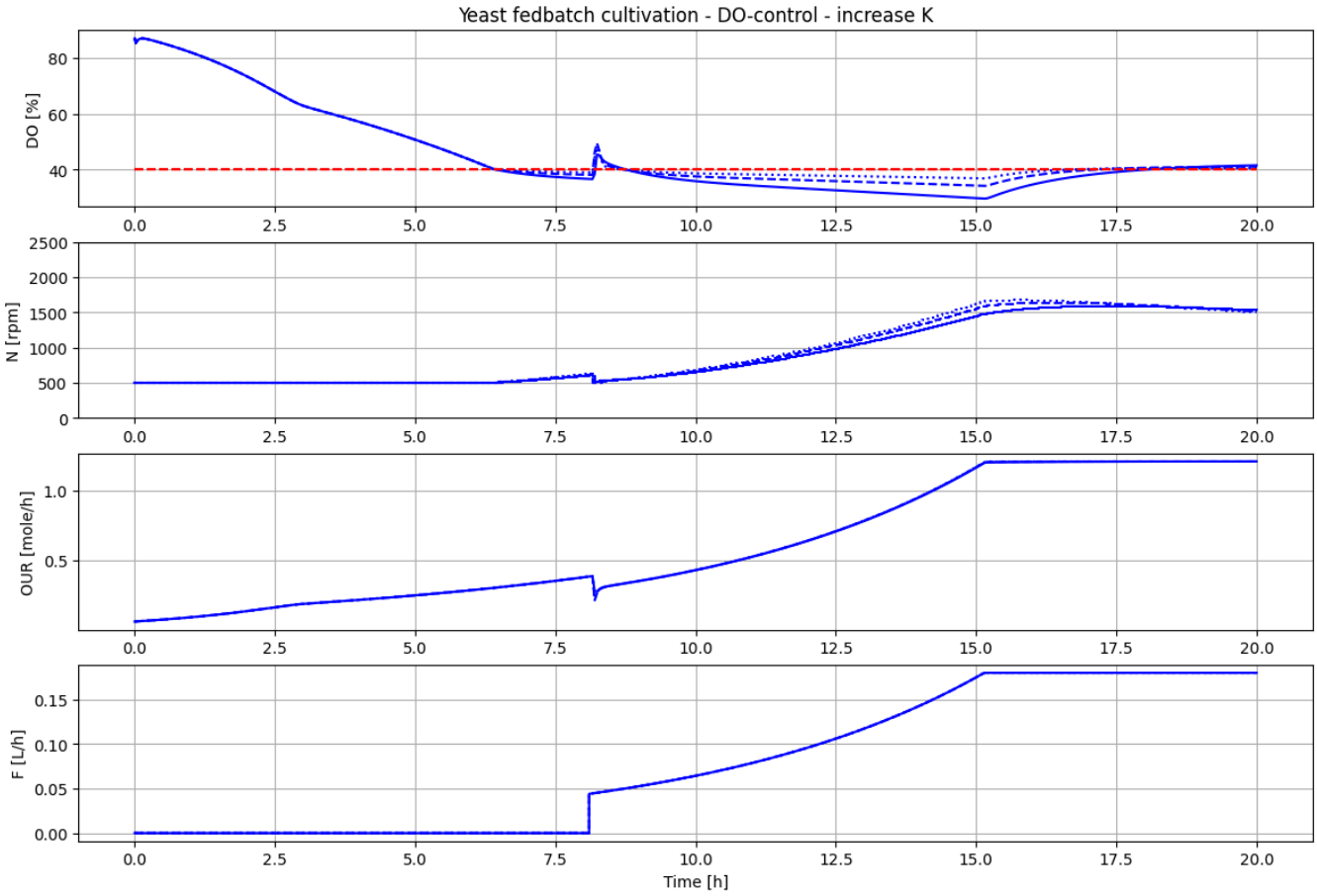
The increase of DO to about 50 % at end of batch phase should be possible to detect easily. This simulation is more realistic and we use these settings from now on.

DO-control - tuning of PI-regulator parameters

Let us focus on the DO-control system and choose a more limited plotType. We study the impact of PI control parameters and see if we can decrease the control error without looing stability.

```
# Let us take a closer look at the DO-control system and try to make control error smaller by increaseing K
newplot(title='Yeast fedbatch cultivation - DO-control - increase K', plotType='Focus DO-control')
for value in [10, 20, 40]: par(K=value); simu(20)

# Reset K to the original value
par(K=10)
```



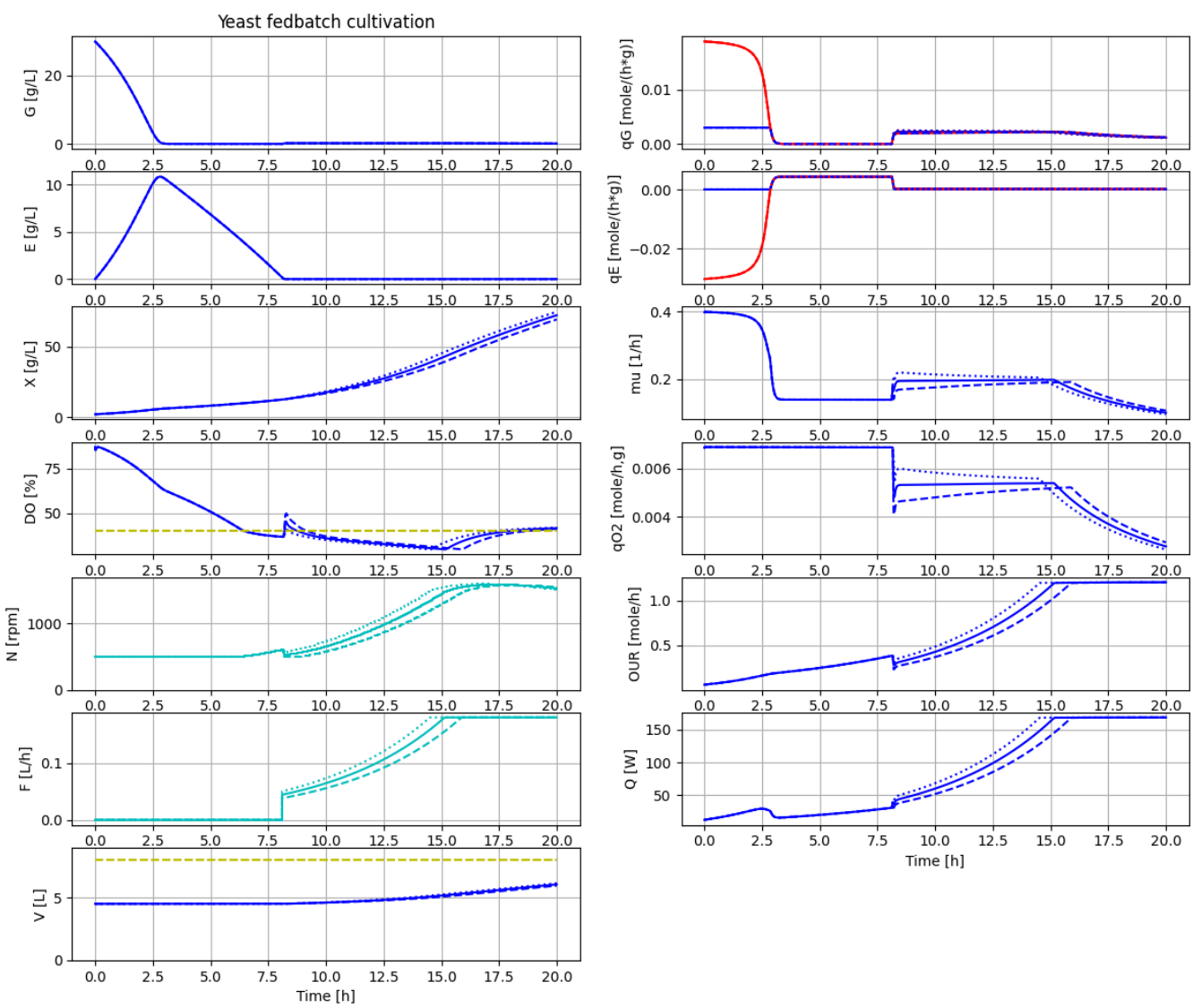
We see that by a higher control gain K the DO-control error get smaller and the stability of the control system is maintained.

Exercise I leave for you to study the impact variation of the Ti-parameter. Just make a new cell below. Then copy and paste the cell above and change parameter to Ti.

Sensitivity to changes in feed-profile

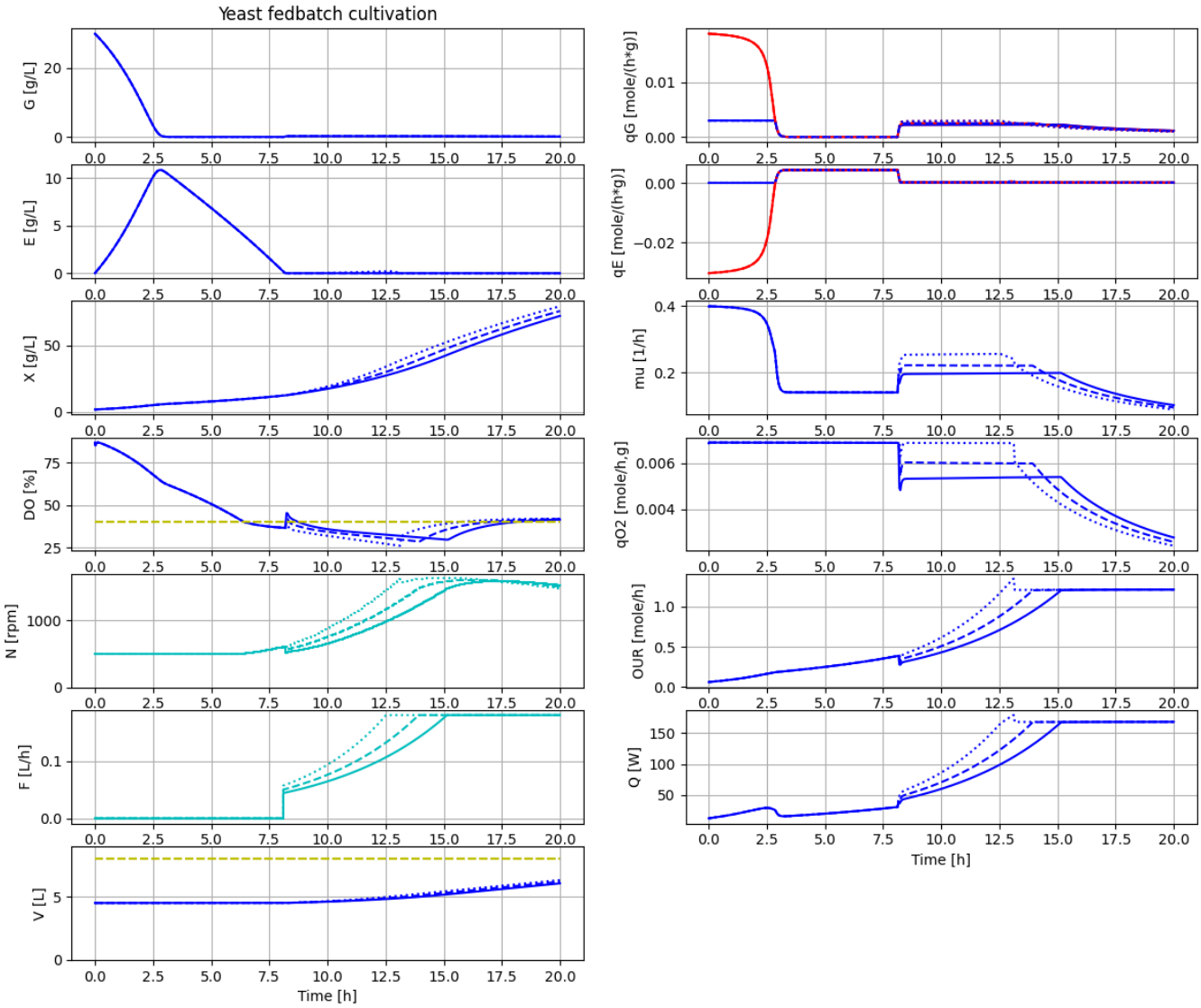
Now, let us focus on investigating impact of changes in the feed-profile. The goal is to increase the produced cell mass without accumulation of by-product ethanol. Simulation can bring some insight into how behaviour of the differen variables change when by-product is formed. This insight can help to interpret experimental results.

```
# Let us check the sensitivity to changes in the feed profile design
newplot(title='Yeast fedbatch cultivation', plotType='Overview')
for value in [0.044, 0.038, 0.050]: par(F_startExp=value); simu(20)
```

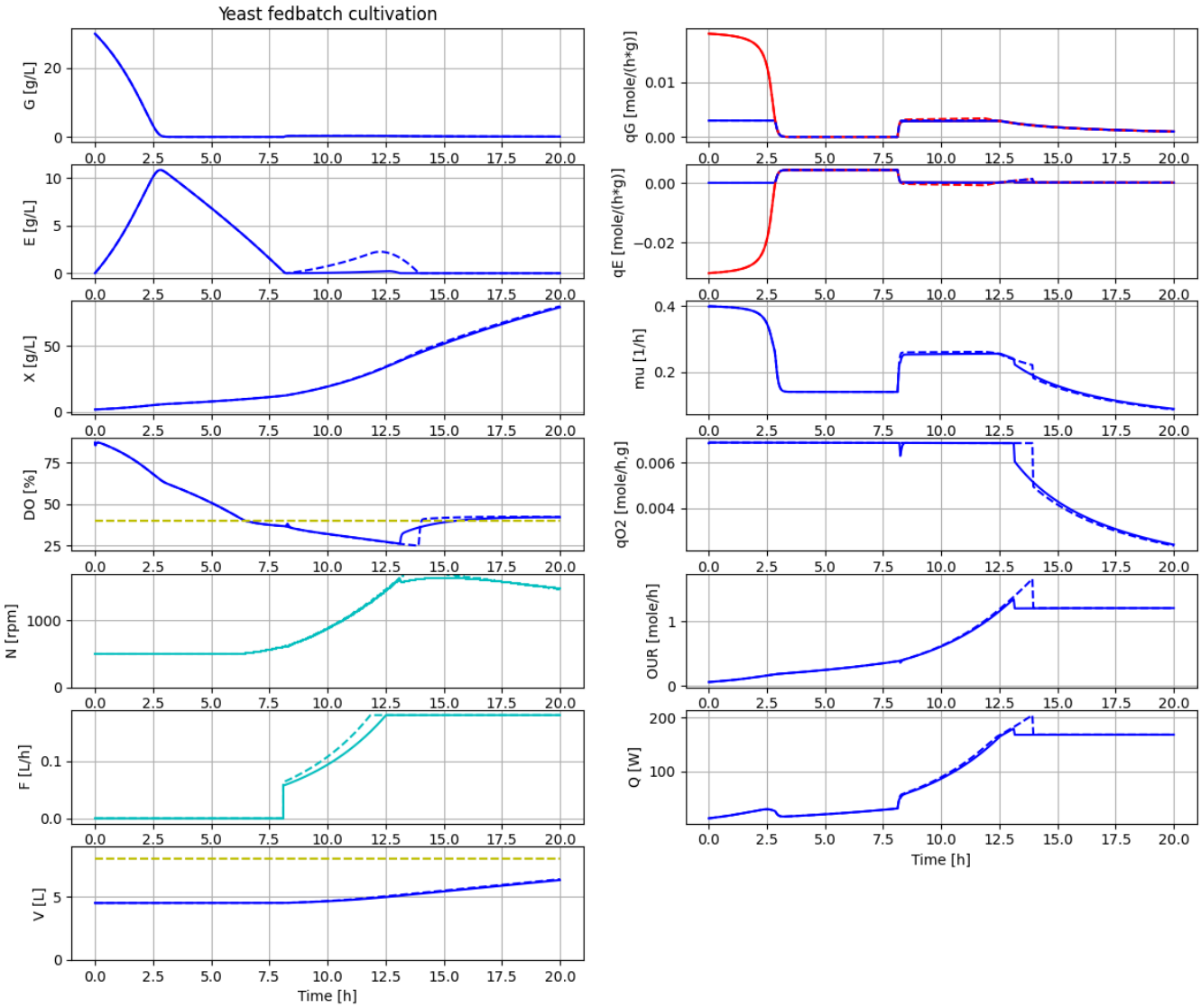


The variation in F_start has an impact and we see that the actual growth rate during fedbatch phase do converge to the set growth rate of the feed, but i takes more than 5 hours.

```
# Let us investigate a feedprofile that is closer to the maximal capacity
newplot(title='Yeast fedbatch cultivation', plotType='Overview')
par(F_startExp=0.044, mu_feed=0.20); simu(20)
par(F_startExp=0.050, mu_feed=0.22); simu(20)
par(F_startExp=0.057, mu_feed=0.26); simu(20)
```




```
# And let us see what happens if the feedprofile exceed the culture capacity
newplot(title='Yeast fedbatch cultivation', plotType='Overview')
par(F_startExp=0.057, mu_feed=0.26); simu(20)
par(F_startExp=0.063, mu_feed=0.28); simu(20)
par(F_startExp=0.044, mu_feed=0.20)
```



Note that with the feedprofile that exceed culture respiratory capacity, ethanol is accumulated during time 8-12.5 hours. When the feedprofile then is constant from time 12.5 hours and on, then the accumulated ethanol is consumed over about an hour. This leads to a higher oxygen demand and heat production during this time. The specific cell growth rate is also slightly higher during this period.

Exercise You can investiate the impact of changing the maximal feedrate F_max. Make sure that the DO level do not get too low.

Sensitivity to low dissolved oxygen levels

The detailed understanding of process dynamics at very low dissolved oxygen levels is not that well studied, to the authors knowledge. Here is a section on "Growth under oxygen limiation" in the original paper [1]. This model we use here. The idea is that the oxygen uptake is essentially described with a Monod-function with a parameter here denoted KsO2. At higher dissolved oxygen levels the uptake is saturated by the respiratory capacity and in our model denoted qO2max. Thus, at lower dissolved oxygen levels the respiration capacity is lowered and denoted here qO2lim and at sufficiently high dissolved oygen levels the qO2lim corresponds to qO2max. The value of qO2lim controls metabolism and growth.

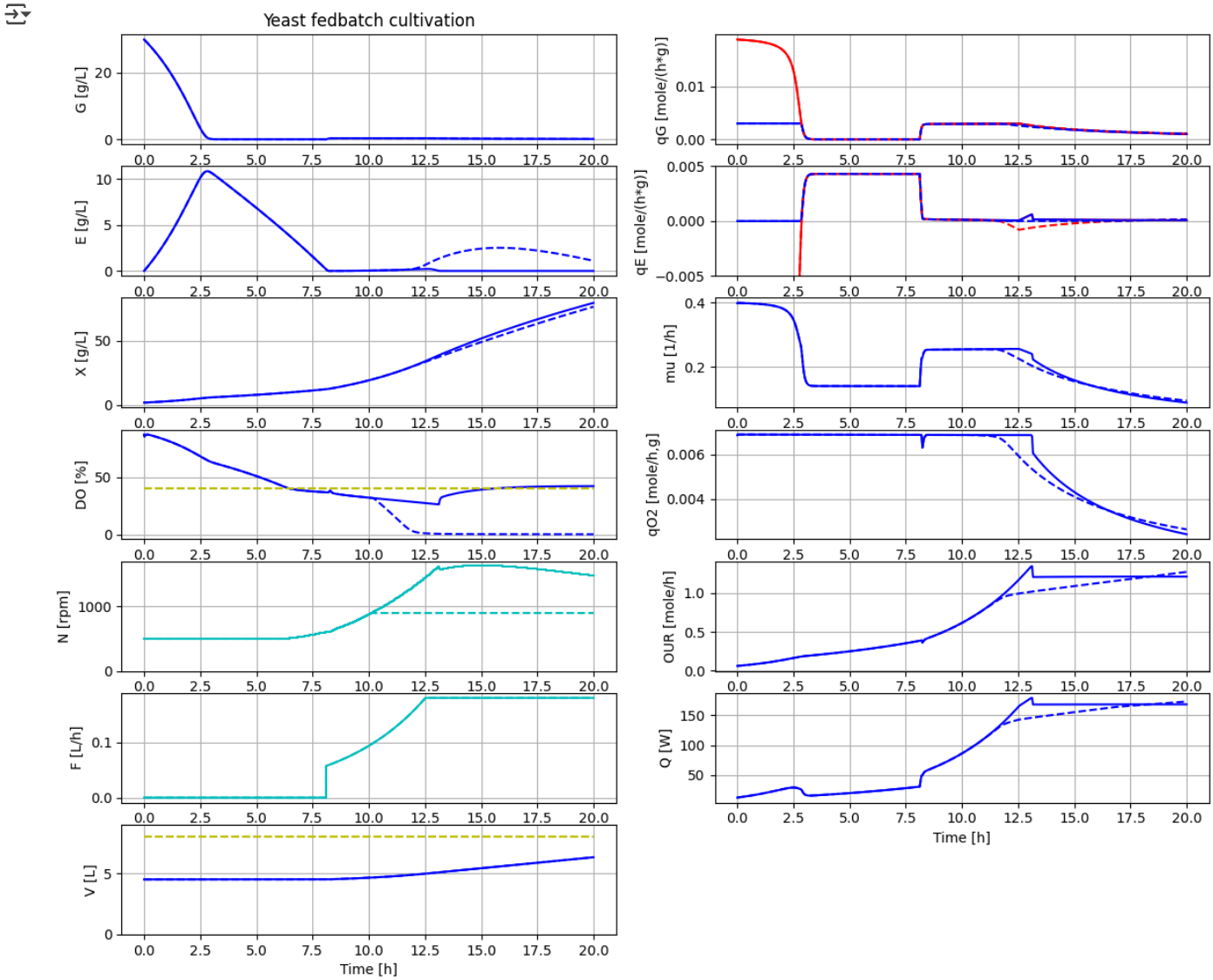
$$qO2_{lim} = qO2^{max} \frac{c[O2]}{KsO2 + c[O2]}$$

The process dyanmics for variation of glucose feed rate around the respiratory capacity under well-aerated condtions are well described in [3]. It would be interesting to make similar studies at low dissolved oxygens levels. Also interesting to describe the impact of variation in the aeration during condition of a constant glucose supply to see the impact of ethanol production and consumption this way.

During the time 1980-90 there was an interest to study the impact variation in dissolved oxygen in two-reactor setups [5]. In the smaller reactor dissolved oxygen level was low while higher in the larger reactor and the culture was circulated in the system at a rate related to

typical mixing times in a large reactor. This experimental setup has been simulated with Bioprocess Library with focus on substrate gradients rather than oxygen gradients [6].

```
# Let us instead see what happens when the aeration read a limit in terms of stirrer speed.
newplot(title='Yeast fedbatch cultivation', plotType='Overview'); ax22.set_ylim([-0.005,0.005])
par(F_startExp=0.057, mu_feed=0.26); simu(20)
par(F_startExp=0.057, mu_feed=0.26, N_high=900); simu(20)
par(F_startExp=0.044, mu_feed=0.20, N_high=2000)
```



We see that after 10 hours the maximal stirrer speed is reached and that limits oxygen transfer and dissolved oxygen goes down. At about 12.5 hours the dissolved oxygen has come down to low levels that limits the specfic oxygen uptake rate and lower than the specific respiraotry capacity and we get ethanol production.