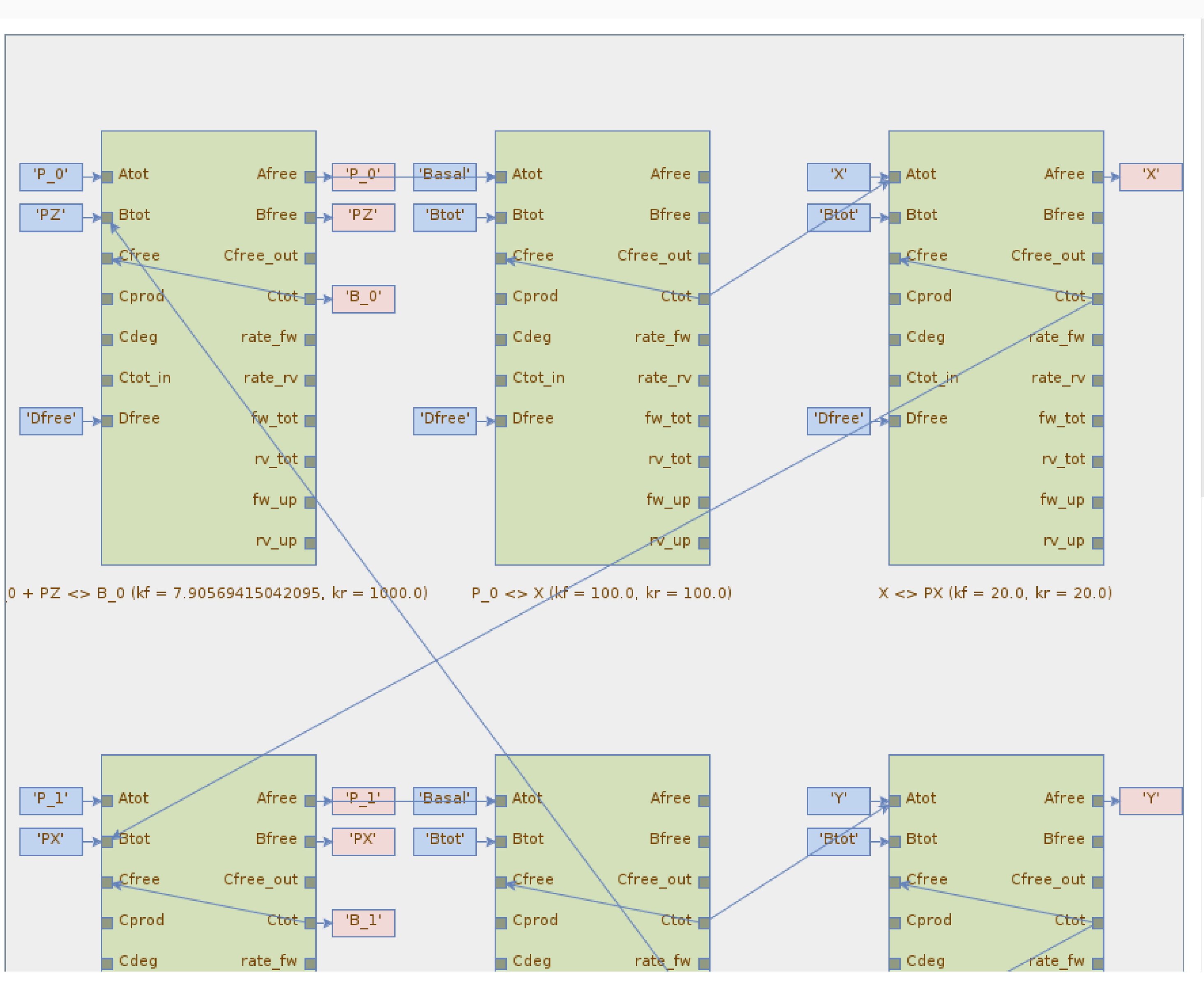
```
[2]
import mtt
from os.path import join, realpath
from os import getcwd
sbfile = join(realpath(getcwd()),'three_steps.sb')
wiring = mtt.Wiring.fromFile2(sbfile, default_margin=1000., input_val=10.,
current_assignment='maxKD')
wiring
```

```
Block P_0 + PZ <> B_0 (kf = 7.90569415042095, kr = 1000.0)
  Afree -> 'P_0 <> X (kf = 100.0, kr = 100.0)'.Atot
  Ctot -> Cfree
Block P_0 <> X (kf = 100.0, kr = 100.0)
  Ctot -> 'X <> PX (kf = 20.0, kr = 20.0)'.Atot
  Ctot -> Cfree
Block X <> PX (kf = 20.0, kr = 20.0)
  Ctot -> 'P_1 + PX <> B_1 (kf = 7.90569415042095, kr = 1000.0)'.Btot
  Ctot -> Cfree
Block P_1 + PX <> B_1 (kf = 7.90569415042095, kr = 1000.0)
  Afree -> 'P_1 <> Y (kf = 100.0, kr = 100.0)'.Atot
  Ctot -> Cfree
Block P_1 <> Y (kf = 100.0, kr = 100.0)
  Ctot -> 'Y <> PY (kf = 20.0, kr = 20.0)'.Atot
  Ctot -> Cfree
Block Y <> PY (kf = 20.0, kr = 20.0)
  Ctot -> 'P_2 + PY <> B_2 (kf = 7.90569415042095, kr = 1000.0)'.Btot
  Ctot -> Cfree
Block P_2 + PY <> B_2 (kf = 7.90569415042095, kr = 1000.0)
  Afree -> 'P_2 <> Z (kf = 100.0, kr = 100.0)'.Atot
  Ctot -> Cfree
Block P_2 <> Z (kf = 100.0, kr = 100.0)
  Ctot -> 'Z <> PZ (kf = 20.0, kr = 20.0)'.Atot
  Ctot -> Cfree
Block Z <> PZ (kf = 20.0, kr = 20.0)
  Ctot -> 'P_0 + PZ <> B_0 (kf = 7.90569415042095, kr = 1000.0)'.Btot
```

```
[3]
model = mtt.MTT(wiring)
model.draw(3)
```

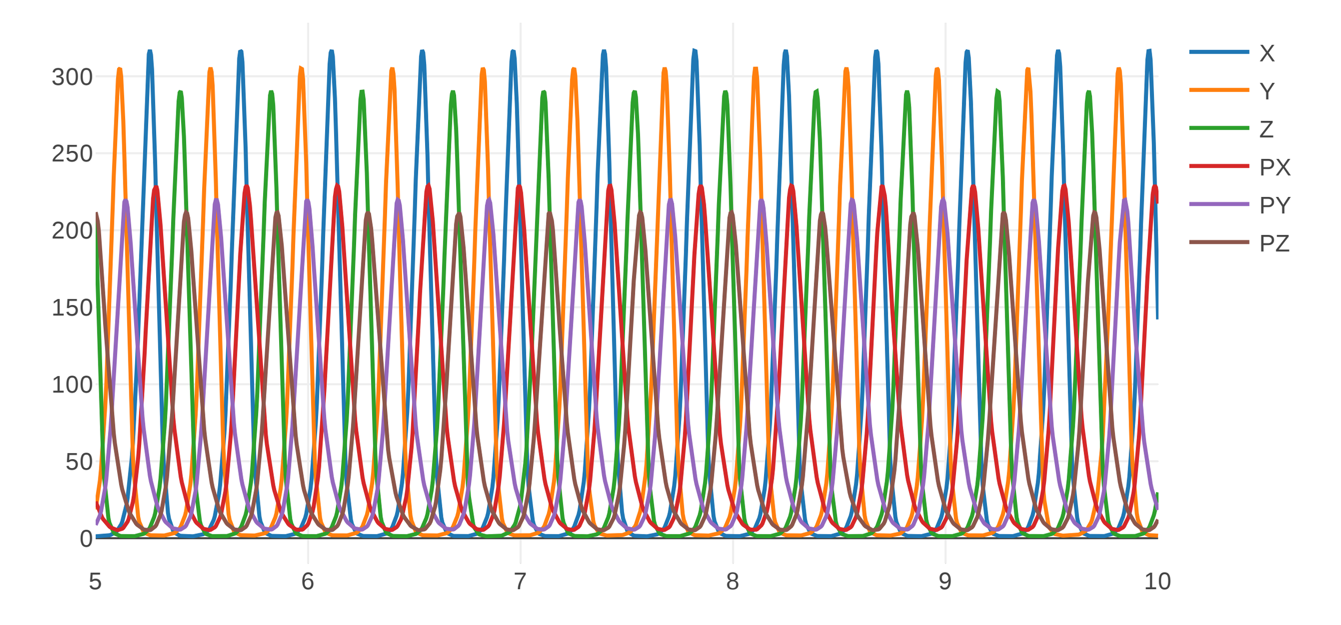


Digitized current config

```
[4]
# for block in wiring.blocks:
#     print(block.getDigitizedParameterString(model))
```

Simulation

```
[5]
sbml = model.toSBML()
# print(sbml)
from roadrunner import RoadRunner
with open('out-sbml.xml', 'w') as f:
    f.write(sbml)
xmod = RoadRunner(sbml)
xmod.reset()
# simulate over transient
xmod.simulate(0,5,1000,selections=['time']+['X','Y','Z','PX','PY','PZ'])
xmod.simulate(5,10,1000,selections=['time']+['X','Y','Z','PX','PY','PZ'])
xmod.plot()
```

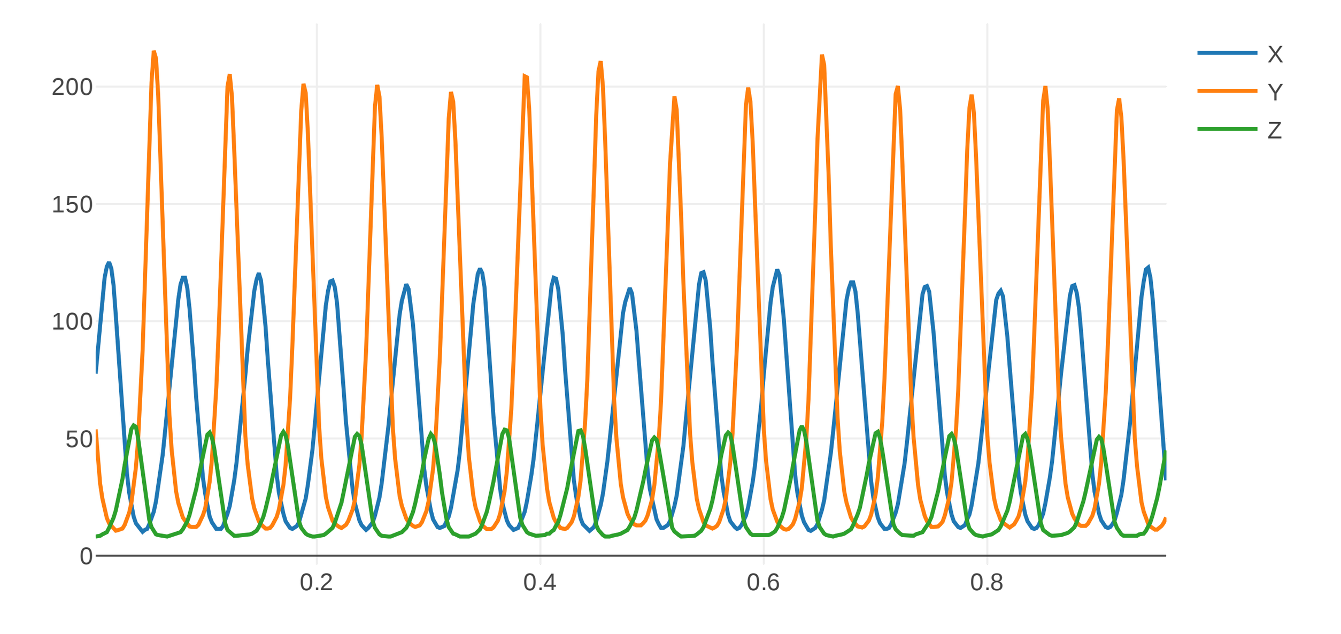



```
[6]
from scipy.io import loadmat
from tellurium import plot, show
from numpy import vstack, correlate, amax, argmax, zeros, linspace
```

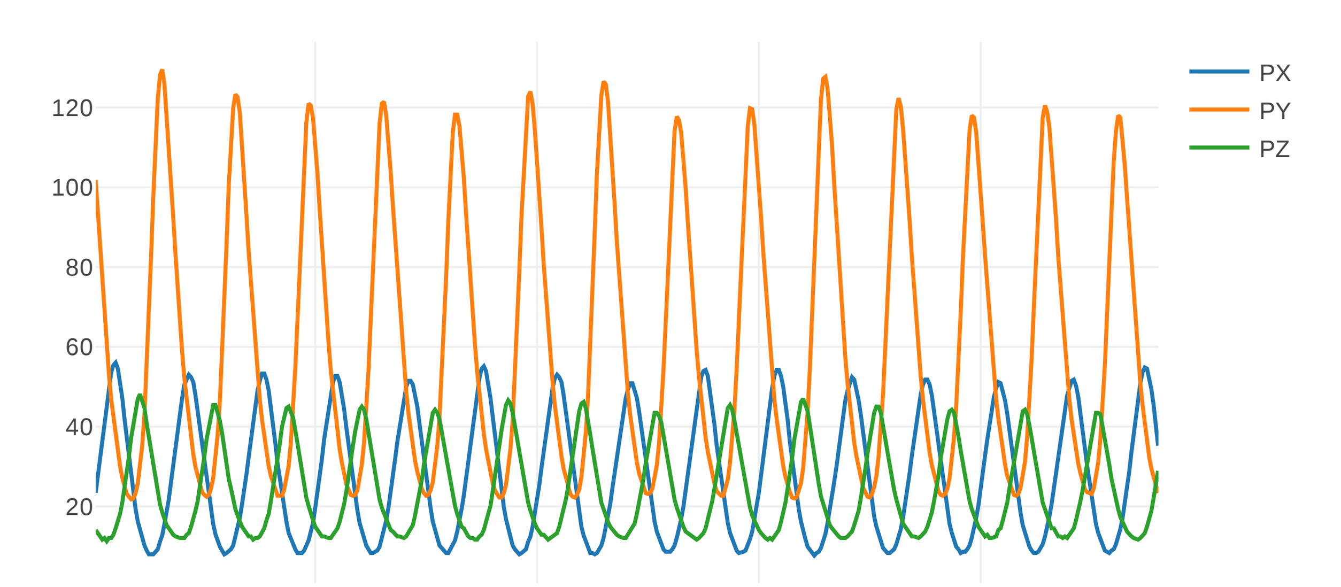
```
[7]
rna_data = loadmat('data/rna.mat')
rna1X = rna_data['rna1X'].flatten()
rna1Y = rna_data['rna1Y'].flatten()
rna2X = rna_data['rna2X'].flatten()
rna2Y = rna_data['rna2Y'].flatten()
rna3X = rna_data['rna3X'].flatten()
rna3Y = rna_data['rna3Y'].flatten()
rna_x = vstack((rna1X,rna2X,rna3X)).T
chip_duration = float(rna_x[-1,0] - rna_x[0,0])
rna_y = vstack((rna1Y,rna2Y,rna3Y)).T
```

```
[8]
protein_data = loadmat('data/protein.mat')
Protein1X = protein_data['Protein1X'].flatten()
Protein1Y = protein_data['Protein1Y'].flatten()
Protein2X = protein_data['Protein2X'].flatten()
Protein2Y = protein_data['Protein2Y'].flatten()
Protein3X = protein_data['Protein3X'].flatten()
Protein3Y = protein_data['Protein3Y'].flatten()
protein_x = vstack((Protein1X,Protein2X,Protein3X)).T
protein_y = vstack((Protein1Y,Protein2Y,Protein3Y)).T
```

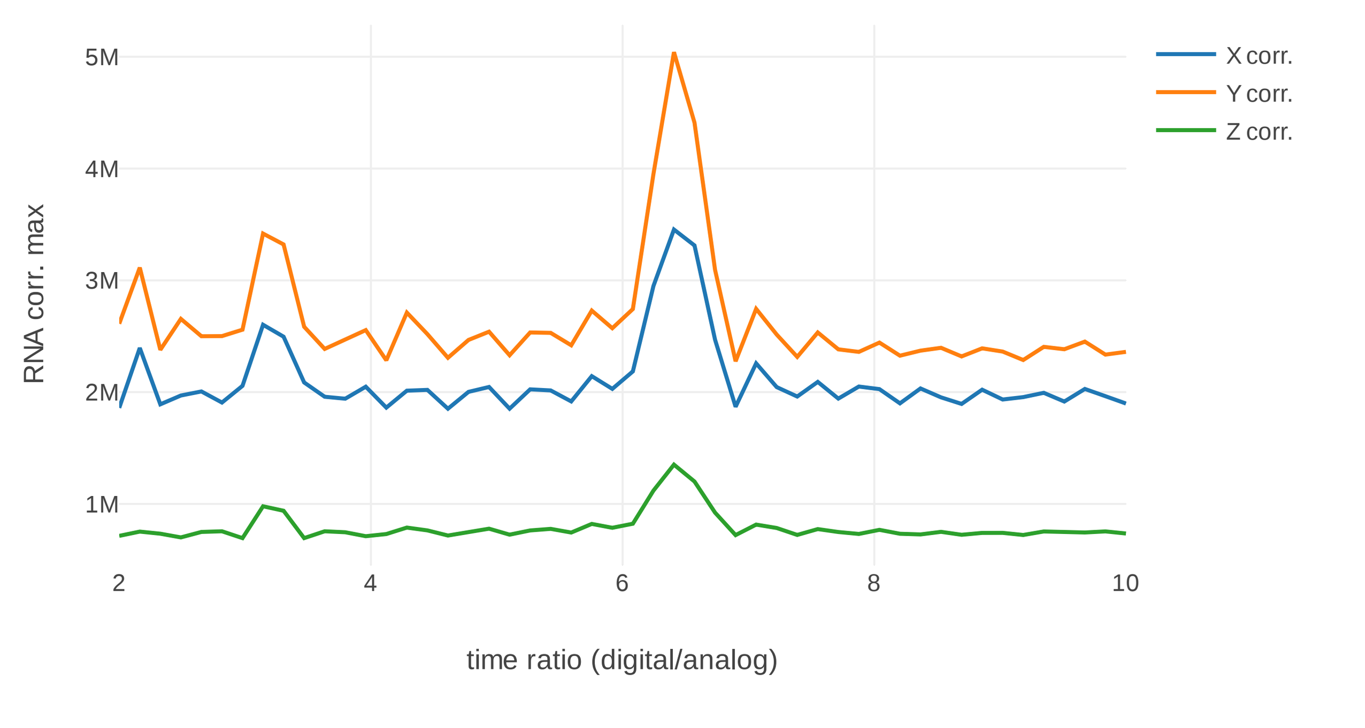
```
[9]
plot(rna1X, rna1Y, name='X', show=False)
plot(rna2X, rna2Y, name='Y', show=False)
plot(rna3X, rna3Y, name='Z', show=False)
show()
```



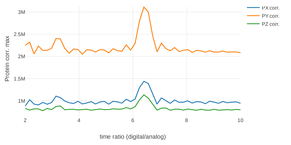

```
[10]
plot(Protein1X, Protein1Y, name='PX', show=False)
plot(Protein2X, Protein2Y, name='PY', show=False)
plot(Protein3X, Protein3Y, name='PZ', show=False)
show()
```




```
[19]
N = 50

max_rna_cor = zeros((3,N))
max_rna_index = zeros((3,N))
max_rna_time = zeros((3,N))

max_protein_cor = zeros((3,N))
max_protein_index = zeros((3,N))
max_protein_time = zeros((3,N))

ratio_range = linspace(2., 10., N)
for n,t_ratio in enumerate(ratio_range):
    sim_duration = t_ratio*chip_duration
    xmod.reset()
    # simulate over transient
    xmod.simulate(0,5,1000,selections=['time']+['X','Y','Z','PX','PY','PZ'])
    r = xmod.simulate(5.,5.+sim_duration,480,selections=['time']+
['X','Y','Z','PX','PY','PZ'])

    sim_rna = r[:,1:4]
    sim_protein = r[:,4:7]

    for k,name in enumerate(('X corr.','Y corr.','Z corr.')):
        rna_corr = correlate(sim_rna[:,k],rna_y[:,k],mode='same')
        max_rna_cor[k,n] = amax(rna_corr)
        max_rna_index[k,n] = argmax(rna_corr)
        max_rna_time[k,n] = r[int(max_rna_index[k,n]),0]
#           plot(r[:,0], rna_corr, name=name, xtitle = 'time (s)', ytitle='RNA corr. max =
'+str(max_rna_time[0]), show=False)
#       show()
    for k,name in enumerate(('PX corr.','PY corr.','PZ corr.')):
        protein_corr = correlate(sim_protein[:,k],protein_y[:,k],mode='same')
        max_protein_cor[k,n] = amax(protein_corr)
        max_protein_index[k,n] = argmax(protein_corr)
        max_protein_time[k,n] = r[int(max_protein_index[k,n]),0]
#           plot(r[:,0], protein_corr, name=name, xtitle = 'time (s)', ytitle='Protein corr.
max t = '+str(max_protein_time[0]), show=False)
#       show()

for k,name in enumerate(('X corr.','Y corr.','Z corr.')):
    plot(ratio_range, max_rna_cor[k,:], name=name, xtitle = 'time ratio (digital/analog)',
ytitle='RNA corr. max', show=False)
show()
```




```
[20]
for k,name in enumerate(('PX corr.','PY corr.','PZ corr.')):
    plot(ratio_range, max_protein_cor[k,:], name=name, xtitle = 'time ratio
(digital/analog)', ytitle='Protein corr. max', show=False)
show()
```


PX corr.
PY corr.
PZ corr.
3M
2.5M
2M
1.5M
1M
Photon corr. max
2
4
6
8
10
time ratio (digital/analog)