In [1]:
```
import Pkg; Pkg.add("DifferentialEquations")
import Pkg; Pkg.add("Plots")
import Pkg; Pkg.add("Formatting")
import Pkg; Pkg.add("Catalyst")
import Pkg; Pkg.add("Noise")
import Pkg; Pkg.add("LinearAlgebra")
import Pkg; Pkg.add("Latexify")
import Pkg; Pkg.add("ModelingToolkit")
import Pkg; Pkg.add("PlotlyJS")
import Pkg; Pkg.add("MATLABDiffEq")

_ = IJulia.clear_output(true)
```

Out[1]:
```
0
```

In [2]:
```
using DifferentialEquations;
using Random;
using Plots;
using Formatting;
using LinearAlgebra;
using Noise;
using Catalyst;
using Latexify;
using MATLABDiffEq;
```

Globals

In [3]:
```
ALPHA = 1e-3
BETA = 1e-8

BASE_RATE = 1.0
BASE_CONC = 1.0

INFRATESCALE = 1e3
INFCONCSCALE = 1e4 # In 10nM order or something like that.

UNIMOL_K = BASE_RATE*ALPHA
BIMOL_K = 2*BASE_RATE*ALPHA/BETA

INFRATE = INFRATESCALE*BASE_RATE*ALPHA/BETA
INFCONC = INFCONCSCALE*BASE_CONC*BETA
LINKER_RATE=INFRATESCALE/INFCONCSCALE*BASE_RATE*ALPHA/BETA

NCAT = 3 # No. of catalytic reactions per each autocatalytic reaction

LEAK = 1
SHADOW = 0
LEAK_RATE = 1e-5
```

Out[3]:
```
1.0e-5
```

Utils

In [4]:

```
# Returns the sum of concentrations of the species with a given prefix
function aggregate_values(rn, odesol; prefix='x')
    ss = species(rn)

    ret = zeros(length(odesol[1, :]))
    total = 0
    for i in 1:length(ss)
        item = ss[i]
        item_str = "$item"
        if startswith(item_str, prefix)
#             println(item_str)
            ret += odesol[i, :] # Gets the variable at index i for all ti
        end
    end
    return ret
end


function print_species_array(reaction_network)
    ss = species(reaction_network)
    print("species = [")
    for s in ss
        print("'", s,"'", ",")
    end
    print("]")

end


function simulate(rn, u0, p; tspan=(0.0, 1000.0), reltol=1e-12, abstol=1e
    oprob = ODEProblem(rn, u0, tspan, p)
    sol = solve(oprob, Rosenbrock23(), reltol=reltol, abstol=abstol)
    return sol
end
```

Out[4]:

```
simulate (generic function with 1 method)
```

In [5]:
```
function simulation_results(rn,
                            u0,
                            p;
                            tspan=(0, 2.0),
                            track_prefix="x")
    sol = simulate(rn, u0, p, tspan=tspan)
    xs = sol.t
    ys = aggregate_values(rn, sol, prefix=track_prefix)
    return xs, ys
end

# TODO(rajiv): Try to fit all the ideal, cat, dna curves into the same pl
function dna_simulation_results(dna_rn,
                dna_u0,
                dna_p;
                tspan=(0, 6000),
                leak_index=4,
                shadow_index=6,
                track_prefixes=["x"],
                abstol=1e-9,
                reltol=1e-9)

    dna_p_leak = deepcopy(dna_p)
    dna_p_leak[leak_index] = 1

    dna_p_leak_shadow = deepcopy(dna_p)
    dna_p_leak_shadow[leak_index] = 1
    dna_p_leak_shadow[shadow_index] = 1

    dna_sol = simulate(dna_rn, dna_u0, dna_p, tspan=tspan)
    dna_leak_sol = simulate(dna_rn, dna_u0, dna_p_leak, tspan=tspan)
    dna_leak_shadow_sol = simulate(dna_rn, dna_u0, dna_p_leak_shadow, tsp
    xs = []
    ys = []
    labels = []
    for track_prefix in track_prefixes

        dna_output = aggregate_values(dna_rn, dna_sol, prefix=track_prefi
        dna_leak_output = aggregate_values(dna_rn, dna_leak_sol, prefix=t
        dna_leak_shadow_output = aggregate_values(dna_rn, dna_leak_shadow
        append!(xs, [dna_sol.t, dna_leak_sol.t, dna_leak_shadow_sol.t])
        append!(ys, [dna_output, dna_leak_output, dna_leak_shadow_output]
        append!(labels, ["$track_prefix leak:0|shadow:0" "$track_prefix l
    end
    return xs, ys, labels
end
```

Out[5]:
```
dna_simulation_results (generic function with 1 method)
```

1. Rock Paper Scissor Oscillator

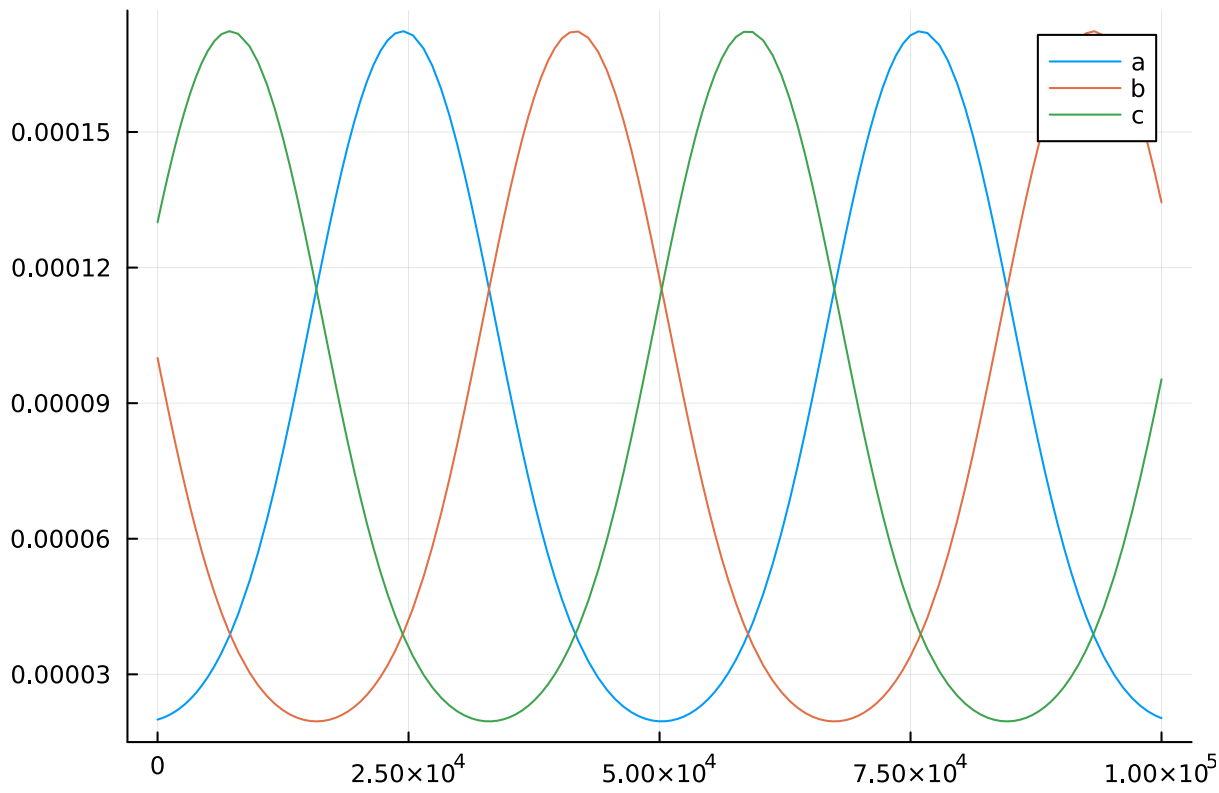
1.1 Ideal

In [6]:
```
rps_ideal_rn = @reaction_network begin
    q, c + b --> c + c
    q, a + c --> a + a
    q, b + a --> b + b
end q
print_species_array(rps_ideal_rn)
```

```
species = ['c(t)','b(t)','a(t)',]
```

In [7]:
```
tspan = (0, 100000)
ainit = 2e-5
binit = 10e-5
cinit = 13e-5
u0 = [cinit, binit, ainit]
p = [1]
oprob = ODEProblem(rps_ideal_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)

aplot = aggregate_values(rps_ideal_rn, sol, prefix="a")
bplot = aggregate_values(rps_ideal_rn, sol, prefix="b")
cplot = aggregate_values(rps_ideal_rn, sol, prefix="c")
t = sol.t
plot([t, t, t], [aplot, bplot, cplot], labels=["a" "b" "c"])
```

Out[7]:



1.2 RPS (Leak = No Shadow = No)

In [8]:

```
rps_rn = @reaction_network begin
    q, c + gate_L_c_b --> buffer_H_c_b_leaks_c + BSbicb
    qmax, BSbicb + buffer_H_c_b_leaks_c --> c + gate_L_c_b
    qmax, b + buffer_H_c_b_leaks_c --> Ibicb
    qmax, Ibicb + gate_T_c_b_leaks_c_c --> c + c
    q, a + gate_L_a_c --> buffer_H_a_c_leaks_a + BSbiac
    qmax, BSbiac + buffer_H_a_c_leaks_a --> a + gate_L_a_c
    qmax, c + buffer_H_a_c_leaks_a --> Ibiac
    qmax, Ibiac + gate_T_a_c_leaks_a_a --> a + a
    q, b + gate_L_b_a --> buffer_H_b_a_leaks_b + BSbiba
    qmax, BSbiba + buffer_H_b_a_leaks_b --> b + gate_L_b_a
    qmax, a + buffer_H_b_a_leaks_b --> Ibiba
    qmax, Ibiba + gate_T_b_a_leaks_b_b --> b + b
end q qmax
```

Out[8]:

$$\mathrm{c} + \mathrm{gate_{L\_c\_b}} \underset{\$qmax\$}{\overset{\$q\$}{\rightleftharpoons}} \mathrm{buffer_{H\_c\_b\_leaks\_c}} + \mathrm{BSbicb}$$

$$\mathrm{b} + \mathrm{buffer_{H\_c\_b\_leaks\_c}} \xrightarrow{\$qmax\$} \mathrm{Ibicb}$$

$$\mathrm{Ibicb} + \mathrm{gate_{T\_c\_b\_leaks\_c\_c}} \xrightarrow{\$qmax\$} 2\,\mathrm{c}$$

$$\mathrm{a} + \mathrm{gate_{L\_a\_c}} \underset{\$qmax\$}{\overset{\$q\$}{\rightleftharpoons}} \mathrm{buffer_{H\_a\_c\_leaks\_a}} + \mathrm{BSbiac}$$

$$\mathrm{c} + \mathrm{buffer_{H\_a\_c\_leaks\_a}} \xrightarrow{\$qmax\$} \mathrm{Ibiac}$$

$$\mathrm{Ibiac} + \mathrm{gate_{T\_a\_c\_leaks\_a\_a}} \xrightarrow{\$qmax\$} 2\,\mathrm{a}$$

$$\mathrm{b} + \mathrm{gate_{L\_b\_a}} \underset{\$qmax\$}{\overset{\$q\$}{\rightleftharpoons}} \mathrm{buffer_{H\_b\_a\_leaks\_b}} + \mathrm{BSbiba}$$

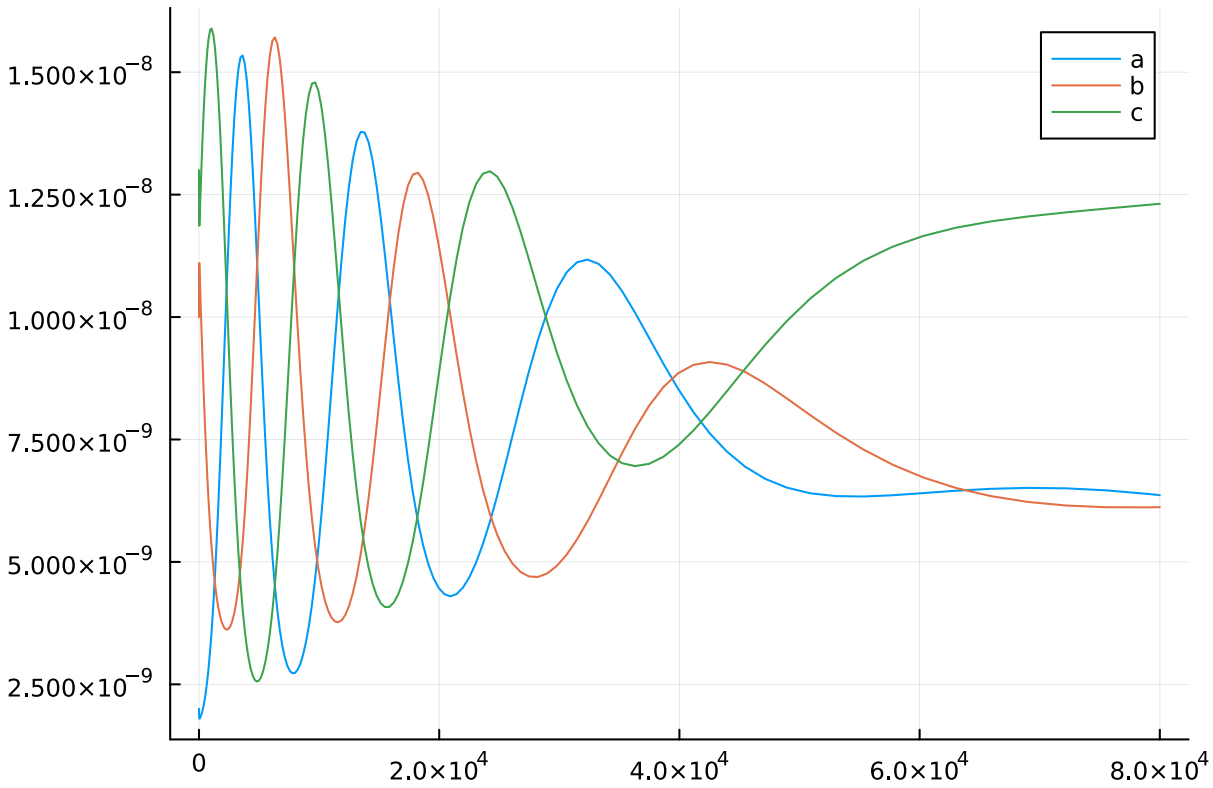
$$\mathrm{a} + \mathrm{buffer_{H\_b\_a\_leaks\_b}} \xrightarrow{\$qmax\$} \mathrm{Ibiba}$$

$$\mathrm{Ibiba} + \mathrm{gate_{T\_b\_a\_leaks\_b\_b}} \xrightarrow{\$qmax\$} 2\,\mathrm{b}$$

In [9]:

```
plot()

qmax = 1e6 # /M sec
k = 1e5 # scaled.
sigma = k

gamma_inv = qmax/(qmax-sigma) # 2
q = gamma_inv*k

ainit = 2*1e-9
binit = 10*1e-9
cinit = 13*1e-9

Cmax = 100e-9 # 100nM

u0 = [
cinit, Cmax, 0.0, Cmax, binit,
0.0, Cmax, ainit, Cmax, 0.0,
Cmax, 0.0, Cmax, Cmax, 0.0,
Cmax, 0.0, Cmax,
]

p = [q, qmax]

tspan = (0, 80000)
oprob = ODEProblem(rps_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)

aplot = aggregate_values(rps_rn, sol, prefix="a")
bplot = aggregate_values(rps_rn, sol, prefix="b")
cplot = aggregate_values(rps_rn, sol, prefix="c")
t = sol.t
plot([t, t, t], [aplot, bplot, cplot], labels=["a" "b" "c"])
```

Out[9]:



1.3 RPS (Leak = Yes, Shadow = No)

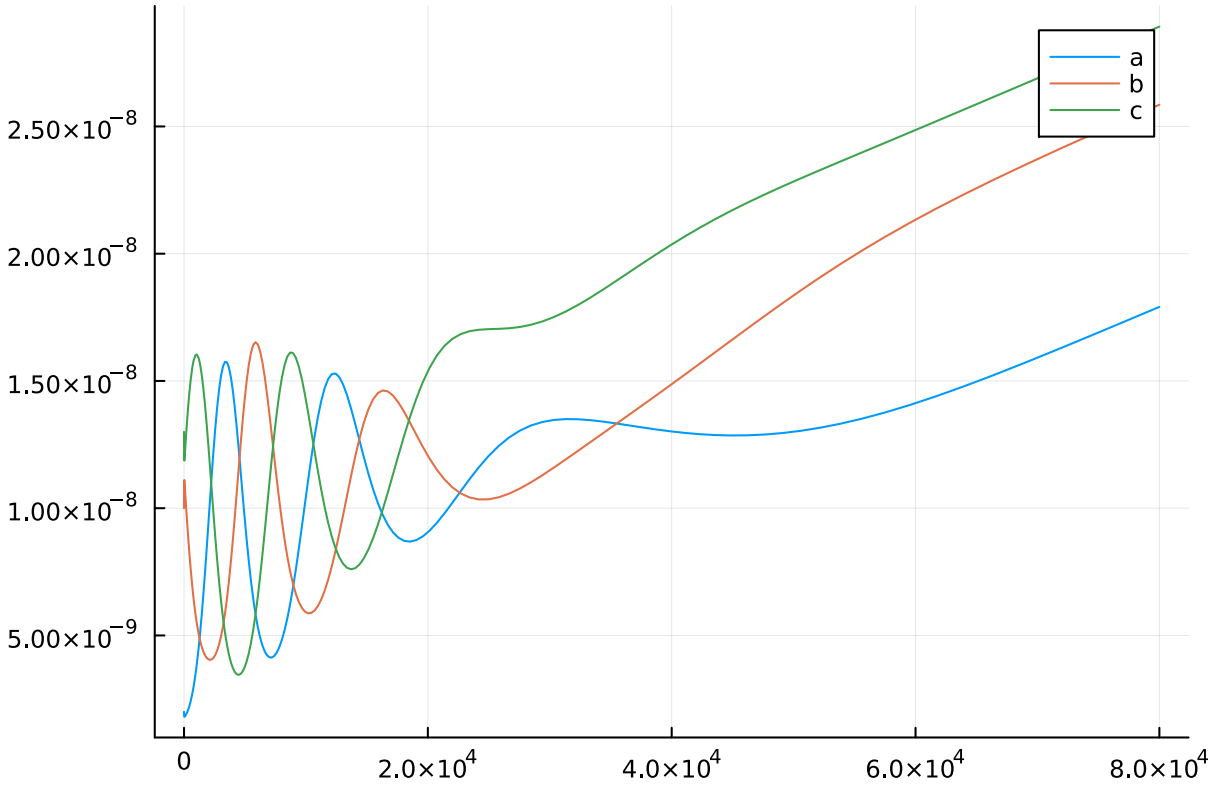
In [10]:
```
rps_rn = @reaction_network begin
    q, c + gate_L_c_b --> buffer_H_c_b_leaks_c + BSbicb
    qmax, BSbicb + buffer_H_c_b_leaks_c --> c + gate_L_c_b
    qmax, b + buffer_H_c_b_leaks_c --> Ibicb
    qmax, Ibicb + gate_T_c_b_leaks_c_c --> c + c
    q, a + gate_L_a_c --> buffer_H_a_c_leaks_a + BSbiac
    qmax, BSbiac + buffer_H_a_c_leaks_a --> a + gate_L_a_c
    qmax, c + buffer_H_a_c_leaks_a --> Ibiac
    qmax, Ibiac + gate_T_a_c_leaks_a_a --> a + a
    q, b + gate_L_b_a --> buffer_H_b_a_leaks_b + BSbiba
    qmax, BSbiba + buffer_H_b_a_leaks_b --> b + gate_L_b_a
    qmax, a + buffer_H_b_a_leaks_b --> Ibiba
    qmax, Ibiba + gate_T_b_a_leaks_b_b --> b + b

    # Leak reactions
    leak*leak_rate, 0 --> c + c
    leak*leak_rate, 0 --> a + a
    leak*leak_rate, 0 --> b + b
end q qmax leak leak_rate
print_species_array(rps_rn)
```

```
species = ['c(t)','gate_L_c_b(t)','buffer_H_c_b_leaks_c(t)','BSbicb(t)','
b(t)','Ibicb(t)','gate_T_c_b_leaks_c_c(t)','a(t)','gate_L_a_c(t)','buffer
_H_a_c_leaks_a(t)','BSbiac(t)','Ibiac(t)','gate_T_a_c_leaks_a_a(t)','gate
_L_b_a(t)','buffer_H_b_a_leaks_b(t)','BSbiba(t)','Ibiba(t)','gate_T_b_a_l
eaks_b_b(t)',]
```

In [11]:

```
plot()

qmax = 1e6 # /M sec
k = 1e5 # scaled.
sigma = k

gamma_inv = qmax/(qmax-sigma) # 2
q = gamma_inv*k

ainit = 2*1e-9
binit = 10*1e-9
cinit = 13*1e-9

Cmax = 100e-9 # 100nM

leak = 1
leak_rate = 1e-13 # 0.36 nM/hr

u0 = [
cinit, Cmax, 0.0, Cmax, binit,
0.0, Cmax, ainit, Cmax, 0.0,
Cmax, 0.0, Cmax, Cmax, 0.0,
Cmax, 0.0, Cmax,
]

p = [q, qmax, leak, leak_rate]

tspan = (0, 80000)
oprob = ODEProblem(rps_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)

aplot = aggregate_values(rps_rn, sol, prefix="a")
bplot = aggregate_values(rps_rn, sol, prefix="b")
cplot = aggregate_values(rps_rn, sol, prefix="c")
t = sol.t
plot([t, t, t], [aplot, bplot, cplot], labels=["a" "b" "c"])
```

Out[11]:



1.4 RPS (Leaks = Yes, Shadow = Yes)

In [12]:

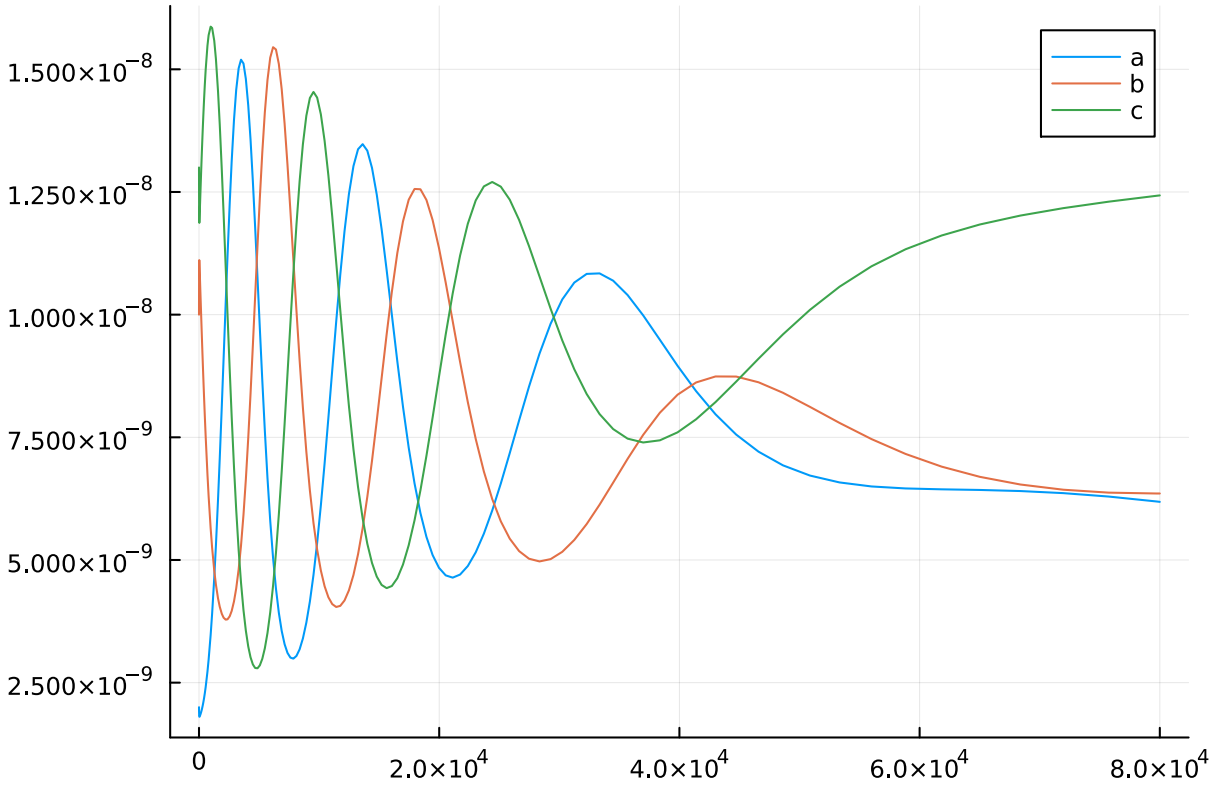
```
rps_rn = @reaction_network begin
    q, c + gate_L_c_b --> buffer_H_c_b_leaks_c + BSbicb
    shadow*q, shadow_c + shadow_gate_L_c_b --> shadow_buffer_H_c_b_leaks_
    qmax, BSbicb + buffer_H_c_b_leaks_c --> c + gate_L_c_b
    shadow*qmax, shadow_BSbicb + shadow_buffer_H_c_b_leaks_c --> shadow_c
    qmax, b + buffer_H_c_b_leaks_c --> Ibicb
    shadow*qmax, shadow_b + shadow_buffer_H_c_b_leaks_c --> shadow_Ibicb
    qmax, Ibicb + gate_T_c_b_leaks_c_c --> c + c
    shadow*qmax, shadow_Ibicb + shadow_gate_T_c_b_leaks_c_c --> shadow_c
    q, a + gate_L_a_c --> buffer_H_a_c_leaks_a + BSbiac
    shadow*q, shadow_a + shadow_gate_L_a_c --> shadow_buffer_H_a_c_leaks_
    qmax, BSbiac + buffer_H_a_c_leaks_a --> a + gate_L_a_c
    shadow*qmax, shadow_BSbiac + shadow_buffer_H_a_c_leaks_a --> shadow_a
    qmax, c + buffer_H_a_c_leaks_a --> Ibiac
    shadow*qmax, shadow_c + shadow_buffer_H_a_c_leaks_a --> shadow_Ibiac
    qmax, Ibiac + gate_T_a_c_leaks_a_a --> a + a
    shadow*qmax, shadow_Ibiac + shadow_gate_T_a_c_leaks_a_a --> shadow_a
    q, b + gate_L_b_a --> buffer_H_b_a_leaks_b + BSbiba
    shadow*q, shadow_b + shadow_gate_L_b_a --> shadow_buffer_H_b_a_leaks_
    qmax, BSbiba + buffer_H_b_a_leaks_b --> b + gate_L_b_a
    shadow*qmax, shadow_BSbiba + shadow_buffer_H_b_a_leaks_b --> shadow_b
    qmax, a + buffer_H_b_a_leaks_b --> Ibiba
    shadow*qmax, shadow_a + shadow_buffer_H_b_a_leaks_b --> shadow_Ibiba
    qmax, Ibiba + gate_T_b_a_leaks_b_b --> b + b
    shadow*qmax, shadow_Ibiba + shadow_gate_T_b_a_leaks_b_b --> shadow_b

    # Leak reactions
    leak*leak_rate, 0 --> c + c
    leak*leak_rate, 0 --> a + a
    leak*leak_rate, 0 --> b + b

    # Shadow leaks
    shadow*leak*leak_rate, 0 --> shadow_c + shadow_c
    shadow*leak*leak_rate, 0 --> shadow_a + shadow_a
    shadow*leak*leak_rate, 0 --> shadow_b + shadow_b

    # Annihilation
    shadow*leak*annih, shadow_a + a --> 0
    shadow*leak*annih, shadow_b + b --> 0
    shadow*leak*annih, shadow_c + c --> 0


end q qmax leak leak_rate shadow annih
print_species_array(rps_rn)
```

```
species = ['c(t)','gate_L_c_b(t)','buffer_H_c_b_leaks_c(t)','BSbicb(t)','
shadow_c(t)','shadow_gate_L_c_b(t)','shadow_buffer_H_c_b_leaks_c(t)','sha
dow_BSbicb(t)','b(t)','Ibicb(t)','shadow_b(t)','shadow_Ibicb(t)','gate_T_
c_b_leaks_c_c(t)','shadow_gate_T_c_b_leaks_c_c(t)','a(t)','gate_L_a_c(t)'
,'buffer_H_a_c_leaks_a(t)','BSbiac(t)','shadow_a(t)','shadow_gate_L_a_c(t
)','shadow_buffer_H_a_c_leaks_a(t)','shadow_BSbiac(t)','Ibiac(t)','shadow
_Ibiac(t)','gate_T_a_c_leaks_a_a(t)','shadow_gate_T_a_c_leaks_a_a(t)','ga
te_L_b_a(t)','buffer_H_b_a_leaks_b(t)','BSbiba(t)','shadow_gate_L_b_a(t)'
,'shadow_buffer_H_b_a_leaks_b(t)','shadow_BSbiba(t)','Ibiba(t)','shadow_I
biba(t)','gate_T_b_a_leaks_b_b(t)','shadow_gate_T_b_a_leaks_b_b(t)',]
```

In [13]:

```
plot()

qmax = 1e6 # /M sec
k = 1e5 # scaled.
sigma = k

gamma_inv = qmax/(qmax-sigma) # 2
q = gamma_inv*k

ainit = 2*1e-9
binit = 10*1e-9
cinit = 13*1e-9

Cmax = 100e-9 # 100nM

leak = 1
leak_rate = 1e-13 # M/sec equivalent to 0.36 nM/hr
shadow = 1
# Taken from here: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5739081/
annih = 5e5

u0 = [
cinit, Cmax, 0.0, Cmax, 0.0,
Cmax, 0.0, Cmax, binit, 0.0,
0.0, 0.0, Cmax, Cmax, ainit,
Cmax, 0.0, Cmax, 0.0, Cmax,
0.0, Cmax, 0.0, 0.0, Cmax,
Cmax, Cmax, 0.0, Cmax, Cmax,
0.0, Cmax, 0.0, 0.0, Cmax,
Cmax,
]

p = [q, qmax, leak, leak_rate, shadow, annih]

tspan = (0, 80000)
oprob = ODEProblem(rps_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)

aplot = aggregate_values(rps_rn, sol, prefix="a")
bplot = aggregate_values(rps_rn, sol, prefix="b")
cplot = aggregate_values(rps_rn, sol, prefix="c")
t = sol.t
plot([t, t, t], [aplot, bplot, cplot], labels=["a" "b" "c"])
```

Out[13]:



2. Consensus

2.1 Ideal

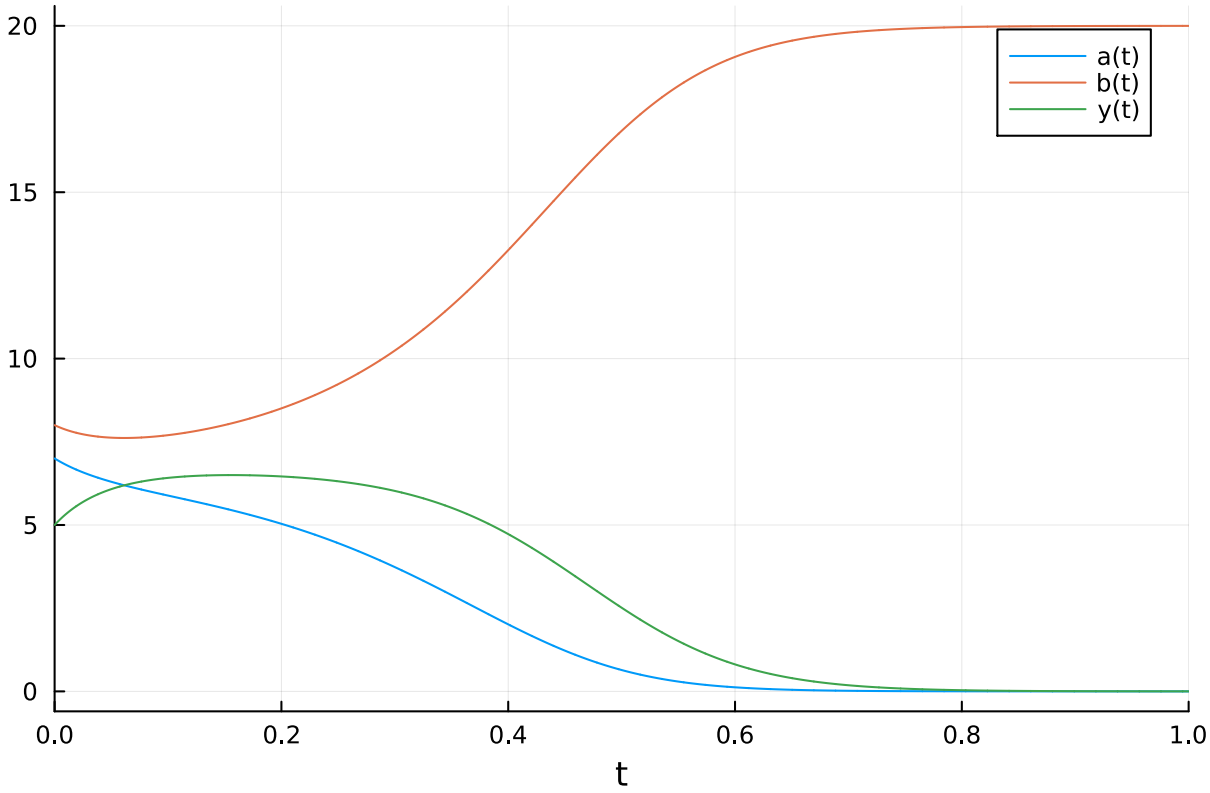
In [14]:
```
consensus_ideal_rn = @reaction_network begin
    1, a + b --> y + y
    1, a + y --> a + a
    1, b + y --> b + b
end
```

Out[14]:

$$\mathrm{a} + \mathrm{b} \xrightarrow{\$1\$} 2\,\mathrm{y}$$
$$\mathrm{a} + \mathrm{y} \xrightarrow{\$1\$} 2\,\mathrm{a}$$
$$\mathrm{b} + \mathrm{y} \xrightarrow{\$1\$} 2\,\mathrm{b}$$

In [15]:
```
plot()
ainit = 7
binit = 8
yinit = 5


u0 = [ainit, binit, yinit]
p = []
tspan = (0, 1)
oprob = ODEProblem(consensus_ideal_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)
plot(sol)
```

Out[15]:



2.2 Consensus (Leak = No, Shadow = No)

In [16]:

```
consensus_rn = @reaction_network begin
    q1, a + gate_L_a_b --> buffer_H_a_b_leaks_a + BSbiab
    qmax, BSbiab + buffer_H_a_b_leaks_a --> a + gate_L_a_b
    qmax, b + buffer_H_a_b_leaks_a --> Ibiab
    qmax, Ibiab + gate_T_a_b_leaks_y_y --> y + y
    q2, a + gate_L_a_y --> buffer_H_a_y_leaks_a + BSbiay
    qmax, BSbiay + buffer_H_a_y_leaks_a --> a + gate_L_a_y
    qmax, y + buffer_H_a_y_leaks_a --> Ibiay
    qmax, Ibiay + gate_T_a_y_leaks_a_a --> a + a
    q3, b + gate_L_b_y --> buffer_H_b_y_leaks_b + BSbiby
    qmax, BSbiby + buffer_H_b_y_leaks_b --> b + gate_L_b_y
    qmax, y + buffer_H_b_y_leaks_b --> Ibiby
    qmax, Ibiby + gate_T_b_y_leaks_b_b --> b + b

    # Additional buffering to adjust
    qsb, b + gate_LS_b --> buffer_HS_b_leaks_b + BSbib
    qmax, buffer_HS_b_leaks_b + BSbib --> b + gate_LS_b

    qsy, y + gate_LS_y --> buffer_HS_y_leaks_y + BSbiy
    qmax, buffer_HS_y_leaks_y + BSbiy --> y + gate_LS_y

end q1 q2 q3 qmax qsb qsy
```

Out[16]:

$$\mathrm{a} + \mathrm{gate}_{\mathrm{L\_a\_b}} \underset{\$qmax\$}{\overset{\$q1\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{H\_a\_b\_leaks\_a}} + \mathrm{BSbiab}$$

$$\mathrm{b} + \mathrm{buffer}_{\mathrm{H\_a\_b\_leaks\_a}} \xrightarrow{\$qmax\$} \mathrm{Ibiab}$$

$$\mathrm{Ibiab} + \mathrm{gate}_{\mathrm{T\_a\_b\_leaks\_y\_y}} \xrightarrow{\$qmax\$} 2\,\mathrm{y}$$

$$\mathrm{a} + \mathrm{gate}_{\mathrm{L\_a\_y}} \underset{\$qmax\$}{\overset{\$q2\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{H\_a\_y\_leaks\_a}} + \mathrm{BSbiay}$$

$$\mathrm{y} + \mathrm{buffer}_{\mathrm{H\_a\_y\_leaks\_a}} \xrightarrow{\$qmax\$} \mathrm{Ibiay}$$

$$\mathrm{Ibiay} + \mathrm{gate}_{\mathrm{T\_a\_y\_leaks\_a\_a}} \xrightarrow{\$qmax\$} 2\,\mathrm{a}$$

$$\mathrm{b} + \mathrm{gate}_{\mathrm{L\_b\_y}} \underset{\$qmax\$}{\overset{\$q3\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{H\_b\_y\_leaks\_b}} + \mathrm{BSbiby}$$

$$\mathrm{y} + \mathrm{buffer}_{\mathrm{H\_b\_y\_leaks\_b}} \xrightarrow{\$qmax\$} \mathrm{Ibiby}$$

$$\mathrm{Ibiby} + \mathrm{gate}_{\mathrm{T\_b\_y\_leaks\_b\_b}} \xrightarrow{\$qmax\$} 2\,\mathrm{b}$$

$$\mathrm{b} + \mathrm{gate}_{\mathrm{LS\_b}} \underset{\$qmax\$}{\overset{\$qsb\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{HS\_b\_leaks\_b}} + \mathrm{BSbib}$$

$$\mathrm{y} + \mathrm{gate}_{\mathrm{LS\_y}} \underset{\$qmax\$}{\overset{\$qsy\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{HS\_y\_leaks\_y}} + \mathrm{BSbiy}$$

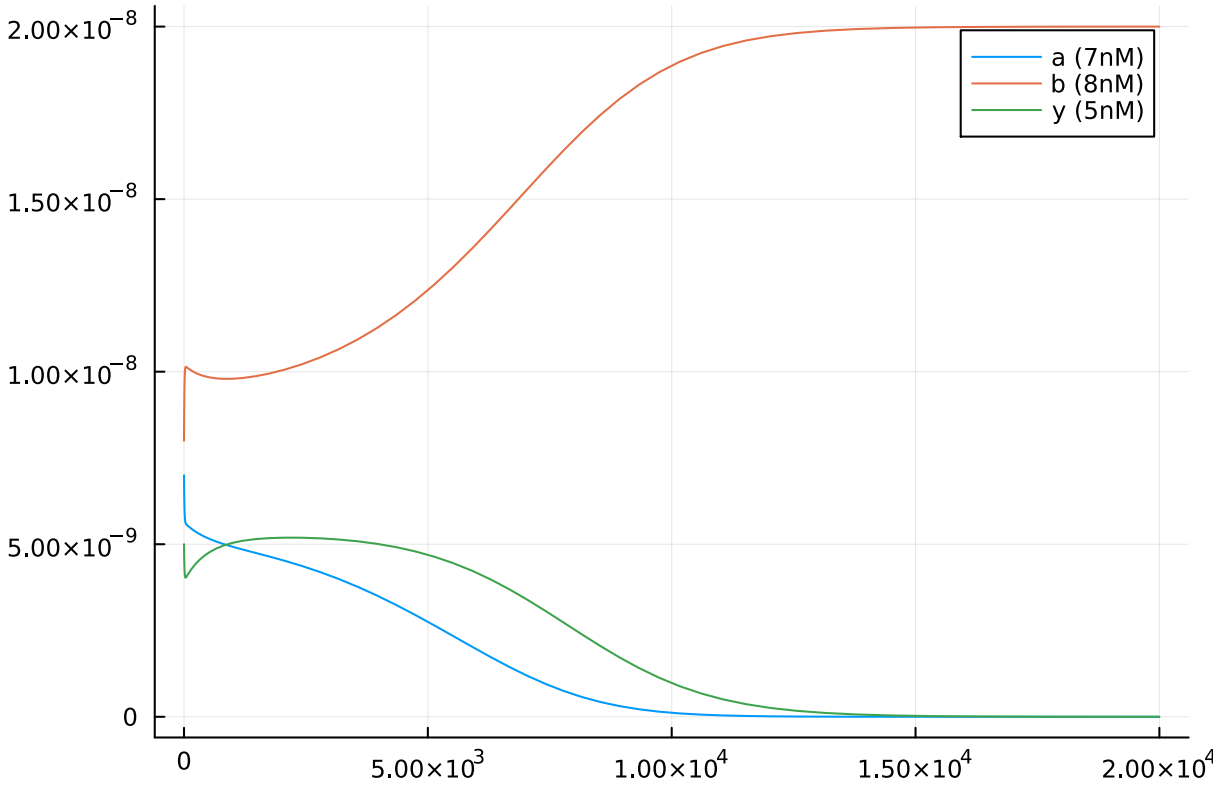
In [17]:
```
print_species_array(consensus_rn)
```

```
species = ['a(t)','gate_L_a_b(t)','buffer_H_a_b_leaks_a(t)','BSbiab(t)','
b(t)','Ibiab(t)','gate_T_a_b_leaks_y_y(t)','y(t)','gate_L_a_y(t)','buffer
_H_a_y_leaks_a(t)','BSbiay(t)','Ibiay(t)','gate_T_a_y_leaks_a_a(t)','gate
_L_b_y(t)','buffer_H_b_y_leaks_b(t)','BSbiby(t)','Ibiby(t)','gate_T_b_y_l
eaks_b_b(t)','gate_LS_b(t)','buffer_HS_b_leaks_b(t)','BSbib(t)','gate_LS_
y(t)','buffer_HS_y_leaks_y(t)','BSbiy(t)',]
```

In [18]:

```
plot()

qmax = 1e6 # /M sec
k = 1e5 # scaled. 1/(1e3*1e-8)

sigma = 2*k # sigma_a = 2k, sigma_b = k, sigma_y = 0
sigma_a = 2*k
sigma_b = k
sigma_y = 0

gamma_inv = qmax/(qmax-sigma) # 1e6/(8*1e5) = 5/4
q1 = gamma_inv*k
q2 = gamma_inv*k
q3 = gamma_inv*k
qsb = gamma_inv*k
qsy = gamma_inv*2*k

ainit = gamma_inv*7*1e-9
binit = gamma_inv*8*1e-9
yinit = gamma_inv*5*1e-9

Cmax = 100e-9 # 100nM


u0 = [
ainit, Cmax, 0.0, Cmax, binit,
0.0, Cmax, yinit, Cmax, 0.0,
Cmax, 0.0, Cmax, Cmax, 0.0,
Cmax, 0.0, Cmax, Cmax, 0.0,
Cmax, Cmax, 0.0, Cmax,
]

p = [q1, q2, q3, qmax, qsb, qsy]

tspan = (0, 20000)
oprob = ODEProblem(consensus_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)
aplot = aggregate_values(consensus_rn, sol, prefix="a")/gamma_inv
bplot = aggregate_values(consensus_rn, sol, prefix="b")/gamma_inv
yplot = aggregate_values(consensus_rn, sol, prefix="y")/gamma_inv
t = sol.t
plot([t, t, t], [aplot, bplot, yplot], labels=["a (7nM)" "b (8nM)" "y (5n
# bsbib = aggregate_values(consensus_dsd_rn, sol, prefix="BSbib")
# plot(bsbib)
```

Out[18]:



2.3 Consensus (Leak = Yes, Shadow = No)

In [19]:

```
consensus_rn = @reaction_network begin
    q1, a + gate_L_a_b --> buffer_H_a_b_leaks_a + BSbiab
    qmax, BSbiab + buffer_H_a_b_leaks_a --> a + gate_L_a_b
    qmax, b + buffer_H_a_b_leaks_a --> Ibiab
    qmax, Ibiab + gate_T_a_b_leaks_y_y --> y + y
    q2, a + gate_L_a_y --> buffer_H_a_y_leaks_a + BSbiay
    qmax, BSbiay + buffer_H_a_y_leaks_a --> a + gate_L_a_y
    qmax, y + buffer_H_a_y_leaks_a --> Ibiay
    qmax, Ibiay + gate_T_a_y_leaks_a_a --> a + a
    q3, b + gate_L_b_y --> buffer_H_b_y_leaks_b + BSbiby
    qmax, BSbiby + buffer_H_b_y_leaks_b --> b + gate_L_b_y
    qmax, y + buffer_H_b_y_leaks_b --> Ibiby
    qmax, Ibiby + gate_T_b_y_leaks_b_b --> b + b

    # Additional buffering to adjust
    qsb, b + gate_LS_b --> buffer_HS_b_leaks_b + BSbib
    qmax, buffer_HS_b_leaks_b + BSbib --> b + gate_LS_b

    qsy, y + gate_LS_y --> buffer_HS_y_leaks_y + BSbiy
    qmax, buffer_HS_y_leaks_y + BSbiy --> y + gate_LS_y

    # Leak reactions
    leak*leak_rate, 0 --> y + y
    leak*leak_rate, 0 --> b + b
    leak*leak_rate, 0 --> a + a


end q1 q2 q3 qmax qsb qsy leak leak_rate
```

Out[19]:

$$\begin{aligned}
\mathrm{a} + \mathrm{gate}_{\mathrm{L_a_b}} &\underset{\$qmax\$}{\overset{\$q1\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{H_a_b_leaks_a}} + \mathrm{BSbiab} \\
\mathrm{b} + \mathrm{buffer}_{\mathrm{H_a_b_leaks_a}} &\xrightarrow{\$qmax\$} \mathrm{Ibiab} \\
\mathrm{Ibiab} + \mathrm{gate}_{\mathrm{T_a_b_leaks_y_y}} &\xrightarrow{\$qmax\$} 2\,\mathrm{y} \\
\mathrm{a} + \mathrm{gate}_{\mathrm{L_a_y}} &\underset{\$qmax\$}{\overset{\$q2\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{H_a_y_leaks_a}} + \mathrm{BSbiay} \\
\mathrm{y} + \mathrm{buffer}_{\mathrm{H_a_y_leaks_a}} &\xrightarrow{\$qmax\$} \mathrm{Ibiay} \\
\mathrm{Ibiay} + \mathrm{gate}_{\mathrm{T_a_y_leaks_a_a}} &\xrightarrow{\$qmax\$} 2\,\mathrm{a} \\
\mathrm{b} + \mathrm{gate}_{\mathrm{L_b_y}} &\underset{\$qmax\$}{\overset{\$q3\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{H_b_y_leaks_b}} + \mathrm{BSbiby} \\
\mathrm{y} + \mathrm{buffer}_{\mathrm{H_b_y_leaks_b}} &\xrightarrow{\$qmax\$} \mathrm{Ibiby} \\
\mathrm{Ibiby} + \mathrm{gate}_{\mathrm{T_b_y_leaks_b_b}} &\xrightarrow{\$qmax\$} 2\,\mathrm{b} \\
\mathrm{b} + \mathrm{gate}_{\mathrm{LS_b}} &\underset{\$qmax\$}{\overset{\$qsb\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{HS_b_leaks_b}} + \mathrm{BSbib} \\
\mathrm{y} + \mathrm{gate}_{\mathrm{LS_y}} &\underset{\$qmax\$}{\overset{\$qsy\$}{\rightleftharpoons}} \mathrm{buffer}_{\mathrm{HS_y_leaks_y}} + \mathrm{BSbiy} \\
\varnothing &\xrightarrow{\$leakleak_{rate}\$} 2\,\mathrm{y} \\
\varnothing &\xrightarrow{\$leakleak_{rate}\$} 2\,\mathrm{b} \\
\varnothing &\xrightarrow{\$leakleak_{rate}\$} 2\,\mathrm{a}
\end{aligned}$$

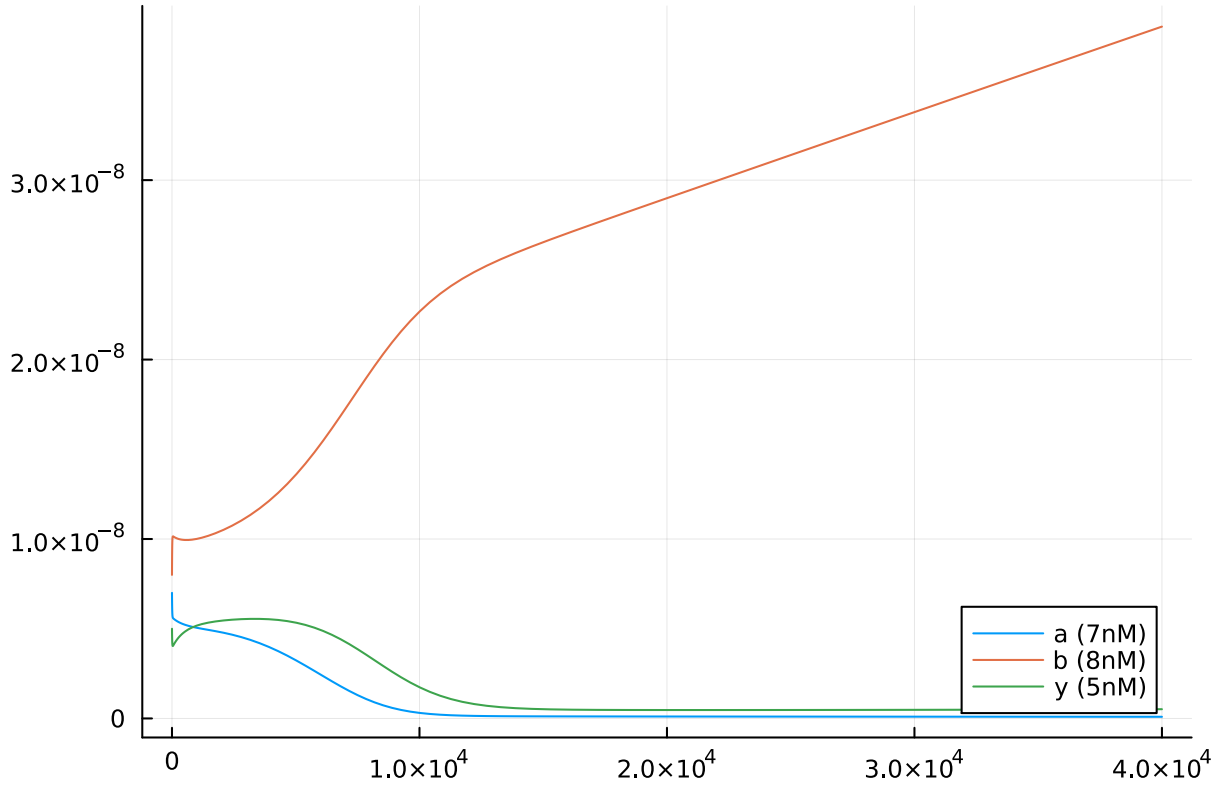
In [20]:
```
print_species_array(consensus_rn)
```

```
species = ['a(t)','gate_L_a_b(t)','buffer_H_a_b_leaks_a(t)','BSbiab(t)','
b(t)','Ibiab(t)','gate_T_a_b_leaks_y_y(t)','y(t)','gate_L_a_y(t)','buffer
_H_a_y_leaks_a(t)','BSbiay(t)','Ibiay(t)','gate_T_a_y_leaks_a_a(t)','gate
_L_b_y(t)','buffer_H_b_y_leaks_b(t)','BSbiby(t)','Ibiby(t)','gate_T_b_y_l
eaks_b_b(t)','gate_LS_b(t)','buffer_HS_b_leaks_b(t)','BSbib(t)','gate_LS_
y(t)','buffer_HS_y_leaks_y(t)','BSbiy(t)',]
```

In [21]:

```
plot()

qmax = 1e6 # /M sec
k = 1e5 # scaled. 1/(1e3*1e-8)

sigma = 2*k # sigma_a = 2k, sigma_b = k, sigma_y = 0
sigma_a = 2*k
sigma_b = k
sigma_y = 0

gamma_inv = qmax/(qmax-sigma) # 1e6/(8*1e5) = 5/4
q1 = gamma_inv*k
q2 = gamma_inv*k
q3 = gamma_inv*k
qsb = gamma_inv*k
qsy = gamma_inv*2*k

ainit = gamma_inv*7*1e-9
binit = gamma_inv*8*1e-9
yinit = gamma_inv*5*1e-9

Cmax = 100e-9 # 100nM

leak = 1
leak_rate = 1e-13 # M/sec or 0.36 nM/hr


u0 = [
ainit, Cmax, 0.0, Cmax, binit,
0.0, Cmax, yinit, Cmax, 0.0,
Cmax, 0.0, Cmax, Cmax, 0.0,
Cmax, 0.0, Cmax, Cmax, 0.0,
Cmax, Cmax, 0.0, Cmax,
]

p = [q1, q2, q3, qmax, qsb, qsy, leak, leak_rate]

tspan = (0, 40000)
oprob = ODEProblem(consensus_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)
aplot = aggregate_values(consensus_rn, sol, prefix="a")/gamma_inv
bplot = aggregate_values(consensus_rn, sol, prefix="b")/gamma_inv
yplot = aggregate_values(consensus_rn, sol, prefix="y")/gamma_inv
t = sol.t
plot([t, t, t], [aplot, bplot, yplot], labels=["a (7nM)" "b (8nM)" "y (5n
# bsbib = aggregate_values(consensus_dsd_rn, sol, prefix="BSbib")
# plot(bsbib)
```

Out[21]:



2.4 Consensus (Leak = Yes, Shadow = Yes)

In [22]:

```
consensus_rn = @reaction_network begin
    q1, a + gate_L_a_b --> buffer_H_a_b_leaks_a + BSbiab
    shadow*q1, shadow_a + shadow_gate_L_a_b --> shadow_buffer_H_a_b_leaks
    qmax, BSbiab + buffer_H_a_b_leaks_a --> a + gate_L_a_b
    shadow*qmax, shadow_BSbiab + shadow_buffer_H_a_b_leaks_a --> shadow_a
    qmax, b + buffer_H_a_b_leaks_a --> Ibiab
    shadow*qmax, shadow_b + shadow_buffer_H_a_b_leaks_a --> shadow_Ibiab
    qmax, Ibiab + gate_T_a_b_leaks_y_y --> y + y
    shadow*qmax, shadow_Ibiab + shadow_gate_T_a_b_leaks_y_y --> shadow_y
    q2, a + gate_L_a_y --> buffer_H_a_y_leaks_a + BSbiay
    shadow*q2, shadow_a + shadow_gate_L_a_y --> shadow_buffer_H_a_y_leaks
    qmax, BSbiay + buffer_H_a_y_leaks_a --> a + gate_L_a_y
    shadow*qmax, shadow_BSbiay + shadow_buffer_H_a_y_leaks_a --> shadow_a
    qmax, y + buffer_H_a_y_leaks_a --> Ibiay
    shadow*qmax, shadow_y + shadow_buffer_H_a_y_leaks_a --> shadow_Ibiay
    qmax, Ibiay + gate_T_a_y_leaks_a_a --> a + a
    shadow*qmax, shadow_Ibiay + shadow_gate_T_a_y_leaks_a_a --> shadow_a
    q3, b + gate_L_b_y --> buffer_H_b_y_leaks_b + BSbiby
    shadow*q3, shadow_b + shadow_gate_L_b_y --> shadow_buffer_H_b_y_leaks
    qmax, BSbiby + buffer_H_b_y_leaks_b --> b + gate_L_b_y
    shadow*qmax, shadow_BSbiby + shadow_buffer_H_b_y_leaks_b --> shadow_b
    qmax, y + buffer_H_b_y_leaks_b --> Ibiby
    shadow*qmax, shadow_y + shadow_buffer_H_b_y_leaks_b --> shadow_Ibiby
    qmax, Ibiby + gate_T_b_y_leaks_b_b --> b + b
    shadow*qmax, shadow_Ibiby + shadow_gate_T_b_y_leaks_b_b --> shadow_b

    # Additional buffering to adjust
    qsb, b + gate_LS_b --> buffer_HS_b_leaks_b + BSbib
    leak*shadow*qsb, shadow_b + shadow_gate_LS_b --> shadow_buffer_HS_b_l
    qmax, buffer_HS_b_leaks_b + BSbib --> b + gate_LS_b
    leak*shadow*qmax, shadow_buffer_HS_b_leaks_b + shadow_BSbib --> shado

    qsy, y + gate_LS_y --> buffer_HS_y_leaks_y + BSbiy
    qmax, buffer_HS_y_leaks_y + BSbiy --> y + gate_LS_y
    leak*shadow*qsy, shadow_y + shadow_gate_LS_y --> shadow_buffer_HS_y_l
    leak*shadow*qmax, shadow_buffer_HS_y_leaks_y + shadow_BSbiy --> shado

    # Leak reactions
    leak*leak_rate, 0 --> y + y
    leak*leak_rate, 0 --> b + b
    leak*leak_rate, 0 --> a + a

    # Shadow leak reactions
    shadow*leak*leak_rate, 0 --> shadow_y + shadow_y
    shadow*leak*leak_rate, 0 --> shadow_b + shadow_b
    shadow*leak*leak_rate, 0 --> shadow_a + shadow_a

    # Leak cancellation
    shadow*leak*annih, shadow_a + a --> 0
    shadow*leak*annih, shadow_b + b --> 0
    shadow*leak*annih, shadow_y + y --> 0

end q1 q2 q3 qmax qsb qsy leak leak_rate shadow annih
```

Out[22]:

$$\mathrm{a} + \mathrm{gate_{L\_a\_b}} \xrightarrow{\$q1\$} \mathrm{buffer_{H\_a\_b\_leaks\_a}} + \mathrm{BSbiab}$$

$$\mathrm{shadow_a} + \mathrm{shadow_{gate\_L\_a\_b}} \xrightarrow[\$qmax\$]{\$q1shadow\$} \mathrm{shadow_{buffer\_H\_a\_b\_leaks\_a}} + \mathrm{sh}$$

$$\mathrm{BSbiab} + \mathrm{buffer}_{\mathrm{H\_a\_b\_leaks\_a}} \longrightarrow \mathrm{a} + \mathrm{gate}_{\mathrm{L\_a\_b}}$$

$$\mathrm{shadow}_{\mathrm{BSbiab}} + \mathrm{shadow}_{\mathrm{buffer\_H\_a\_b\_leaks\_a}} \xrightarrow{\$qmaxshadow\$} \mathrm{shadow}_{\mathrm{a}} + \mathrm{shadow}_{\mathrm{gate\_L\_a}}$$

$$\mathrm{b} + \mathrm{buffer}_{\mathrm{H\_a\_b\_leaks\_a}} \xrightarrow{\$qmax\$} \mathrm{Ibiab}$$

$$\mathrm{shadow}_{\mathrm{b}} + \mathrm{shadow}_{\mathrm{buffer\_H\_a\_b\_leaks\_a}} \xrightarrow{\$qmaxshadow\$} \mathrm{shadow}_{\mathrm{Ibiab}}$$

$$\mathrm{Ibiab} + \mathrm{gate}_{\mathrm{T\_a\_b\_leaks\_y\_y}} \xrightarrow{\$qmax\$} 2\,\mathrm{y}$$

$$\mathrm{shadow}_{\mathrm{Ibiab}} + \mathrm{shadow}_{\mathrm{gate\_T\_a\_b\_leaks\_y\_y}} \xrightarrow{\$qmaxshadow\$} 2\,\mathrm{shadow}_{\mathrm{y}}$$

$$\mathrm{a} + \mathrm{gate}_{\mathrm{L\_a\_y}} \xrightarrow{\$q2\$} \mathrm{buffer}_{\mathrm{H\_a\_y\_leaks\_a}} + \mathrm{BSbiay}$$

$$\mathrm{shadow}_{\mathrm{a}} + \mathrm{shadow}_{\mathrm{gate\_L\_a\_y}} \xrightarrow{\$q2shadow\$} \mathrm{shadow}_{\mathrm{buffer\_H\_a\_y\_leaks\_a}} + \mathrm{sh}$$

$$\mathrm{BSbiay} + \mathrm{buffer}_{\mathrm{H\_a\_y\_leaks\_a}} \xrightarrow{\$qmax\$} \mathrm{a} + \mathrm{gate}_{\mathrm{L\_a\_y}}$$

$$\mathrm{shadow}_{\mathrm{BSbiay}} + \mathrm{shadow}_{\mathrm{buffer\_H\_a\_y\_leaks\_a}} \xrightarrow{\$qmaxshadow\$} \mathrm{shadow}_{\mathrm{a}} + \mathrm{shadow}_{\mathrm{gate\_L\_a}}$$

$$\mathrm{y} + \mathrm{buffer}_{\mathrm{H\_a\_y\_leaks\_a}} \xrightarrow{\$qmax\$} \mathrm{Ibiay}$$

$$\mathrm{shadow}_{\mathrm{y}} + \mathrm{shadow}_{\mathrm{buffer\_H\_a\_y\_leaks\_a}} \xrightarrow{\$qmaxshadow\$} \mathrm{shadow}_{\mathrm{Ibiay}}$$

$$\mathrm{Ibiay} + \mathrm{gate}_{\mathrm{T\_a\_y\_leaks\_a\_a}} \xrightarrow{\$qmax\$} 2\,\mathrm{a}$$

$$\mathrm{shadow}_{\mathrm{Ibiay}} + \mathrm{shadow}_{\mathrm{gate\_T\_a\_y\_leaks\_a\_a}} \xrightarrow{\$qmaxshadow\$} 2\,\mathrm{shadow}_{\mathrm{a}}$$

$$\mathrm{b} + \mathrm{gate}_{\mathrm{L\_b\_y}} \xrightarrow{\$q3\$} \mathrm{buffer}_{\mathrm{H\_b\_y\_leaks\_b}} + \mathrm{BSbiby}$$

$$\mathrm{shadow}_{\mathrm{b}} + \mathrm{shadow}_{\mathrm{gate\_L\_b\_y}} \xrightarrow{\$q3shadow\$} \mathrm{shadow}_{\mathrm{buffer\_H\_b\_y\_leaks\_b}} + \mathrm{sh}$$

$$\mathrm{BSbiby} + \mathrm{buffer}_{\mathrm{H\_b\_y\_leaks\_b}} \xrightarrow{\$qmax\$} \mathrm{b} + \mathrm{gate}_{\mathrm{L\_b\_y}}$$

$$\mathrm{shadow}_{\mathrm{BSbiby}} + \mathrm{shadow}_{\mathrm{buffer\_H\_b\_y\_leaks\_b}} \xrightarrow{\$qmaxshadow\$} \mathrm{shadow}_{\mathrm{b}} + \mathrm{shadow}_{\mathrm{gate\_L\_b}}$$

$$\mathrm{y} + \mathrm{buffer}_{\mathrm{H\_b\_y\_leaks\_b}} \xrightarrow{\$qmax\$} \mathrm{Ibiby}$$

$$\mathrm{shadow}_{\mathrm{y}} + \mathrm{shadow}_{\mathrm{buffer\_H\_b\_y\_leaks\_b}} \xrightarrow{\$qmaxshadow\$} \mathrm{shadow}_{\mathrm{Ibiby}}$$

$$\mathrm{Ibiby} + \mathrm{gate}_{\mathrm{T\_b\_y\_leaks\_b\_b}} \xrightarrow{\$qmax\$} 2\,\mathrm{b}$$

$$\mathrm{shadow}_{\mathrm{Ibiby}} + \mathrm{shadow}_{\mathrm{gate\_T\_b\_y\_leaks\_b\_b}} \xrightarrow{\$qmaxshadow\$} 2\,\mathrm{shadow}_{\mathrm{b}}$$

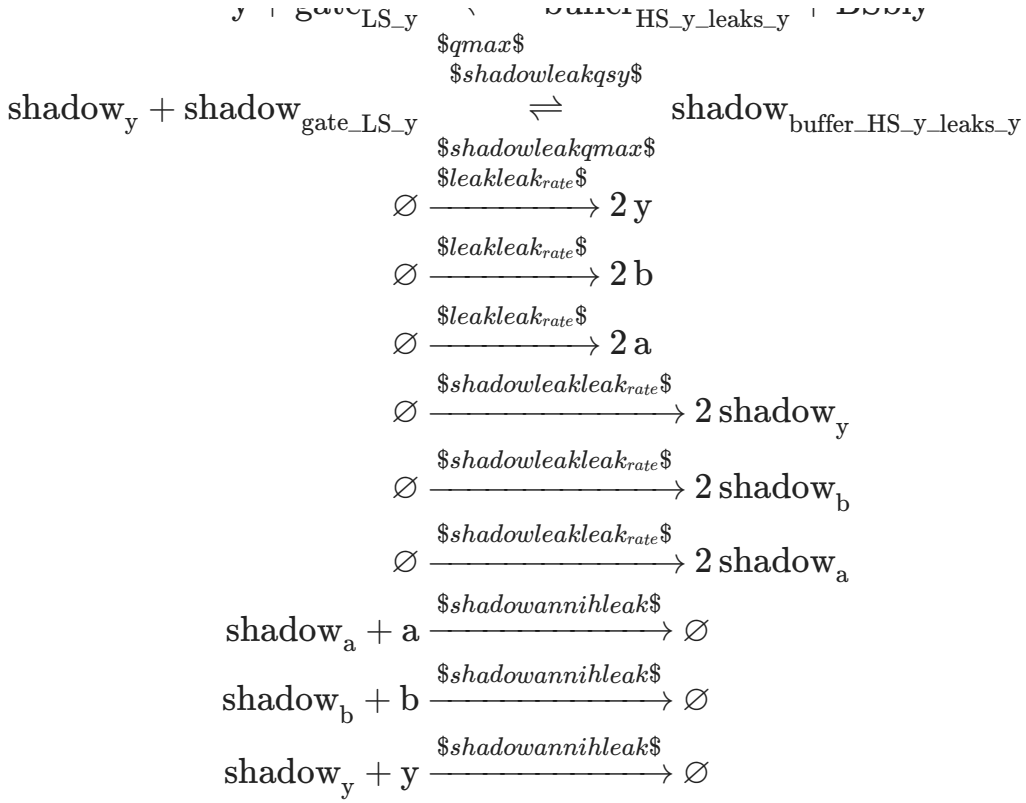
$$\mathrm{b} + \mathrm{gate}_{\mathrm{LS\_b}} \xrightarrow{\$qsb\$} \mathrm{buffer}_{\mathrm{HS\_b\_leaks\_b}} + \mathrm{BSbib}$$

$$\mathrm{shadow}_{\mathrm{b}} + \mathrm{shadow}_{\mathrm{gate\_LS\_b}} \xrightarrow{\$shadowleakqsb\$} \mathrm{shadow}_{\mathrm{buffer\_HS\_b\_leaks\_b}} +$$

$$\mathrm{buffer}_{\mathrm{HS\_b\_leaks\_b}} + \mathrm{BSbib} \xrightarrow{\$qmax\$} \mathrm{b} + \mathrm{gate}_{\mathrm{LS\_b}}$$

$$\mathrm{shadow}_{\mathrm{buffer\_HS\_b\_leaks\_b}} + \mathrm{shadow}_{\mathrm{BSbib}} \xrightarrow{\$shadowleakqmax\$} \mathrm{shadow}_{\mathrm{b}} + \mathrm{shadow}_{\mathrm{gate\_}}$$

$$\mathrm{y} + \mathrm{gate} \xrightleftharpoons{\$qsy\$} \mathrm{buffer} + \mathrm{BSbiy}$$



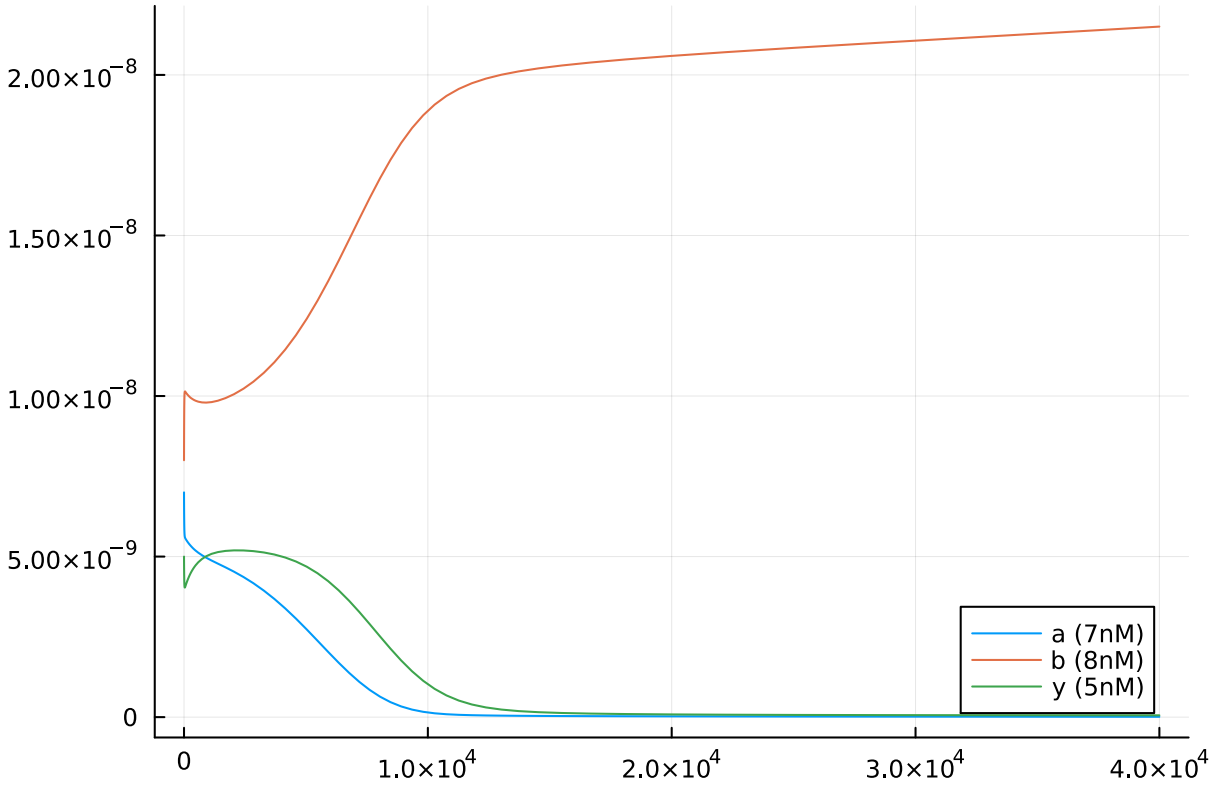
In [23]:
```
print_species_array(consensus_rn)
```

```
species = ['a(t)','gate_L_a_b(t)','buffer_H_a_b_leaks_a(t)','BSbiab(t)','
shadow_a(t)','shadow_gate_L_a_b(t)','shadow_buffer_H_a_b_leaks_a(t)','sha
dow_BSbiab(t)','b(t)','Ibiab(t)','shadow_b(t)','shadow_Ibiab(t)','gate_T_
a_b_leaks_y_y(t)','y(t)','shadow_gate_T_a_b_leaks_y_y(t)','shadow_y(t)','
gate_L_a_y(t)','buffer_H_a_y_leaks_a(t)','BSbiay(t)','shadow_gate_L_a_y(t
)','shadow_buffer_H_a_y_leaks_a(t)','shadow_BSbiay(t)','Ibiay(t)','shadow
_Ibiay(t)','gate_T_a_y_leaks_a_a(t)','shadow_gate_T_a_y_leaks_a_a(t)','ga
te_L_b_y(t)','buffer_H_b_y_leaks_b(t)','BSbiby(t)','shadow_gate_L_b_y(t)'
,'shadow_buffer_H_b_y_leaks_b(t)','shadow_BSbiby(t)','Ibiby(t)','shadow_I
biby(t)','gate_T_b_y_leaks_b_b(t)','shadow_gate_T_b_y_leaks_b_b(t)','gate
_LS_b(t)','buffer_HS_b_leaks_b(t)','BSbib(t)','shadow_gate_LS_b(t)','shad
ow_buffer_HS_b_leaks_b(t)','shadow_BSbib(t)','gate_LS_y(t)','buffer_HS_y_
leaks_y(t)','BSbiy(t)','shadow_gate_LS_y(t)','shadow_buffer_HS_y_leaks_y(
t)','shadow_BSbiy(t)',]
```

In [24]:

```
plot()

qmax = 1e6 # /M sec
k = 1e5 # scaled. 1/(1e3*1e-8)

sigma = 2*k # sigma_a = 2k, sigma_b = k, sigma_y = 0
sigma_a = 2*k
sigma_b = k
sigma_y = 0

gamma_inv = qmax/(qmax-sigma) # 1e6/(8*1e5) = 5/4
q1 = gamma_inv*k
q2 = gamma_inv*k
q3 = gamma_inv*k
qsb = gamma_inv*k
qsy = gamma_inv*2*k

ainit = gamma_inv*7*1e-9
binit = gamma_inv*8*1e-9
yinit = gamma_inv*5*1e-9

Cmax = 100e-9 # 10uM

leak = 1
leak_rate = 1e-13 # M/sec equivalent to 0.36 nM/hr
shadow = 1
# Taken from here: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5739081/
annih = 1e7

u0 = [
ainit, Cmax, 0.0, Cmax, 0.0,
Cmax, 0.0, Cmax, binit, 0.0,
0.0, 0.0, Cmax, yinit, Cmax,
0.0, Cmax, 0.0, Cmax, Cmax,
0.0, Cmax, 0.0, 0.0, Cmax,
Cmax, Cmax, 0.0, Cmax, Cmax,
0.0, Cmax, 0.0, 0.0, Cmax,
Cmax, Cmax, 0.0, Cmax, Cmax,
0.0, Cmax, Cmax, 0.0, Cmax,
Cmax, 0.0, Cmax,
]

p = [q1, q2, q3, qmax, qsb, qsy, leak, leak_rate, shadow, annih]

tspan = (0, 40000)
oprob = ODEProblem(consensus_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)
aplot = aggregate_values(consensus_rn, sol, prefix="a")/gamma_inv
bplot = aggregate_values(consensus_rn, sol, prefix="b")/gamma_inv
yplot = aggregate_values(consensus_rn, sol, prefix="y")/gamma_inv
t = sol.t
plot([t, t, t], [aplot, bplot, yplot], labels=["a (7nM)" "b (8nM)" "y (5n
```

Out[24]:



3. Control

3.1 Ideal

In [25]:

```
control_ideal_rn = @reaction_network begin
    ## Controller
    # Integration term
    cat, ep --> ep + xp
    cat, em --> em + xm

    # Proportional term summation
    kicat, xp --> xp + vp
    kicat, xm --> xm + vm
    kpcat, ep --> ep + vp
    kpcat, em --> em + vm
    deg, vp --> 0
    deg, vm --> 0

    # Error difference
    cat, rp --> rp + ep
    cat, rm --> rm + em
    cat, yp --> yp + em
    cat, ym --> ym + ep
    deg, ep --> 0
    deg, em --> 0

    ## Plant
    produce, vp --> vp + yp
    produce, vm --> vm + ym
    consume, yp --> 0
    consume, ym --> 0
    load, yp + loadp --> loadp
    load, ym + loadm --> loadm

    ## Annihilations
    ann, xp + xm --> 0
    ann, ep + em --> 0
    ann, yp + ym --> 0
    ann, vp + vm --> 0
    ann, rp + rm --> 0
    ann, loadp + loadm --> 0

end cat kicat kpcat deg produce consume ann load
```

Out[25]:

$$
\begin{aligned}
\mathrm{ep} &\xrightarrow{\$cat\$} \mathrm{ep} + \mathrm{xp} \\
\mathrm{em} &\xrightarrow{\$cat\$} \mathrm{em} + \mathrm{xm} \\
\mathrm{xp} &\xrightarrow{\$kicat\$} \mathrm{xp} + \mathrm{vp} \\
\mathrm{xm} &\xrightarrow{\$kicat\$} \mathrm{xm} + \mathrm{vm} \\
\mathrm{ep} &\xrightarrow{\$kpcat\$} \mathrm{ep} + \mathrm{vp} \\
\mathrm{em} &\xrightarrow{\$kpcat\$} \mathrm{em} + \mathrm{vm} \\
\mathrm{vp} &\xrightarrow{\$deg\$} \varnothing \\
\mathrm{vm} &\xrightarrow{\$deg\$} \varnothing \\
\mathrm{rp} &\xrightarrow{\$cat\$} \mathrm{rp} + \mathrm{ep} \\
\mathrm{rm} &\xrightarrow{\$cat\$} \mathrm{rm} + \mathrm{em} \\
\mathrm{yp} &\xrightarrow{\$cat\$} \mathrm{yp} + \mathrm{em} \\
\mathrm{ym} &\xrightarrow{\$cat\$} \mathrm{ym} + \mathrm{ep} \\
\mathrm{ep} &\xrightarrow{\$deg\$} \varnothing \\
\mathrm{em} &\xrightarrow{\$deg\$} \varnothing \\
\mathrm{vp} &\xrightarrow{\$produce\$} \mathrm{vp} + \mathrm{yp} \\
\mathrm{vm} &\xrightarrow{\$produce\$} \mathrm{vm} + \mathrm{ym} \\
\mathrm{yp} &\xrightarrow{\$consume\$} \varnothing \\
\mathrm{ym} &\xrightarrow{\$consume\$} \varnothing \\
\mathrm{yp} + \mathrm{loadp} &\xrightarrow{\$load\$} \mathrm{loadp} \\
\mathrm{ym} + \mathrm{loadm} &\xrightarrow{\$load\$} \mathrm{loadm} \\
\mathrm{xp} + \mathrm{xm} &\xrightarrow{\$ann\$} \varnothing \\
\mathrm{ep} + \mathrm{em} &\xrightarrow{\$ann\$} \varnothing \\
\mathrm{yp} + \mathrm{ym} &\xrightarrow{\$ann\$} \varnothing \\
\mathrm{vp} + \mathrm{vm} &\xrightarrow{\$ann\$} \varnothing \\
\mathrm{rp} + \mathrm{rm} &\xrightarrow{\$ann\$} \varnothing \\
\mathrm{loadp} + \mathrm{loadm} &\xrightarrow{\$ann\$} \varnothing
\end{aligned}
$$

In [26]:
```
reactions(control_ideal_rn)
```

Out[26]:
```
26-element Vector{Reaction}:
 cat, ep --> ep + xp
 cat, em --> em + xm
 kicat, xp --> xp + vp
 kicat, xm --> xm + vm
 kpcat, ep --> ep + vp
 kpcat, em --> em + vm
 deg, vp --> ∅
 deg, vm --> ∅
 cat, rp --> rp + ep
 cat, rm --> rm + em
 cat, yp --> yp + em
 cat, ym --> ym + ep
 deg, ep --> ∅
 deg, em --> ∅
 produce, vp --> vp + yp
 produce, vm --> vm + ym
 consume, yp --> ∅
 consume, ym --> ∅
 load, yp + loadp --> loadp
 load, ym + loadm --> loadm
 ann, xp + xm --> ∅
 ann, ep + em --> ∅
 ann, yp + ym --> ∅
 ann, vp + vm --> ∅
 ann, rp + rm --> ∅
 ann, loadp + loadm --> ∅
```

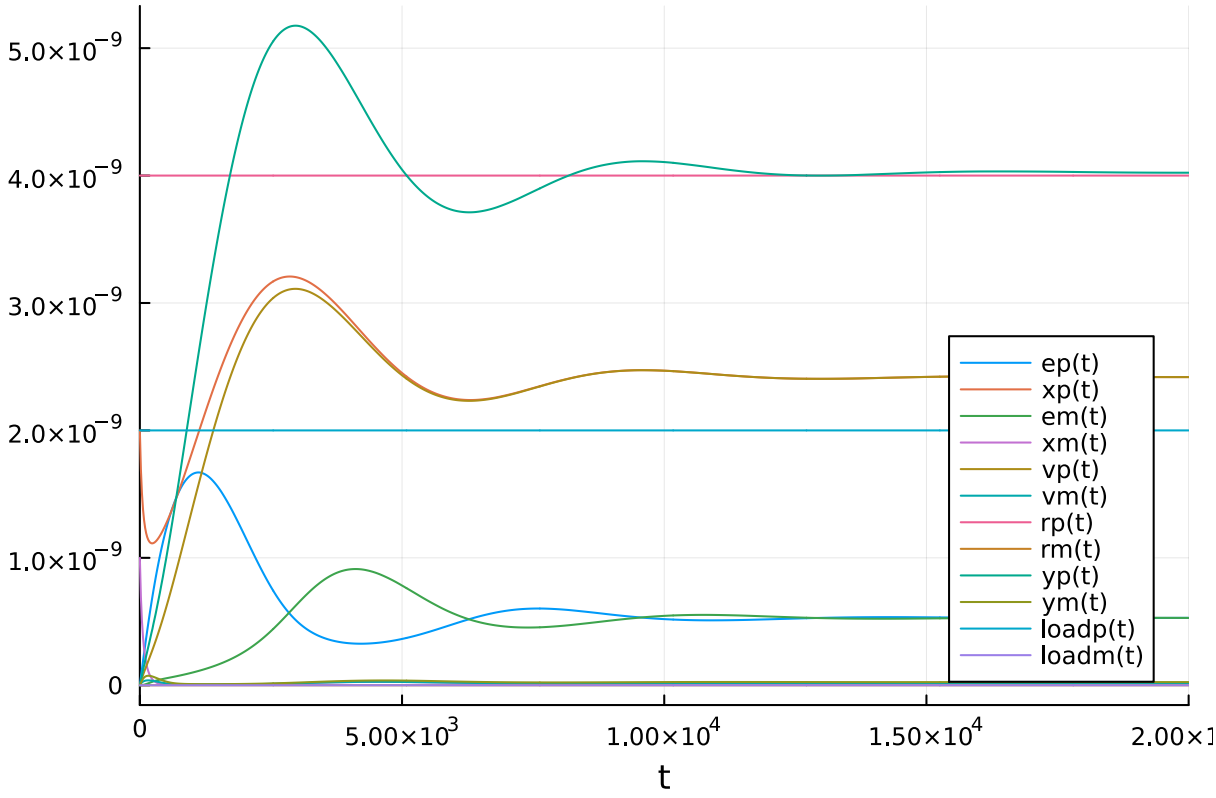
In [27]:
```
print_species_array(control_ideal_rn)
```

```
species = ['ep(t)','xp(t)','em(t)','xm(t)','vp(t)','vm(t)','rp(t)','rm(t)
','yp(t)','ym(t)','loadp(t)','loadm(t)',]
```

In [28]:

```
plot()

ki = 1
kp = 1
deg = 8e-4
cat = 8e-4
kicat = ki*cat
kpcat = kp*cat
ann = 1e7
produce = 0.2
consume = 0.1
load = 1e7

rp = 4e-9
rm = 0
xp = 2e-9
xm = 1e-9
vp = 0
vm = 0
ep = 0
em = 0
yp = 0
ym = 0
loadp = 2e-9
loadm = 0



u0 = [
    ep, xp, em, xm, vp, vm, rp, rm, yp, ym, loadp, loadm
]

p = [cat,kicat, kpcat, deg, produce, consume, ann, load]
tspan = (0, 20000)
oprob = ODEProblem(control_ideal_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)
plot(sol, legend=:bottomright)
```

Out[28]:



3.2 Ideal (From Oishi and Klavins 2011)

In [29]:

```
# control_ideal_rn = @reaction_network begin
#     ## Plant
#     gd1, 0 --> x5p
#     gd1, 0 --> x5m
#     gd2, x5p --> 0
#     gd2, x5m --> 0

#     ## Summation
#     gamma, up --> up + x1p
#     gamma, um --> um + x1m
#     eta, x1p + x1m --> 0

#     ## Weighted integration
#     kI, x1p --> x1p + x4p
#     kI, x1m --> x1m + x4m
#     eta, x4p + x4m --> 0

#     ## Weighted summation
#     gkP, x1p --> x1p + x5p
#     gkP, x1m --> x1m + x5m
#     gamma, x4p --> x4p + x5p
#     gamma, x4m --> x4m + x5m
#     eta, x5p + x5m --> 0
# end gd1 gd2 gamma eta kI gkP
```

In [30]:

```
print_species_array(control_ideal_rn)
```

```
species = ['ep(t)','xp(t)','em(t)','xm(t)','vp(t)','vm(t)','rp(t)','rm(t)
','yp(t)','ym(t)','loadp(t)','loadm(t)',]
```

3.3 Control (Leak = No, Shadow = No)

In [31]:

```
control_rn = @reaction_network begin
    ## Controller
    # Integration term
    cat/Cmax, ep + gate_G__leaks_Iuniep --> Iuniep
    qmax, Iuniep + gate_G__leaks_ep_xp --> ep + xp
    cat/Cmax, em + gate_G__leaks_Iuniem --> Iuniem
    qmax, Iuniem + gate_G__leaks_em_xm --> em + xm

    # Proportional term summation
    kicat/Cmax, xp + gate_G__leaks_Iunixp --> Iunixp
    qmax, Iunixp + gate_G__leaks_xp_vp --> xp + vp
    kicat/Cmax, xm + gate_G__leaks_Iunixm --> Iunixm
    qmax, Iunixm + gate_G__leaks_xm_vm --> xm + vm
    kpcat/Cmax, ep + gate_G__leaks_Iuniep --> Iuniep
    qmax, Iuniep + gate_G__leaks_ep_vp --> ep + vp
    kpcat/Cmax, em + gate_G__leaks_Iuniem --> Iuniem
    qmax, Iuniem + gate_G__leaks_em_vm --> em + vm

    deg, vp --> 0
    deg, vm --> 0

    # Error difference
    cat/Cmax, rp + gate_G__leaks_Iunirp --> Iunirp
    qmax, Iunirp + gate_G__leaks_rp_ep --> rp + ep
    cat/Cmax, rm + gate_G__leaks_Iunirm --> Iunirm
    qmax, Iunirm + gate_G__leaks_rm_em --> rm + em
    cat/Cmax, yp + gate_G__leaks_Iuniyp --> Iuniyp
    qmax, Iuniyp + gate_G__leaks_yp_em --> yp + em
    cat/Cmax, ym + gate_G__leaks_Iuniym --> Iuniym
    qmax, Iuniym + gate_G__leaks_ym_ep --> ym + ep
    deg, ep --> 0
    deg, em --> 0

    ## Plant
    produce, vp --> vp + yp
    produce, vm --> vm + ym
    consume, yp --> 0
    consume, ym --> 0
    load, yp + loadp --> loadp
    load, ym + loadm --> loadm

    ## Annihilations
    ann, xp + xm --> 0
    ann, ep + em --> 0
    ann, yp + ym --> 0
    ann, vp + vm --> 0
    ann, rp + rm --> 0
    ann, loadp + loadm --> 0

end cat kicat kpcat deg produce consume ann load Cmax qmax
species(control_rn)
```

Out[31]:
```
38-element Vector{Term{Real, Base.ImmutableDict{DataType, Any}}}:
 ep(t)
 gate_G__leaks_Iuniep(t)
 Iuniep(t)
 gate_G__leaks_ep_xp(t)
 xp(t)
 em(t)
 gate_G__leaks_Iuniem(t)
 Iuniem(t)
 gate_G__leaks_em_xm(t)
 xm(t)
 gate_G__leaks_Iunixp(t)
 Iunixp(t)
 gate_G__leaks_xp_vp(t)
 ⋮
 Iunirm(t)
 gate_G__leaks_rm_em(t)
 yp(t)
 gate_G__leaks_Iuniyp(t)
 Iuniyp(t)
 gate_G__leaks_yp_em(t)
 ym(t)
 gate_G__leaks_Iuniym(t)
 Iuniym(t)
 gate_G__leaks_ym_ep(t)
 loadp(t)
 loadm(t)
```

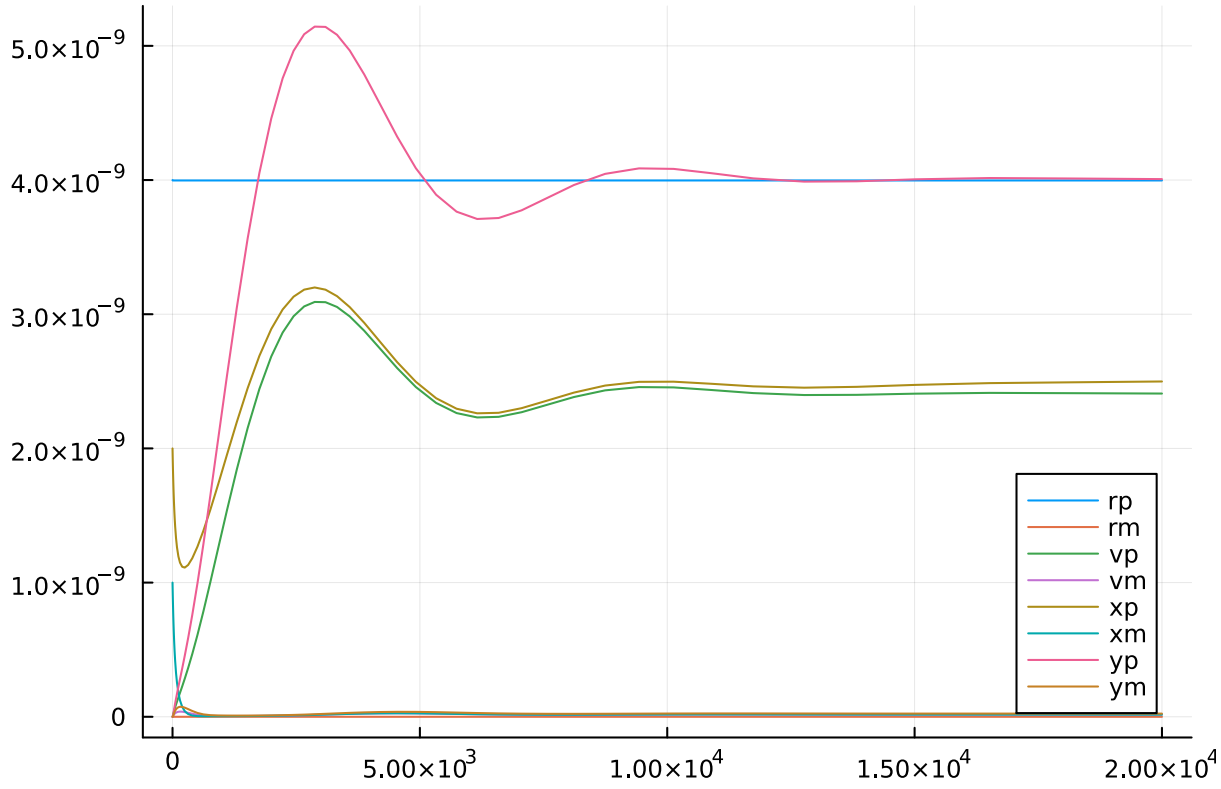
In [32]:
```
print_species_array(control_rn)
```

```
species = ['ep(t)','gate_G__leaks_Iuniep(t)','Iuniep(t)','gate_G__leaks_e
p_xp(t)','xp(t)','em(t)','gate_G__leaks_Iuniem(t)','Iuniem(t)','gate_G__l
eaks_em_xm(t)','xm(t)','gate_G__leaks_Iunixp(t)','Iunixp(t)','gate_G__lea
ks_xp_vp(t)','vp(t)','gate_G__leaks_Iunixm(t)','Iunixm(t)','gate_G__leaks
_xm_vm(t)','vm(t)','gate_G__leaks_ep_vp(t)','gate_G__leaks_em_vm(t)','rp(
t)','gate_G__leaks_Iunirp(t)','Iunirp(t)','gate_G__leaks_rp_ep(t)','rm(t)
','gate_G__leaks_Iunirm(t)','Iunirm(t)','gate_G__leaks_rm_em(t)','yp(t)',
'gate_G__leaks_Iuniyp(t)','Iuniyp(t)','gate_G__leaks_yp_em(t)','ym(t)','g
ate_G__leaks_Iuniym(t)','Iuniym(t)','gate_G__leaks_ym_ep(t)','loadp(t)','
loadm(t)',]
```

In [33]:

```
plot()

ki = 1
kp = 1
deg = 8e-4
cat = 8e-4
kicat = ki*cat
kpcat = kp*cat
ann = 1e7
produce = 0.2
consume = 0.1
load = 1e7

rp = 4e-9
rm = 0
xp = 2e-9
xm = 1e-9
vp = 0
vm = 0
ep = 0
em = 0
yp = 0
ym = 0
loadp = 2e-9
loadm = 0

Cmax = 1000e-9
qmax = 1e6
ccmax = cat/Cmax


u0 = [
ep, Cmax, 0.0, Cmax, xp,
em, Cmax, 0.0, Cmax, xm,
Cmax, 0.0, Cmax, vp, Cmax,
0.0, Cmax, vm, Cmax, Cmax,
rp, Cmax, 0.0, Cmax, rm,
Cmax, 0.0, Cmax, yp, Cmax,
0.0, Cmax, ym, Cmax, 0.0,
Cmax, loadp, loadm,
]

p = [cat, kicat, kpcat, deg, produce, consume, ann, load, Cmax, qmax]
tspan = (0, 20000)
oprob = ODEProblem(control_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)

rp = aggregate_values(control_rn, sol, prefix="rp")
rm = aggregate_values(control_rn, sol, prefix="rm")
vp = aggregate_values(control_rn, sol, prefix="vp")
vm = aggregate_values(control_rn, sol, prefix="vm")
xp = aggregate_values(control_rn, sol, prefix="xp")
xm = aggregate_values(control_rn, sol, prefix="xm")
yp = aggregate_values(control_rn, sol, prefix="yp")
ym = aggregate_values(control_rn, sol, prefix="ym")

t = sol.t
plot([t, t, t, t, t, t, t, t], [rp, rm, vp, vm, xp, xm, yp, ym], label=["
# plot([t, t, t, t], [rp, rm, yp, ym], label=["rp" "rm" "yp" "ym"])
```

Out[33]:



3.3 Control (Leak = Yes, Shadow = No)

In [50]:

```
control_rn = @reaction_network begin
    ## Controller
    # Integration term
    cat/Cmax, ep + gate_G__leaks_Iuniep --> Iuniep
    qmax, Iuniep + gate_G__leaks_ep_xp --> ep + xp
    cat/Cmax, em + gate_G__leaks_Iuniem --> Iuniem
    qmax, Iuniem + gate_G__leaks_em_xm --> em + xm

    # Proportional term summation
    kicat/Cmax, xp + gate_G__leaks_Iunixp --> Iunixp
    qmax, Iunixp + gate_G__leaks_xp_vp --> xp + vp
    kicat/Cmax, xm + gate_G__leaks_Iunixm --> Iunixm
    qmax, Iunixm + gate_G__leaks_xm_vm --> xm + vm
    kpcat/Cmax, ep + gate_G__leaks_Iuniep --> Iuniep
    qmax, Iuniep + gate_G__leaks_ep_vp --> ep + vp
    kpcat/Cmax, em + gate_G__leaks_Iuniem --> Iuniem
    qmax, Iuniem + gate_G__leaks_em_vm --> em + vm

    deg, vp --> 0
    deg, vm --> 0

    # Error difference
    cat/Cmax, rp + gate_G__leaks_Iunirp --> Iunirp
    qmax, Iunirp + gate_G__leaks_rp_ep --> rp + ep
    cat/Cmax, rm + gate_G__leaks_Iunirm --> Iunirm
    qmax, Iunirm + gate_G__leaks_rm_em --> rm + em
    cat/Cmax, yp + gate_G__leaks_Iuniyp --> Iuniyp
    qmax, Iuniyp + gate_G__leaks_yp_em --> yp + em
    cat/Cmax, ym + gate_G__leaks_Iuniym --> Iuniym
    qmax, Iuniym + gate_G__leaks_ym_ep --> ym + ep
    deg, ep --> 0
    deg, em --> 0

    ## Plant
    produce, vp --> vp + yp
    produce, vm --> vm + ym
    consume, yp --> 0
    consume, ym --> 0
    load, yp + loadp --> loadp
    load, ym + loadm --> loadm

    ## Annihilations
    ann, xp + xm --> 0
    ann, ep + em --> 0
    ann, yp + ym --> 0
    ann, vp + vm --> 0
    ann, rp + rm --> 0
    ann, loadp + loadm --> 0

    ## Leak reactions
    leak*leak_rate, 0 --> xp
    leak*leak_rate, 0 --> vp
    leak*leak_rate, 0 --> yp
    leak*leak_rate, 0 --> rp
    leak*leak_rate, 0 --> ep

end cat kicat kpcat deg produce consume ann load Cmax qmax leak leak_rate
species(control_rn)
```

Out[50]:
```
38-element Vector{Term{Real, Base.ImmutableDict{DataType, Any}}}:
 ep(t)
 gate_G__leaks_Iuniep(t)
 Iuniep(t)
 gate_G__leaks_ep_xp(t)
 xp(t)
 em(t)
 gate_G__leaks_Iuniem(t)
 Iuniem(t)
 gate_G__leaks_em_xm(t)
 xm(t)
 gate_G__leaks_Iunixp(t)
 Iunixp(t)
 gate_G__leaks_xp_vp(t)
 ⋮
 Iunirm(t)
 gate_G__leaks_rm_em(t)
 yp(t)
 gate_G__leaks_Iuniyp(t)
 Iuniyp(t)
 gate_G__leaks_yp_em(t)
 ym(t)
 gate_G__leaks_Iuniym(t)
 Iuniym(t)
 gate_G__leaks_ym_ep(t)
 loadp(t)
 loadm(t)
```

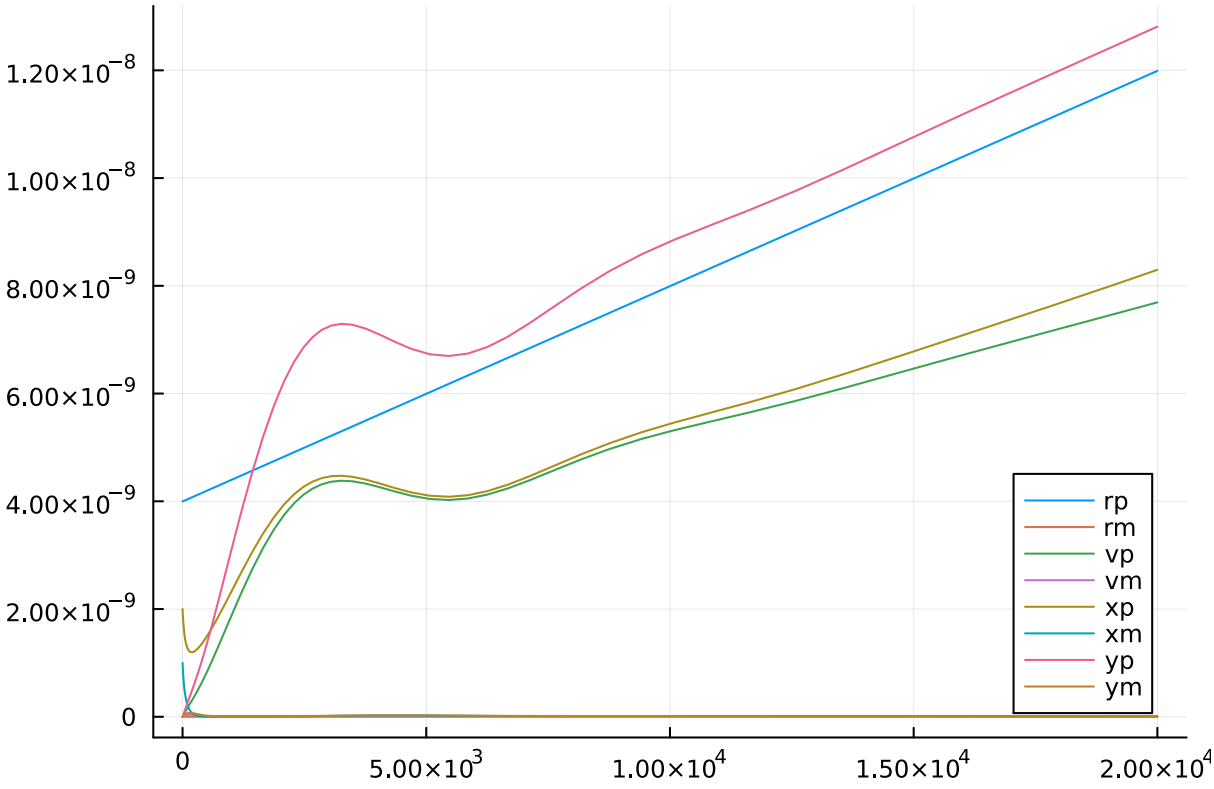
In [51]:
```
print_species_array(control_rn)
```

```
species = ['ep(t)','gate_G__leaks_Iuniep(t)','Iuniep(t)','gate_G__leaks_e
p_xp(t)','xp(t)','em(t)','gate_G__leaks_Iuniem(t)','Iuniem(t)','gate_G__l
eaks_em_xm(t)','xm(t)','gate_G__leaks_Iunixp(t)','Iunixp(t)','gate_G__lea
ks_xp_vp(t)','vp(t)','gate_G__leaks_Iunixm(t)','Iunixm(t)','gate_G__leaks
_xm_vm(t)','vm(t)','gate_G__leaks_ep_vp(t)','gate_G__leaks_em_vm(t)','rp(
t)','gate_G__leaks_Iunirp(t)','Iunirp(t)','gate_G__leaks_rp_ep(t)','rm(t)
','gate_G__leaks_Iunirm(t)','Iunirm(t)','gate_G__leaks_rm_em(t)','yp(t)',
'gate_G__leaks_Iuniyp(t)','Iuniyp(t)','gate_G__leaks_yp_em(t)','ym(t)','g
ate_G__leaks_Iuniym(t)','Iuniym(t)','gate_G__leaks_ym_ep(t)','loadp(t)','
loadm(t)',]
```

In [52]:

```
plot()

ki = 1
kp = 1
deg = 8e-4
cat = 8e-4
kicat = ki*cat
kpcat = kp*cat
ann = 1e7
produce = 0.2
consume = 0.1
load = 1e7

rp = 4e-9
rm = 0
xp = 2e-9
xm = 1e-9
vp = 0
vm = 0
ep = 0
em = 0
yp = 0
ym = 0
loadp = 2e-9
loadm = 0

Cmax = 1000e-9
qmax = 1e6
leak = 1
leak_rate = 4e-13

u0 = [
ep, Cmax, 0.0, Cmax, xp,
em, Cmax, 0.0, Cmax, xm,
Cmax, 0.0, Cmax, vp, Cmax,
0.0, Cmax, vm, Cmax, Cmax,
rp, Cmax, 0.0, Cmax, rm,
Cmax, 0.0, Cmax, yp, Cmax,
0.0, Cmax, ym, Cmax, 0.0,
Cmax, loadp, loadm,
]

p = [cat, kicat, kpcat, deg, produce, consume, ann, load, Cmax, qmax, lea
tspan = (0, 20000)
oprob = ODEProblem(control_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)

rp = aggregate_values(control_rn, sol, prefix="rp")
rm = aggregate_values(control_rn, sol, prefix="rm")
vp = aggregate_values(control_rn, sol, prefix="vp")
vm = aggregate_values(control_rn, sol, prefix="vm")
xp = aggregate_values(control_rn, sol, prefix="xp")
xm = aggregate_values(control_rn, sol, prefix="xm")
yp = aggregate_values(control_rn, sol, prefix="yp")
ym = aggregate_values(control_rn, sol, prefix="ym")

t = sol.t
plot([t, t, t, t, t, t, t, t], [rp, rm, vp, vm, xp, xm, yp, ym], label=["
# plot([t, t, t, t], [rp, rm, yp, ym], label=["rp" "rm" "yp" "ym"])
```

Out[52]:



3.4 Control (Leak = Yes, Shadow = Yes)

In [57]:

```
control_rn = @reaction_network begin
    ## Controller
    # Integration term
    cat/Cmax, ep + gate_G__leaks_Iuniep --> Iuniep
    qmax, Iuniep + gate_G__leaks_ep_xp --> ep + xp
    cat/Cmax, em + gate_G__leaks_Iuniem --> Iuniem
    qmax, Iuniem + gate_G__leaks_em_xm --> em + xm
    # --
    shadow*cat/Cmax, shadow_ep + gate_G__leaks_Iuniep --> shadow_Iuniep
    shadow*qmax, shadow_Iuniep + shadow_gate_G__leaks_ep_xp --> shadow_ep
    shadow*cat/Cmax, shadow_em + shadow_gate_G__leaks_Iuniem --> shadow_I
    shadow*qmax, shadow_Iuniem + shadow_gate_G__leaks_em_xm --> shadow_em

    # Proportional term summation
    kicat/Cmax, xp + gate_G__leaks_Iunixp --> Iunixp
    qmax, Iunixp + gate_G__leaks_xp_vp --> xp + vp
    kicat/Cmax, xm + gate_G__leaks_Iunixm --> Iunixm
    qmax, Iunixm + gate_G__leaks_xm_vm --> xm + vm
    kpcat/Cmax, ep + gate_G__leaks_Iuniep --> Iuniep
    qmax, Iuniep + gate_G__leaks_ep_vp --> ep + vp
    kpcat/Cmax, em + gate_G__leaks_Iuniem --> Iuniem
    qmax, Iuniem + gate_G__leaks_em_vm --> em + vm
    deg, vp --> 0
    deg, vm --> 0

    # --

    shadow*kicat/Cmax, shadow_xp + shadow_gate_G__leaks_Iunixp --> shadow
    shadow*qmax, shadow_Iunixp + shadow_gate_G__leaks_xp_vp --> shadow_xp
    shadow*kicat/Cmax, shadow_xm + shadow_gate_G__leaks_Iunixm --> shadow
    shadow*qmax, shadow_Iunixm + shadow_gate_G__leaks_xm_vm --> shadow_xm
    shadow*kpcat/Cmax, shadow_ep + shadow_gate_G__leaks_Iuniep --> shadow
```

```
    shadow*qmax, shadow_Iuniep + shadow_gate_G__leaks_ep_vp --> shadow_ep
    shadow*kpcat/Cmax, shadow_em + shadow_gate_G__leaks_Iuniem --> shadow
    shadow*qmax, shadow_Iuniem + shadow_gate_G__leaks_em_vm --> shadow_em
    shadow*deg, shadow_vp --> 0
    shadow*deg, shadow_vm --> 0

    # Error difference
    cat/Cmax, rp + gate_G__leaks_Iunirp --> Iunirp
    qmax, Iunirp + gate_G__leaks_rp_ep --> rp + ep
    cat/Cmax, rm + gate_G__leaks_Iunirm --> Iunirm
    qmax, Iunirm + gate_G__leaks_rm_em --> rm + em
    cat/Cmax, yp + gate_G__leaks_Iuniyp --> Iuniyp
    qmax, Iuniyp + gate_G__leaks_yp_em --> yp + em
    cat/Cmax, ym + gate_G__leaks_Iuniym --> Iuniym
    qmax, Iuniym + gate_G__leaks_ym_ep --> ym + ep
    deg, ep --> 0
    deg, em --> 0
    # --
    shadow*cat/Cmax, shadow_rp + shadow_gate_G__leaks_Iunirp --> shadow_I
    shadow*qmax, shadow_Iunirp + shadow_gate_G__leaks_rp_ep --> shadow_rp
    shadow*cat/Cmax, shadow_rm + shadow_gate_G__leaks_Iunirm --> shadow_I
    shadow*qmax, shadow_Iunirm + shadow_gate_G__leaks_rm_em --> shadow_rm
    shadow*cat/Cmax, shadow_yp + shadow_gate_G__leaks_Iuniyp --> shadow_I
    shadow*qmax, shadow_Iuniyp + shadow_gate_G__leaks_yp_em --> shadow_yp
    shadow*cat/Cmax, shadow_ym + shadow_gate_G__leaks_Iuniym --> shadow_I
    shadow*qmax, shadow_Iuniym + shadow_gate_G__leaks_ym_ep --> shadow_ym
    shadow*deg, shadow_ep --> 0
    shadow*deg, shadow_em --> 0

    ## Plant
    produce, vp --> vp + yp
    produce, vm --> vm + ym
    consume, yp --> 0
    consume, ym --> 0
    load, yp + loadp --> loadp
    load, ym + loadm --> loadm
    # --
    shadow*produce, shadow_vp --> shadow_vp + shadow_yp
    shadow*produce, shadow_vm --> shadow_vm + shadow_ym
    shadow*consume, shadow_yp --> 0
    shadow*consume, shadow_ym --> 0
    shadow*load, shadow_yp + shadow_loadp --> shadow_loadp
    shadow*load, shadow_ym + shadow_loadm --> shadow_loadm

    ## Annihilations
    ann, xp + xm --> 0
    ann, ep + em --> 0
    ann, yp + ym --> 0
    ann, vp + vm --> 0
    ann, rp + rm --> 0
    ann, loadp + loadm --> 0
    # --
    shadow*ann, shadow_xp + shadow_xm --> 0
    shadow*ann, shadow_ep + shadow_em --> 0
    shadow*ann, shadow_yp + shadow_ym --> 0
    shadow*ann, shadow_vp + shadow_vm --> 0
    shadow*ann, shadow_rp + shadow_rm --> 0
    shadow*ann, shadow_loadp + shadow_loadm --> 0

    ## Leak reactions
    leak*leak_rate, 0 --> xp
```

```
    leak*leak_rate, 0 --> vp
    leak*leak_rate, 0 --> yp
    leak*leak_rate, 0 --> rp
    leak*leak_rate, 0 --> ep
    # --
    shadow*leak*leak_rate, 0 --> shadow_xp
    shadow*leak*leak_rate, 0 --> shadow_vp
    shadow*leak*leak_rate, 0 --> shadow_yp
    shadow*leak*leak_rate, 0 --> shadow_rp
    shadow*leak*leak_rate, 0 --> shadow_ep

    ## Leak cancellation reactions
    shadow*leak*ann, xp + shadow_xp --> 0
    shadow*leak*ann, xm + shadow_xm --> 0
    shadow*leak*ann, vp + shadow_vp --> 0
    shadow*leak*ann, vm + shadow_vm --> 0
    shadow*leak*ann, yp + shadow_yp --> 0
    shadow*leak*ann, ym + shadow_ym --> 0
    shadow*leak*ann, rp + shadow_rp --> 0
    shadow*leak*ann, rm + shadow_rm --> 0
    shadow*leak*ann, ep + shadow_ep --> 0
    shadow*leak*ann, em + shadow_em --> 0
    shadow*leak*ann, loadp + shadow_loadp --> 0
    shadow*leak*ann, loadm + shadow_loadm --> 0

end cat kicat kpcat deg produce consume ann load Cmax qmax leak leak_rate
species(control_rn)
```

Out[57]:

```
76-element Vector{Term{Real, Base.ImmutableDict{DataType, Any}}}:
 ep(t)
 gate_G__leaks_Iuniep(t)
 Iuniep(t)
 gate_G__leaks_ep_xp(t)
 xp(t)
 em(t)
 gate_G__leaks_Iuniem(t)
 Iuniem(t)
 gate_G__leaks_em_xm(t)
 xm(t)
 shadow_ep(t)
 shadow_Iuniep(t)
 shadow_gate_G__leaks_ep_xp(t)
 ⋮
 shadow_yp(t)
 shadow_gate_G__leaks_Iuniyp(t)
 shadow_Iuniyp(t)
 shadow_gate_G__leaks_yp_em(t)
 shadow_ym(t)
 shadow_gate_G__leaks_Iuniym(t)
 shadow_Iuniym(t)
 shadow_gate_G__leaks_ym_ep(t)
 loadp(t)
 loadm(t)
 shadow_loadp(t)
 shadow_loadm(t)
```

In [59]:

```
print_species_array(control_rn)
```

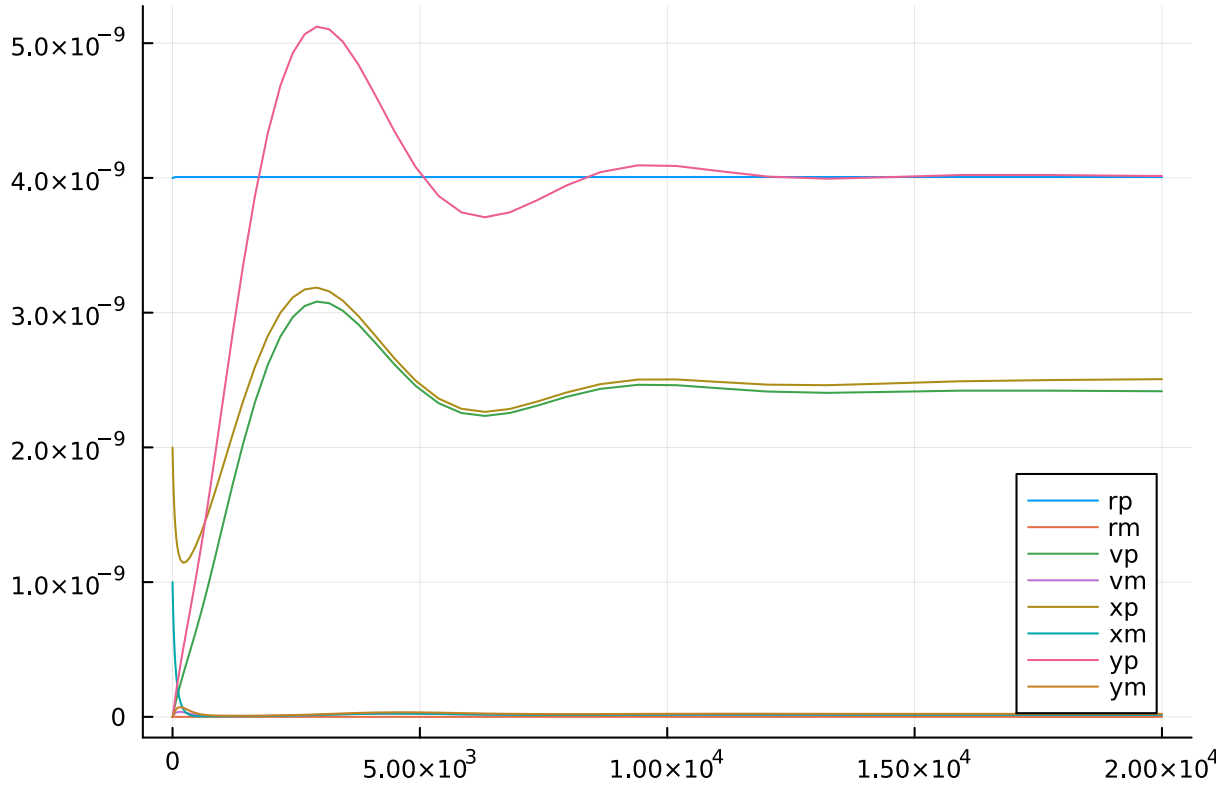
```
species = ['ep(t)','gate_G__leaks_Iuniep(t)','Iuniep(t)','gate_G__leaks_e
p_xp(t)','xp(t)','em(t)','gate_G__leaks_Iuniem(t)','Iuniem(t)','gate_G__l
eaks_em_xm(t)','xm(t)','shadow_ep(t)','shadow_Iuniep(t)','shadow_gate_G__
leaks_ep_xp(t)','shadow_xp(t)','shadow_em(t)','shadow_gate_G__leaks_Iunie
m(t)','shadow_Iuniem(t)','shadow_gate_G__leaks_em_xm(t)','shadow_xm(t)','
gate_G__leaks_Iunixp(t)','Iunixp(t)','gate_G__leaks_xp_vp(t)','vp(t)','ga
te_G__leaks_Iunixm(t)','Iunixm(t)','gate_G__leaks_xm_vm(t)','vm(t)','gate
_G__leaks_ep_vp(t)','gate_G__leaks_em_vm(t)','shadow_gate_G__leaks_Iunixp
(t)','shadow_Iunixp(t)','shadow_gate_G__leaks_xp_vp(t)','shadow_vp(t)','s
hadow_gate_G__leaks_Iunixm(t)','shadow_Iunixm(t)','shadow_gate_G__leaks_x
m_vm(t)','shadow_vm(t)','shadow_gate_G__leaks_Iuniep(t)','shadow_gate_G__
leaks_ep_vp(t)','shadow_gate_G__leaks_em_vm(t)','rp(t)','gate_G__leaks_Iu
nirp(t)','Iunirp(t)','gate_G__leaks_rp_ep(t)','rm(t)','gate_G__leaks_Iuni
rm(t)','Iunirm(t)','gate_G__leaks_rm_em(t)','yp(t)','gate_G__leaks_Iuniyp
(t)','Iuniyp(t)','gate_G__leaks_yp_em(t)','ym(t)','gate_G__leaks_Iuniym(t
)','Iuniym(t)','gate_G__leaks_ym_ep(t)','shadow_rp(t)','shadow_gate_G__le
aks_Iunirp(t)','shadow_Iunirp(t)','shadow_gate_G__leaks_rp_ep(t)','shadow
_rm(t)','shadow_gate_G__leaks_Iunirm(t)','shadow_Iunirm(t)','shadow_gate_
G__leaks_rm_em(t)','shadow_yp(t)','shadow_gate_G__leaks_Iuniyp(t)','shado
w_Iuniyp(t)','shadow_gate_G__leaks_yp_em(t)','shadow_ym(t)','shadow_gate_
G__leaks_Iuniym(t)','shadow_Iuniym(t)','shadow_gate_G__leaks_ym_ep(t)','l
oadp(t)','loadm(t)','shadow_loadp(t)','shadow_loadm(t)',]
```

In [62]:

```
plot()

ki = 1
kp = 1
deg = 8e-4
cat = 8e-4
kicat = ki*cat
kpcat = kp*cat
ann = 1e7
produce = 0.2
consume = 0.1
load = 1e7

rp = 4e-9
rm = 0
xp = 2e-9
xm = 1e-9
vp = 0
vm = 0
ep = 0
em = 0
yp = 0
ym = 0
loadp = 2e-9
loadm = 0

Cmax = 1000e-9
qmax = 1e6
leak = 1
leak_rate = 4e-13
shadow = 1


u0 = [
ep, Cmax, 0.0, Cmax, xp,
em, Cmax, 0.0, Cmax, xm,
0.0, 0.0, Cmax, 0.0, 0.0,
Cmax, 0.0, Cmax, 0.0, Cmax,
```

```
0.0, Cmax, vp, Cmax, 0.0,
Cmax, vm, Cmax, Cmax, Cmax,
0.0, Cmax, 0.0, Cmax, 0.0,
Cmax, 0.0, Cmax, Cmax, Cmax,
rp, Cmax, 0.0, Cmax, rm,
Cmax, 0.0, Cmax, yp, Cmax,
0.0, Cmax, ym, Cmax, 0.0,
Cmax, 0.0, Cmax, 0.0, Cmax,
0.0, Cmax, 0.0, Cmax, 0.0,
Cmax, 0.0, Cmax, 0.0, Cmax,
0.0, Cmax, loadp, loadm, 0.0,
0.0,
]

p = [cat, kicat, kpcat, deg, produce, consume, ann, load, Cmax, qmax, lea
tspan = (0, 20000)
oprob = ODEProblem(control_rn, u0, tspan, p)
sol = solve(oprob, AutoTsit5(Rosenbrock23()), reltol=1e-12, abstol=1e-12)

rp = aggregate_values(control_rn, sol, prefix="rp")
rm = aggregate_values(control_rn, sol, prefix="rm")
vp = aggregate_values(control_rn, sol, prefix="vp")
vm = aggregate_values(control_rn, sol, prefix="vm")
xp = aggregate_values(control_rn, sol, prefix="xp")
xm = aggregate_values(control_rn, sol, prefix="xm")
yp = aggregate_values(control_rn, sol, prefix="yp")
ym = aggregate_values(control_rn, sol, prefix="ym")

t = sol.t
plot([t, t, t, t, t, t, t, t], [rp, rm, vp, vm, xp, xm, yp, ym], label=["
# plot([t, t, t, t], [rp, rm, yp, ym], label=["rp" "rm" "yp" "ym"])
```

Out[62]:



In [ ]: