## In the name of Allah

## Project 4:

## Hossein Alizadeh Piralidehi 810100522

### Problem considerations:

$c_t \geq 0$ denotes the amount that the investor commits in period *t.*

$p_t \geq 0$ denotes the amount that the investor pays in to the investment in response to capital calls in period *t*.

$d_t \geq 0$ denotes the amount that the investor receives in distributions from the investment in period *t*.

$n_t \geq 0$ denotes the net asset value (NAV) of the investment in period *t*.

$u_t \geq 0$ denotes the total amount of uncalled commitments, i.e., the difference between the total so far committed and the total so far that has been called (and paid into the investment).

dynamic of these varables are given:

$$n_{t+1} = (1+r)n_t + p_t - d_t \qquad u_{t+1} = u_t - p_t + c_t \qquad t = 1, 2, \ldots, T$$

$$u_1 = n_1 = 0$$

$$p_t = \gamma^{\text{call}} u_t \qquad d_t = \gamma^{\text{dist}} n_t$$

commitements and capital calls are limited:

$$c_t \leq c_{\max} \quad p_t \leq p_{\max} \quad t = 1, 2, \ldots, T$$

we have a total budget B:

$$1^T c \leq B$$

problem objective is to minimize:

$$f_0(c) = \frac{1}{T+1} \sum_{t=1}^{T+1} (n_t - n_{\text{des}})^2 + \frac{\lambda}{T-1} \sum_{t=1}^{T-1} (c_{t+1} - c_t)^2$$

### Description:

given optimization problem is:

$$\min \frac{1}{T+1} \sum_{t=1}^{T+1} (n_t - n_{\text{des}})^2 + \frac{\lambda}{T-1} \sum_{t=1}^{T-1} (c_{t+1} - c_t)^2$$

$s.t.$

$$c_t \le c_{\max} \qquad t = 1,2,\dots,T$$
$$c_t \ge 0 \qquad t = 1,2,\dots,T$$
$$1^T c \le B$$
$$p_t \le p_{\max} \qquad t = 1,2,\dots,T$$
$$p_t \ge 0 \qquad t = 1,2,\dots,T$$
$$n_{t+1} = (1+r)n_t + p_t - d_t \quad t = 1,2,\dots,T$$
$$u_{t+1} = u_t - p_t + c_t \qquad t = 1,2,\dots,T$$
$$u_1 = n_1 = 0$$
$$p_t = \gamma^{\text{call}} u_t \qquad d_t = \gamma^{\text{dist}} n_t$$

at first we try to reduce number of variables and constraints of problem.

**using formulas for $u_{t+1}$ :**

$$\beta = 1 - \gamma^{\text{call}}$$
$$u_1 = 0$$
$$u_{t+1} = u_t - \gamma^{\text{call}} u_t + c_t = \beta u_t + c_t$$
$$u_2 = c_1$$
$$u_3 = c_2 + \beta\, c_1$$
$$u_4 = c_3 + \beta c_2 + \beta^2 c_1$$
$$u_5 = c_4 + \beta c_3 + \beta^2 c_2 + \beta^3 c_1$$
$$\dots$$
$$u_{t+1} = c_t + \beta c_{t-1} + \ \dots\ + \beta^{t-2} c_2 + \beta^{t-1} c_1$$

**using formula for $n_{t+1}$:**

$$\alpha = 1 + r - \gamma^{\text{dist}}$$

$$n_1 = 0$$
$$n_{t+1} = (1+r)n_t + \gamma^{\text{call}} u_t - \gamma^{\text{dist}} n_t = \alpha\, n_t + \gamma^{\text{call}} u_t$$
$$n_2 = 0$$
$$n_3 = \gamma^{\text{call}} u_2$$
$$n_4 = \gamma^{\text{call}} (u_3 + \alpha\, u_2)$$
$$n_5 = \gamma^{\text{call}} \left(u_4 + \alpha\, u_3 + \alpha^2\, u_2\right)$$
$$\dots$$
$$n_{t+1} = \gamma^{\text{call}} \left(u_t + \alpha u_{t-1} + \ \dots\ + \alpha^{t-2} u_2\right)$$

now we replace calculated $u_t$ into $n_{t+1}$ formula:

$n_1 = 0$

$n_2 = 0$

$n_3 = \gamma^{\text{call}} c_1$

$n_4 = \gamma^{\text{call}} [c_2 + (\beta + \alpha) c_1]$

$n_5 = \gamma^{\text{call}} \left[c_3 + (\beta + \alpha) c_2 + (\beta^2 + \alpha\beta + \alpha^2) c_1\right]$

$\ldots$

$n_{t+1} = \gamma^{\text{call}} \left[c_{t-1} + (\beta + \alpha) c_{t-2} + (\beta^2 + \alpha\beta + \alpha^2) c_{t-3} + \ldots + (\beta^{t-2} + \beta^{t-3}\alpha + \ldots + \beta\alpha^{t-3} + \alpha^{t-2}) c_1\right]$

finally the oprimization problem will be rewritten with:

$$\begin{aligned}
&\min f_0(c) \\
&s.t. \\
&\quad n_{t+1} = \gamma^{\text{call}} \left[c_{t-1} + (\beta + \alpha) c_{t-2} + (\beta^2 + \alpha\beta + \alpha^2) c_{t-3} + \ldots + (\beta^{t-2} + \beta^{t-3}\alpha + \ldots + \beta\alpha^{t-3} + \alpha^{t-2}) c_1\right] \quad t = 2, \ldots, T \\
&\quad n_1 = n_2 = 0 \\
&\quad n_t \geq 0 \quad t = 3, \ldots, T \\
&\quad 0 \leq u_t \leq \frac{p_{\max}}{\gamma^{\text{call}}} \quad t = 1, 2, \ldots, T \\
&\quad 0 \leq c_t \leq c_{\max} \quad t = 1, 2, \ldots, T \\
&\quad 1^T c \leq B
\end{aligned}$$

all of constraints are affine in $c_t$ so all of them are convex. objective is quadratic form ($c^T \mathrm{Q} c$) which $Q$ is PD, so objective is convex, finally there is convex optimization problem which can be solved using cvx toolbox.

### Simulation with given parameters:

## Part a)

load parameters

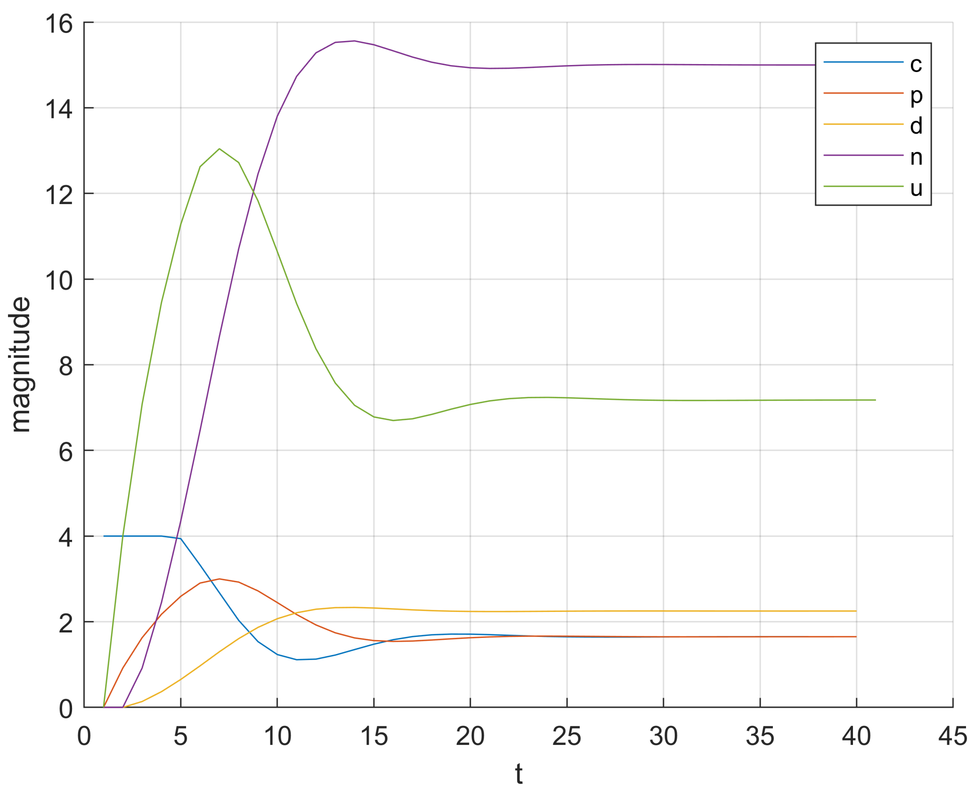
```
clc;clear;close all;

T = 40; r = 0.04;
gamma_call = 0.23; gamma_dist = 0.15;
c_max = 4; p_max = 3; B = 85;
n_des = 15; lambda = 5;

alpha = 1 + r - gamma_dist;
beta = 1 - gamma_call;
```

solve convex problem using cvx toolbox

```
cvx_begin
    variables c(T);
        u(2) = c(1);
        for t = 2 : T
            u(t+1) = (1-gamma_call).^(0:t-1)*c(t:-1:1);
            for i = 1 : t-1
                coeff(i) = sum(alpha.^(0:i-1) .* beta.^(i-1:-1:0));
            end
            n(t+1) = gamma_call*( coeff *c(t-1:-1:1));
        end
    minimize( 1/(T+1) * sum((n-n_des).^2) + lambda/(T-1) * sum( (c(2:end)-c(1:end-1)) .^2 ) );
    subject to
        n(1) == 0;
        n(2) == 0;
        n(:) >= 0;

        u(:) >= 0;
        u(:) <= p_max/gamma_call;

        c(:) >= 0;
        c(:) <= c_max;
        ones(1, T)*c <= B;
cvx_end
```

```
Calling SDPT3 4.0: 435 variables, 118 equality constraints
   For improved efficiency, SDPT3 is solving the dual problem.
------------------------------------------------------------

 num. of constraints = 118
 dim. of sdp    var  = 156,   num. of sdp  blk  = 78
 dim. of linear var  = 201
*******************************************************************
   SDPT3: Infeasible path-following algorithms
*******************************************************************
 version  predcorr  gam  expon  scale_data
   HKM      1      0.000   1        0
it pstep dstep pinfeas dinfeas  gap      prim-obj      dual-obj    cputime
-------------------------------------------------------------------
 0|0.000|0.000|3.4e+02|9.4e+00|4.2e+05| 1.325268e+04  0.000000e+00| 0:0:00| chol  1  1
 1|0.835|0.863|5.6e+01|1.3e+00|7.1e+04| 1.619579e+04 -8.711582e+01| 0:0:00| chol  1  1
 2|0.777|0.826|1.2e+01|2.3e-01|3.1e+04| 1.741752e+04 -4.578164e+02| 0:0:00| chol  1  1
 3|0.996|1.000|4.9e-02|1.0e-04|9.0e+03| 8.293383e+03 -7.291431e+02| 0:0:00| chol  1  1
 4|0.962|0.987|1.9e-03|9.8e-03|3.8e+02| 3.227465e+02 -3.598174e+01| 0:0:00| chol  1  1
 5|0.928|0.920|1.4e-04|1.2e-03|4.1e+01| 1.816677e+01 -2.189790e+01| 0:0:00| chol  1  1
 6|0.757|1.000|3.3e-05|2.7e-05|2.4e+01| 4.083727e+00 -1.945739e+01| 0:0:00| chol  1  1
 7|0.948|0.908|1.7e-06|9.1e-06|1.9e+00|-1.365798e+01 -1.560012e+01| 0:0:00| chol  1  1
 8|1.000|0.966|1.2e-10|6.6e-07|3.4e-01|-1.482938e+01 -1.516815e+01| 0:0:00| chol  1  1
 9|0.960|0.950|2.3e-11|3.3e-08|2.5e-02|-1.506112e+01 -1.508651e+01| 0:0:00| chol  1  1
10|0.979|0.969|4.6e-13|1.0e-09|6.8e-04|-1.507914e+01 -1.507982e+01| 0:0:00| chol  1  1
11|0.967|0.982|1.5e-14|2.0e-11|2.0e-05|-1.507957e+01 -1.507959e+01| 0:0:00| chol  1  1
12|1.000|1.000|3.9e-14|1.0e-12|3.3e-06|-1.507958e+01 -1.507958e+01| 0:0:00| chol  1  1
13|1.000|1.000|8.1e-15|1.0e-12|9.2e-08|-1.507958e+01 -1.507958e+01| 0:0:00|
  stop: max(relative gap, infeasibilities) < 1.49e-08
-------------------------------------------------------------------
 number of iterations   = 13
 primal objective value = -1.50795808e+01
 dual   objective value = -1.50795809e+01
 gap := trace(XZ)       = 9.21e-08
 relative gap           = 2.96e-09
 actual relative gap    = 2.95e-09
 rel. primal infeas (scaled problem)   = 8.05e-15
 rel. dual     "        "       "      = 1.00e-12
 rel. primal infeas (unscaled problem) = 0.00e+00
 rel. dual     "         "        "       = 0.00e+00
 norm(X), norm(y), norm(Z) = 7.4e+00, 2.9e+02, 3.1e+02
 norm(A), norm(b), norm(C) = 3.0e+01, 1.8e+00, 1.8e+02
 Total CPU time (secs)  = 0.26
 CPU time per iteration = 0.02
 termination code       =  0
 DIMACS: 1.3e-14  0.0e+00  2.1e-12  0.0e+00  2.9e-09  3.0e-09
-------------------------------------------------------------------

------------------------------------------------------------
Status: Solved
Optimal value (cvx_optval): +26.0552
```

```
p(1:T) = gamma_call*u(1:T);
d(1:T) = gamma_dist*n(1:T);
figure
hold on
grid on
plot(c)
plot(p)
plot(d)
plot(n)
plot(u)
xlabel('t')
ylabel('magnitude')
legend('c', 'p', 'd', 'n', 'u')
```



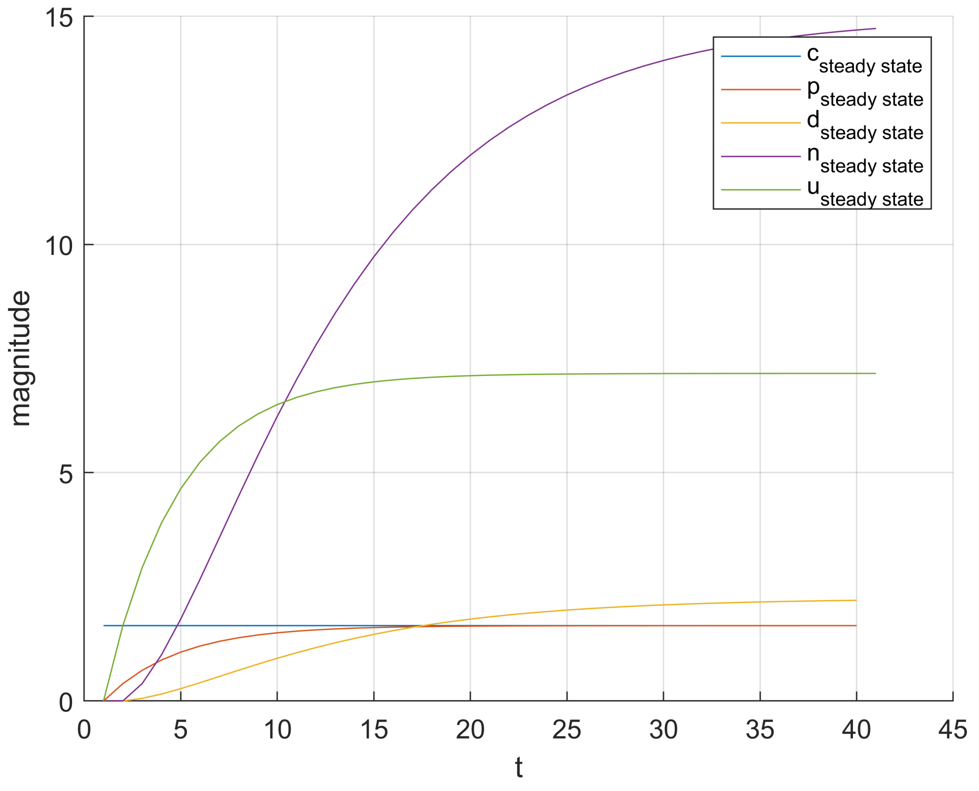
```
disp("Objective value: " +  (1/(T+1) * sum((n-n_des).^2) + lambda/(T-1) * sum( (c(2:end)-c(1:en
Objective value: 26.0552
```

```
disp("Tracking error value: " +  1/(T+1) * sum((n-n_des).^2))
```

```
Tracking error value: 25.8462
```

## Part b)

```
c_ss = (gamma_dist-r)*n_des*ones(T,1);
u_ss(2:T+1) = (gamma_dist-r)*n_des/(1-beta)*(1-beta.^(1:T));
n_ss = zeros(T+1,1);
for t = 2 : T
    coeff_ss = [];
    for i = 1 : t-1
        coeff_ss(i) = sum(alpha.^(0:i-1) .* beta.^(i-1:-1:0));
    end
    n_ss(t+1) = gamma_call*( coeff_ss *c_ss(t-1:-1:1));
end
p_ss(1:T) = gamma_call*u_ss(1:T);
d_ss(1:T) = gamma_dist*n_ss(1:T);

figure
hold on
grid on
plot(c_ss)
plot(p_ss)
plot(d_ss)
plot(n_ss)
plot(u_ss)
xlabel('t')
ylabel('magnitude')
legend('c_{steady state}', 'p_{steady state}', 'd_{steady state}', 'n_{steady state}', 'u_{stea
```



```
disp("Steady state objective value: " +  (1/(T+1) * sum((n_ss-n_des).^2) + lambda/(T-1) * sum(
```

```
Steady state objective value: 47.0143
```

```
disp("Steady state tracking error value: " +  1/(T+1) * sum((n_ss-n_des).^2))
```

```
Steady state tracking error value: 47.0143
```

If we compare final values of optimal solution and steady-state solution, we see that they converge to equal values.

When we use steady state values instead of optimal, regularity term in objective will be zero because "c_ss" is constant for all of values, but mean square tracking error became larger than before and this is beacause we dont choose set of "c" values to make it small as possible.