Math 584 - Homework 5

Diane PERES

In [1]:
```
import numpy as np
import os
import matplotlib.pyplot as plt
import scipy.io as sio  # for loading matlab data
from scipy import stats
from statsmodels.tsa import stattools
from statsmodels.regression import linear_model
import warnings
warnings.filterwarnings('ignore')# get warning for complex type
import time
```

MSFT

In [2]:
```
dataDir = "./tickers/"
mats_MSFT = []
indices = ["03","04","05","06","07","10","11","12","13","14","17","18","19","20","21","24","25","26"]
for i in indices:## MSFT
    mats_MSFT.append( sio.loadmat( dataDir+"MSFT_201411"+i+".mat" ) )
    #print(dataDir+"MSFT_2014110"+i+".mat" )
```

Exercice 1

In this exercise, you will compute the **execution costs** produced by the **TWAP** execution algorithm, using the ticker MSFT from Nov 3 through Nov 26, 2014, 10am–3:30pm, as your sample. In the test, use:

- κ **temporary** impact: $\kappa = 0.005$
- S midprice as the **unaffected** price
- 10 minutes as the size of the estimation window
- 3 minutes as the target execution time
- 1 second as the trading frequency (e.g., $t = 0, 1, 2, \ldots$ denotes the number of seconds)

- λ **permanent** price impact coefficient. The permanent price impact coefficient λ is estimated (using the least-squares linear regression) from the reduced frequency (i.e., using one-second frequency) sample via a linear (least-squares) regression of midprice increments on the increments of the order flow.

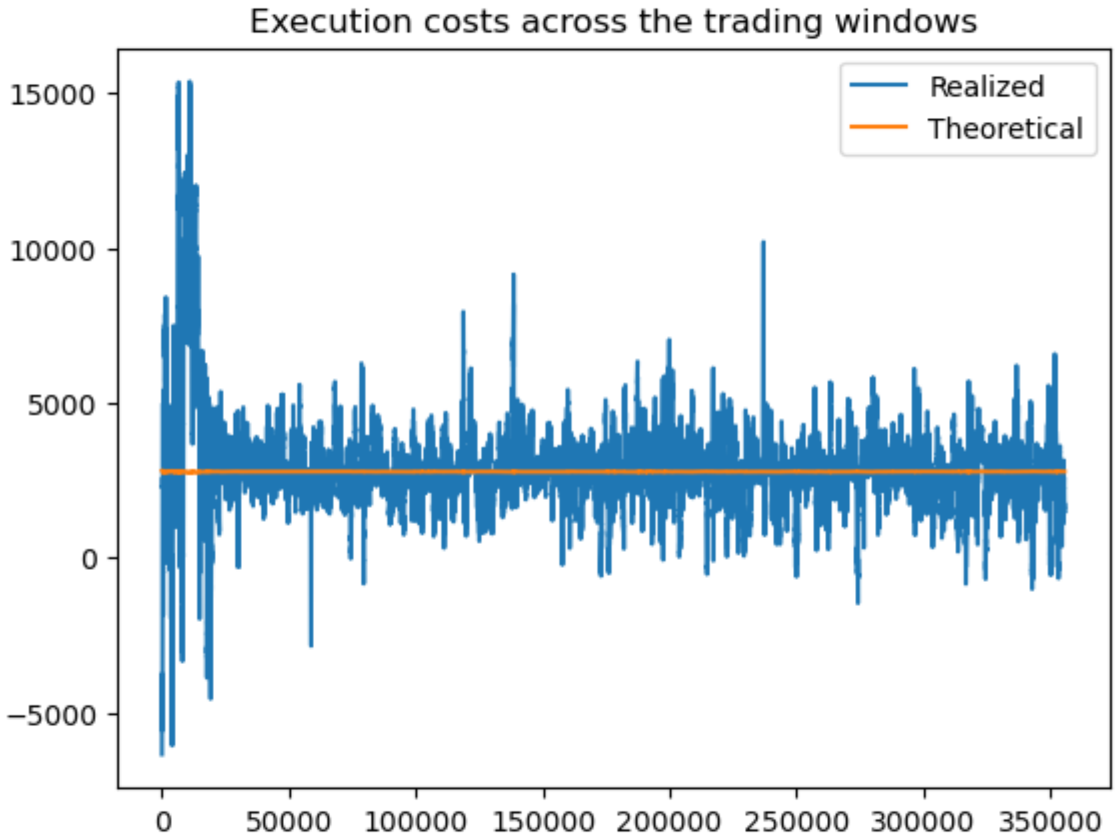
Implement the TWAP liquidation strategy with the initial inventory of Q_0 = 10000 shares. Back-test it using the rolling window method and plot the resulting **execution costs** across the trading (execution) windows. On the same plot, show the **theoretical expected execution costs** for each window. Then, compute

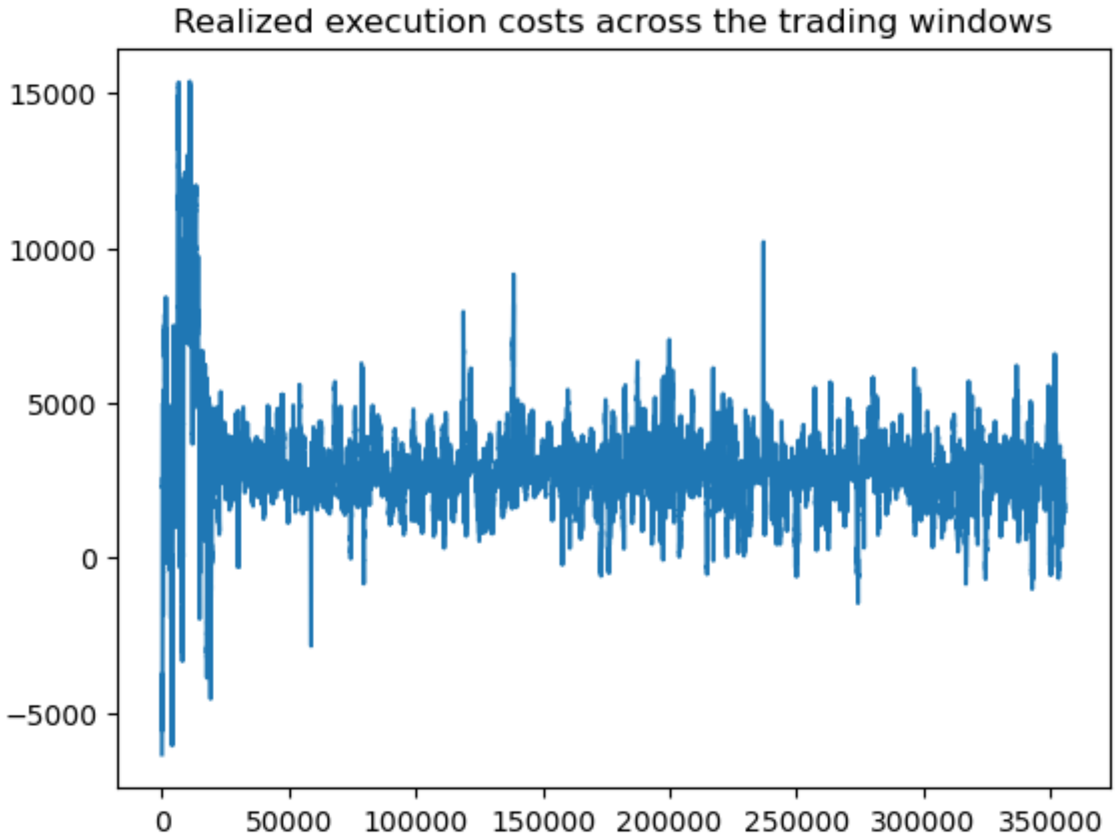
In [3]:
```
k = 0.005 # temporary impact
S = []    # midprice as the unaffected price
N = 60*10 # estimation window of size 10 minutes
T = 60*3  # target execution time of size 3 minutes
f = 10    # trading frequency 1 seconds
Q = [1e4]+[1e4*(T-(i-1)/T) for i in range(0,T)]
nu = -Q[0] /T

start  = 10*60*30          # 10h30
end    = 10*60*(30+11*30) # 15h30
```

In [4]:
```
S, dS, dOF = [], [], []
M = 0
for day in range(len(mats_MSFT)):
    LOB  = mats_MSFT[day]['LOB']
    Time = (np.array((LOB['EventTime'][0][0][:,0]))*1e-3)[start:end:f]#time in seconds, measured from NASDAQ from 10h30
    bid  = np.array(LOB['BuyPrice'][0][0]*1e-4)[start:end:f]
    ask  = np.array(LOB['SellPrice'][0][0]*1e-4)[start:end:f]
    bidvol = np.array(LOB['BuyVolume'][0][0]*1.0)
    askvol = np.array(LOB['SellVolume'][0][0]*1.0)

    OF  = list(np.sum(bidvol[start:end:f,:10],axis=1)-np.sum(askvol[start:end:f,:10],axis=1))
    dOF += list(np.diff(OF))

    s   = list((ask[:,0]+bid[:,0])*0.5)
    S  += s
    dS += list(np.diff(S))

    M+=len(Time)
```

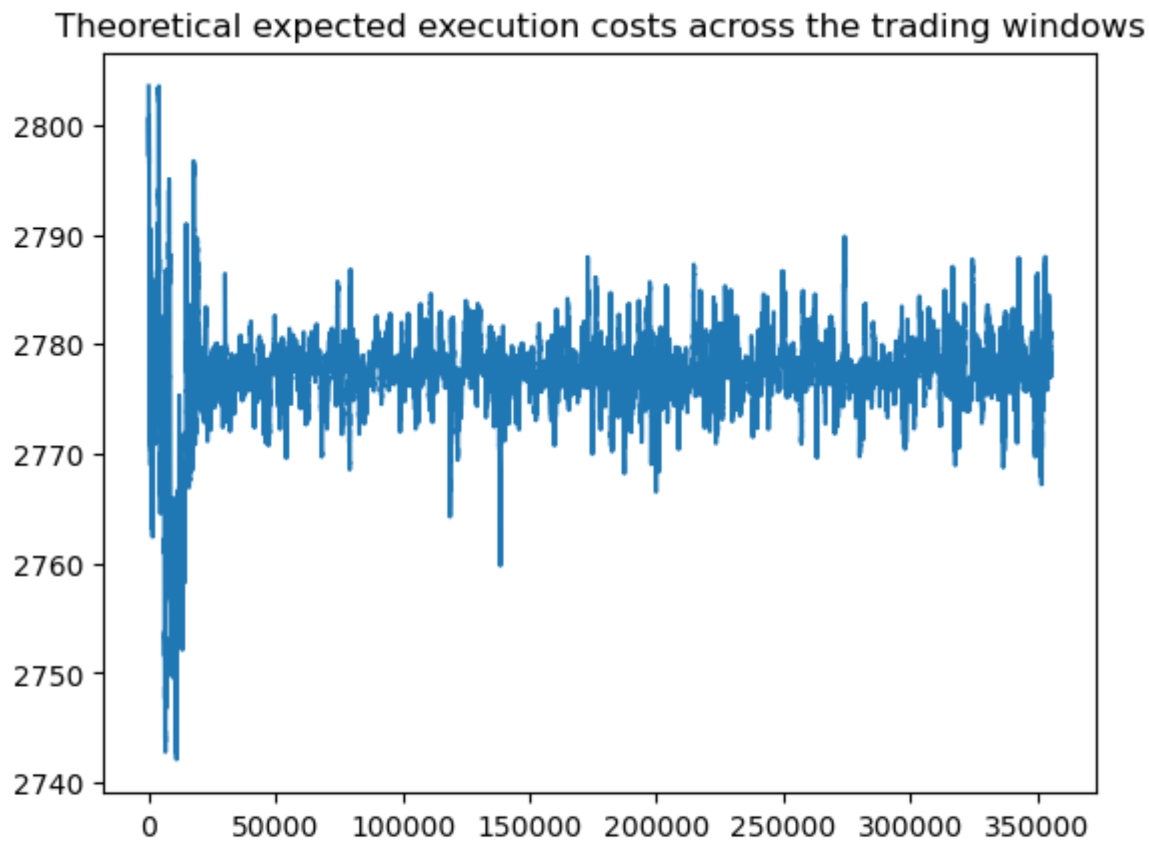
In [5]:
```python
X, cost, cost_th = [], [], [] #absolute PnL, trading cost, theoretical expected execution cost

for t in range(N,M-T):
    #print(f't = {t}')

    # Estimation Window
    l, intercept, r, p, std_err = stats.linregress(dOF[t-N:t], dS[t-N:t])
    #print(f'λ = {l} \np = {p} \nr^2 = {r**2}')

    # Trading window
    x=0
    for i in range(0,T):
        x -= nu* (l*(Q[i+1]-Q[0]) + k*nu + S[t+i+1])
        #print(f'x = {x}')
    X.append(x)

    # Execution cost
    c = Q[0]*S[t]-x
    #print(f'c = {c}')
    cost.append(c)

    # theoretical expected execution cost
    c_th = (Q[0]**2)*k/T + l*(Q[0]**2)*(T+1)/(2*T)
    #print(f'c = {c_th}\n')
    cost_th.append(c_th)
```

In [6]:
```
plt.plot(cost, label = 'Realized')
plt.plot(cost_th, label = 'Theoretical')
plt.title('Execution costs across the trading windows')
plt.legend(loc='best')
plt.show()

plt.plot(cost)
plt.title('Realized execution costs across the trading windows')
plt.show()

plt.plot(cost_th)
plt.title('Theoretical expected execution costs across the trading windows')
plt.show()

#sample averages
mu    = np.mean(cost)
mu_th = np.mean(cost_th)

print(f'The average of the realized execution costs across the trading windows is {mu}.\nThe average of the expected exe
```


Execution costs across the trading windows
Realized
Theoretical
15000
10000
5000
0
−5000
0
50000
100000
150000
200000
250000
300000
350000



Realized execution costs across the trading windows
15000
10000
5000
0
−5000
0
50000
100000
150000
200000
250000
300000
350000




```
The average of the realized execution costs across the trading windows is 2932.3208310514665.
The average of the expected execution costs across the trading windows is 2777.337531492062.
```

Exercice 2

In the discrete-time version of **Almgren-Chriss model**, described in Chapter 5, consider an agent who aims to liquidate the initial inventory Q0 > 0 by time T. The agent is flexible about liquidating the inventory completely, hence, she aims to maximize: max_v [EX^v_T – α(Q^v_T)^2] with some (large) constant α > 0, but **without** the constraint Q^v_T = 0. Find the **optimal liquidation strategy** in a feedback form, using the DPP.

Hint: to solve the recursive DPP equation for the value function V (t, q, s), you can use the ansatz V(t,q,s)=a(t)q^2 +b(t)qs+c(t)s^2 +d(t)q+e(t)s+f(t) and derive the recursive equations for (a, b, c, d, e, f), as well as the formula for the optimal strategy v^∗ in a feedback form, expressed via (a, b, c, d, e, f). This counts as a complete solution.

Problem 2

Objectives: $\underset{\nu}{\text{Max}}\ \mathbb{E}\left[X_T^\nu - \alpha (Q_T^\nu)^2\right]$ with $\begin{cases} Q_T^\nu \neq 0 \\ \alpha \geqslant 0 \end{cases}$

Canonical form:

$$\underset{\nu}{\text{Max}}\ \mathbb{E}\left[X_T^\nu - \alpha(Q_T^\nu)^2\right] = -\underset{\nu_t}{\min}\ \mathbb{E}_t\left[-\sum_{i=0}^{T-1}\left(\lambda(Q_{i+1}-Q_0)+K\nu_i + \tilde{S}_{i+1}\right)\nu_i - \alpha(Q_T^\nu)^2\right]$$

$$= \underset{\nu_t}{\min}\ \mathbb{E}_t \sum_{i=1}^{T}\left[\left(\lambda(Q_i-Q_0)+K\nu_{i-1}+\tilde{S}_i\right)\nu_{i-1} - \frac{\alpha}{T-1}Q_T^{\nu\,2}\right]$$

$$\Rightarrow V(t,q,s) := \underset{\nu_t}{\min}\ \mathbb{E}_t \sum_{i=t+1}^{T}\left[\left(\lambda(Q_i-Q_0)+K\nu_{i-1}+\tilde{S}_i\right)\nu_{i-1} - \frac{\alpha}{T-1}Q_T^{\nu\,2}\right]$$

Dynamics of the states:

$\tilde{S}_i(t,s) = S_{i-1}(t,s)+\sigma\varepsilon_i \quad , \quad \tilde{S}_t(t,s)=s \qquad$ with $\mathbb{E}[\varepsilon_i]=0$.

$Q_i(t,q) = Q_{i-1}(t,q)+\nu_i \quad , \quad Q_t(t,q)=q$

$\Rightarrow Q_T = Q_{T-1}+\nu_T$

$\Rightarrow \nu_T = Q_T - Q_{T-1}$

$\Rightarrow \boxed{\nu_T = Q_T - q}$

Then the DPP reads:

$\forall t = T-2, \dots, 0$

$$* V(t,q,s) = \underset{\nu_t}{\min}\ \mathbb{E}_t\left[\left(\lambda(q+\nu_t-Q_0)+K\nu_t + S + \cancel{\sigma\varepsilon_{t+1}}\right)\nu_t + \frac{\alpha}{T-1}Q_T^2 + V(t+1, q+\nu_t, S+\sigma\varepsilon_{t+1})\right]$$

$$* V(T,q,s) = 0$$

$$* V(T-1,q,s) = \mathbb{E}_{T-1}\left[\left(\lambda(q+\nu_T-Q_0)+K\nu_T + S \underset{}{\overset{+\cancel{\sigma\varepsilon_{t+1}}}{}}\right)\nu_T + \frac{\alpha}{T-1}Q_T^2\right]$$

$$= \mathbb{E}_{T-1}\Big[\underbrace{\left(\lambda(\cancel{q}+Q_T-\cancel{q}-Q_0)+KQ_T-Kq+S\right)(Q_T-q) - \frac{\alpha}{T-1}Q_T^2}_{\lambda(Q_T-Q_0)Q_T + KQ_T^2 - KQ_Tq + Q_TS - \lambda(Q_T-Q_0)q - KQ_Tq + Kq^2 - Sq - \frac{\alpha}{T-1}Q_T^2}\Big]$$

$$V(T-1,q,s) = Kq^2 - \left[2KQ_T + \lambda(Q_T-Q_0)\right]q - Sq + Q_TS + \lambda(Q_T-Q_0)Q_T + KQ_T^2 - \frac{\alpha}{T-1}Q_T^2$$

Anzatz:

$$V(t,q,s) = a(t)q^2 + b(t)qs + c(t)s^2 + d(t)q + e(t)s + f(t)$$

Let's verify the Anzatz

$$a(t)q^2 + b(t)qs + c(t)s^2 + d(t)q + e(t)s + f(t)$$

$$= \min_{\nu_t} \mathbb{E}_t\Big[\big(\lambda(q+\nu_t-Q_0) + k\nu_t + s + \cancel{\sigma\varepsilon_{t+1}}\big)\nu_t + \frac{\alpha}{T-1}Q_T^2$$
$$+ a(t+1)(q+\nu_t)^2 + b(t+1)(q+\nu_t)(s+\cancel{\sigma\varepsilon_{t+1}}) + \underbrace{c(t+1)(s+\sigma\varepsilon_{t+1})^2} + d(t+1)(q+\nu_t) + e(t+1)(s+\cancel{\sigma\varepsilon_{t+1}}) + f(t+1)\Big]$$

$$\mathbb{E}\left[c(t+1)(s+\sigma\varepsilon_{t+1})^2\right] = c(t+1)s^2 + c(t+1)\sigma^2\times 1 + 0$$
$$= c(t+1)(s^2+\sigma^2)$$

$$= \min_{\nu}\Big[(\lambda+k)\nu^2 + a(t+1)(q+\nu_t)^2 + \big(b(t+1)s + d(t+1) + \lambda(q-Q_0)\big)\nu\Big]$$
$$+ \underbrace{\big(b(t+1)s + d(t+1)\big)q}_{b(t+1)qs + d(t+1)q} + c(t+1)(s^2+\sigma^2) + e(t+1)s + f(t+1) + \frac{\alpha}{T-1}Q_T^2$$

$$\min_x (ax^2 + bx + c)$$
$$\Rightarrow 2ax^* + b = 0 \Rightarrow x^* = -\frac{b}{2a}$$

Let's find ν^* :

$$M = (\lambda+k)\nu^2 + a(t+1)(q+\nu_t)^2 + \big(b(t+1)s + d(t+1) + \lambda(q-Q_0)\big)\nu$$
$$= [\lambda + k + a(t+1)]\nu^2 + [b(t+1)s + d(t+1) + \lambda(q-Q_0) + 2a(t+1)q]\nu + a(t+1)q^2$$

$$\Rightarrow \min_\nu [\,M\,] = \boxed{\nu^* = -\frac{2a(t+1)q + b(t+1)s + d(t+1) + \lambda(q-Q_0)}{2[\lambda + k + a(t+1)]}}$$

Let's replace the value of ν^*

$$\min_x (ax^2 + bx + c) = -\frac{b}{2a}$$
$$\Rightarrow ax^{*2} + bx^* + c = a\left(-\frac{b}{2a}\right)^2 + b\left(-\frac{b}{2a}\right) + c$$
$$= -\frac{b^2}{4a} - \frac{b^2}{2a} + c$$
$$= -\frac{3b^2}{4a} + c$$

$$a(t)q^2 + b(t)qs + c(t)s^2 + d(t)q + e(t)s + f(t)$$
$$= -\frac{3}{4}\times\frac{[2a(t+1)q + b(t+1)s + d(t+1) + \lambda(q-Q_0)]^2 \; (*)}{[\lambda + k + a(t+1)]} + a(t+1)q^2$$
$$+ b(t+1)qs + d(t+1)q + c(t+1)s^2 + e(t+1)s + f(t+1) + \frac{\alpha}{T-1}Q_T^2 + c(t+1)\sigma^2$$

Let's find $a(t), b(t), c(t), d(t), e(t), f(t)$:

* For $t = T-1$:

$$\begin{cases} a(T-1) = k \\ b(T-1) = -1 \\ c(T-1) = 0 \\ d(T-1) = -[2kQ_T + \lambda(Q_T - Q_0)] \\ e(T-1) = Q_T \\ f(T-1) = \lambda(Q_T - Q_0) + kQ_T^2 - \frac{\alpha}{T-1}Q_T^2 \end{cases}$$

For $t = T-2, \dots, 0$ Let's compute the numerator of the previous formula: (*)

First: $[2a(t+1)q + b(t+1)s + d(t+1) + \lambda(q-Q_0)]^2$

$$= [2a(t+1)q+b(t+1)s]^2 + 2[2a(t+1)q+b(t+1)s][d(t+1)+\lambda(q-Q_0)] + [d(t+1)+\lambda(q-Q_0)]^2$$

$$= [2a(t+1)q]^2 + 4a(t+1)b(t+1)qs + [b(t+1)s]^2$$

$$+ 4a(t+1)d(t+1)q + \underbrace{4a(t+1)\lambda(q-Q_0)q}_{4a(t+1)\lambda q^2 - 4a(t+1)\lambda Q_0 q} + 2b(t+1)d(t+1)s + \underbrace{2b(t+1)\lambda(q-Q_0)s}_{2b(t+1)\lambda qs - 2b(t+1)\lambda Q_0 s}$$

$$+ d(t+1)^2 + \underbrace{2d(t+1)\lambda(q-Q_0)}_{2d(t+1)\lambda q - 2d(t+1)\lambda Q_0} + \underbrace{\lambda^2(q-Q_0)^2}_{\lambda^2 q^2 - 2\lambda Q_0 q + \lambda^2 Q_0^2}$$

$$= [4a^2(t+1) + 4a(t+1) + \lambda^2]q^2 + [4a(t+1)b(t+1) + 2b(t+1)\lambda]qs + b^2(t+1)s^2$$

$$+ [4a(t+1)d(t+1) - 4a(t+1)\lambda Q_0 + 2d(t+1)\lambda - 2\lambda Q_0]q + [2b(t+1)d(t+1) - 2b(t+1)\lambda Q_0]s$$

$$+ \underbrace{d^2(t+1) - 2d(t+1)\lambda Q_0 + \lambda^2 Q_0^2}_{[d(t+1)-\lambda Q_0]^2}$$

Finally

$$\begin{cases} a(t) = a(t+1) - \frac{3}{4}\left[\frac{4a^2(t+1) + 4a(t+1) + \lambda^2}{\lambda + k + a(t+1)}\right] \\ b(t) = b(t+1) - \frac{3}{4}\left[\frac{4a(t+1)b(t+1) + 2b(t+1)\lambda}{\lambda + k + a(t+1)}\right] \\ c(t) = c(t+1) - \frac{3}{4}\left[\frac{b^2(t+1)}{\lambda + k + a(t+1)}\right] \\ d(t) = d(t+1) - \frac{3}{4}\left[\frac{4a(t+1)(d(t+1) - \lambda Q_0) + 2d(t+1)\lambda - 2\lambda Q_0}{\lambda + k + a(t+1)}\right] \\ e(t) = e(t+1) - \frac{3}{4}\left[\frac{2b(t+1)(d(t+1) - \lambda Q_0)}{\lambda + k + a(t+1)}\right] \\ f(t) = f(t+1) + c(t+1)\sigma^2 + \frac{\alpha}{T-1}Q_T^2 - \frac{3}{4}\frac{[d(t+1) - \lambda Q_0]^2}{[\lambda + k + a(t+1)]} \end{cases}$$