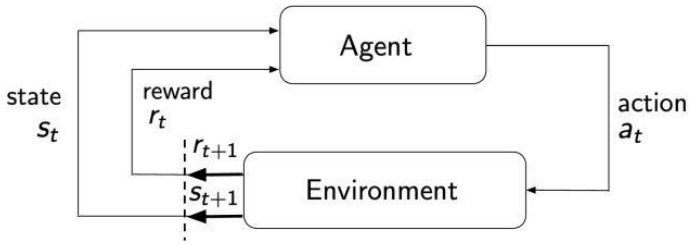
Reinforcement Learning for Optimal Execution

2021 Fall ISE537 Final Project--Fandi MA(3652892113)

Contents

I. Introduction....1
II. State Space....1
III. Action Space....1
IV. Time Horizon....1
V. Reward Design....1
VI. Train Simulation Environment (from blackboard)....1
VII. Q learning....2
VIII. Evidence of Convergence....2
IX. Compare Q-learning with Two Strategies & Sharpe Ratio Evaluation....2
A. Set current mid-price = 96....3
a. Q learning....3
b. Executing everything at time 0....3
c. Executing with a constant trading speed 2....4
B. Pick 5 current mid-price....4
a. Q learning....4
b. Executing everything at time 0....5
c. Executing with a constant trading speed....5
X. Analysis Based on a New Simulation Environment....6
A. Train new environment....6
a. Reward design....6
b. Assumptions....6
c. Settings....6
B. Evidence of Convergence....7
C. Compare Q learning with 2 strategies and Sharpe ratio Evaluation....7
a. Ser current mid-price =96....7
Q learning....7
Executing everything ay time 0....7
Executing with a constant trading speed 2....8
b. Pick 5 current mid-prices....9
Q learning....9
Executing everything ay time 0....9
Executing with a constant trading speed....10
XI. Almgren framework and Reinforcement Learning....10

I. Introduction



The project is based on Value-based reinforcement learning(Q learning). 2 simulated environments are involved and optimal strategies for rewards optimization are searched. Such strategies are then compared with 2 naïve trading strategies and then the report ends with a short comparison between reinforcement learning and Almgren Framework.

II. State Space

State space S = (p, i): p is the mid-price and i is the inventory, where $S=S_p*S_i$

$S_p = \{p_{min}, p_{min} + \Delta, p_{min} + 2\Delta, \ldots, p_{max}\}$ $p_{min} = 90, p_{max} = 99.9, \Delta = 0.1$, *hence 100 price level*
$S_i = \{0, \delta, 2\delta, \ldots, I\}$ $I = 20$*million,* $\delta = 1$*million,hence 20 inventory level*
$|S| = 100 * 20 = 2000$

III. Action Space

Action space $A = S_i$.
For $a \in A$, a can take value between 0 and I.

IV. Time Horizon

Suppose we the time horizon is daily.
The time horizon is from 0 to T, where T = 10

V. Reward Design

Reward design is split into 2 parts

1. Reward for 0 to T-1 period: the reward from 0 to T-1 is determined by average executed price, X(p,a), and action size a
   $r_t(p, a) = X(p, a) * a$
2. Reward for the last period: the reward for the last period is determined by average executed price, X(p, a) and inventory i of the last horizon. There will be punishment if there is inventory at time T, and the punishment factor $\gamma_2 = 0.05$
   $r_T(p, i) = X(p, i) * i - \gamma_2 * i^3$

VI. Train Simulation Environment (from blackboard)

The environment will return the execution price and reward to the agent. Simulation based on the current mid-price *p* (mid-point between the quoted Bid Price and Ask Price) as well as the action size *a*

Assumption:

1. Action size has quadratic impact on the execution price, and the market impact of the current action size is 0.01.
2. The next execution price only based on the current execution price.
3. Action size cannot exceed the inventory size at each state.

Setting:
p: current mid price
p_trade: the current execution price
p_next: the next execution price
a: action size, which refers to the amount to trade
r: reward that will be received by the agent after action

where,
p_trade = p – 0.01 * a^2
p_next = p_trade + e e ~N(0,0.5)
r = p_trade * a

The simulator returns the reward r under each situation and the price p_next for the next state.

VII. Q learning

Algorithm: Bellman Equation

$$Q_t^\star(s,a) = \mathbb{E}[r_t(s,a)] + \mathbb{E}_{s' \sim P_t(s,a)}\left[\sup_{a' \in \mathcal{A}} Q_{t+1}^\star(s',a')\right]$$

Terminal Condition

$$Q_T^\star(s,a) = \mathbb{E}[r_T(s)] \text{ for all } a \in \mathcal{A}$$

Based on Bellman Equation, Q table can be generated:

For t =T:

$$Q_T^{(n+1)}(s_T,a) = (1-\beta)Q_T^{(n)}(s_T,a) + \beta r_T, \quad \forall a \in \mathcal{A}$$

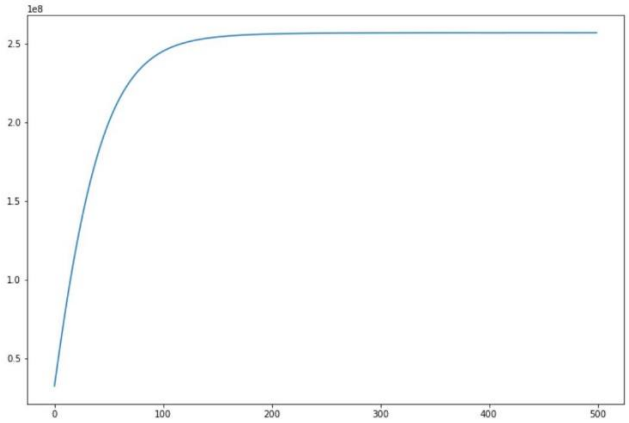
For 0 <= t <= T – 1:

$$Q_t^{(n+1)}(s_t,a_t) \leftarrow (1-\beta)\underbrace{Q_t^{(n)}(s_t,a_t)}_{\text{old estimation}} + \beta\underbrace{\left[r_t + \max_{a'} Q_{t+1}^{(n)}(s_{t+1},a')\right]}_{\text{new estimation}}$$

Therefore, a 4-dimension Q table is generated with T = 10, I = 20, A = 20, P =100 and learning rate β = 0.04.
Besides, 500 iterations were made for generating Q table.

VIII. Evidence of Convergence

To find evidence of convergence, the sum of absolute values of the Q table (from 0 to T-1) are calculated and plotted for each iteration.



It can be seen that the Q table converges fast from 0 to 100 iteration and the speed of convergence slows after the 100th iteration. At around the 200th iteration, the Q table converges.

IX. Compare Q-learning with Two Strategies & Sharpe Ratio Evaluation

Suppose initial inventory is 15 (I = 15)
For evaluation purpose, *Sharpe ratio* is introduced. Sharpe ratio is is the average return earned in excess of the risk-free rate per unit of volatility. It represents the additional amount of return that an investor receives per unit of increase in risk.

Sharpe Ratio = $\frac{E(R)-rf}{\sigma(R)}$
E(R): expected return
rf: risk free rate
σ(R): volatility of the return

Note: for risk free rate, 13-week treasury bill rate, which is 0.00043 is picked and converted to daily risk-free rate.

$rf = \sqrt[90]{1 + 0.00043} - 1$

$= 4.776762258629219 * e^{-6}$

A. Set current mid-price = 96

a. *Q learning*

the *action size* of each horizon suggested by Q learning is

[1, 2, 2,0,1,4,1,1,2]

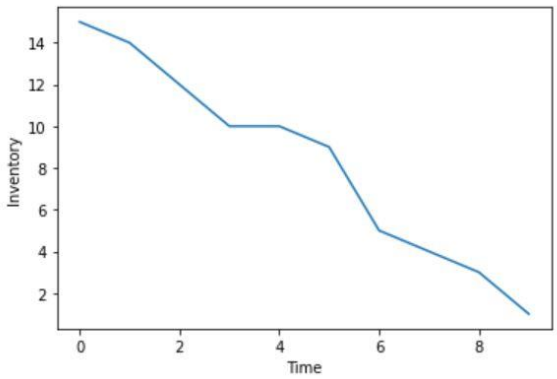
the *inventory* of each horizon suggested by Q learning is

[15, 14, 12, 10, 10, 9, 5, 4, 3, 1]

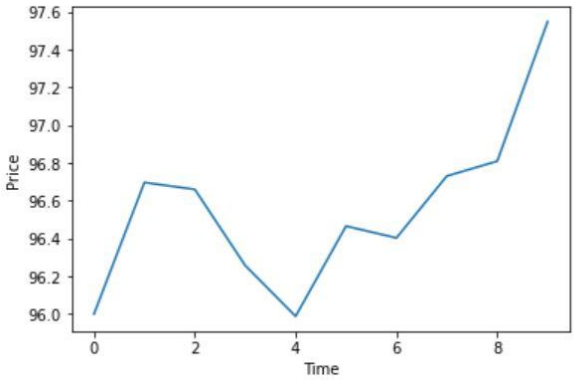
the *price* of each horizon suggested by Q learning is

[96. , 96.69652249, 96.65992085, 96.25684462, 95.98764004, 96.46555674, 96.40296851, 96.73057866, 96.80925936, 97.54848214]

Inventory plot:



Price plot:



Evaluation:

Average Return:0.0017875422875551156

Standard deviation:0.004256841210193

Sharpe Ratio :0.4188001001840653

The overall trend of the price plot is upward, and the Sharpe ratio is 0.4188, which means for every unit of volatility, the investor receives 0.4188 additional amount of return.

b. *Executing everything at time 0*

Here all 15 inventories will be executed at time 0

the action size of each horizon is

[15., 0., 0., 0., 0., 0., 0., 0., 0.]

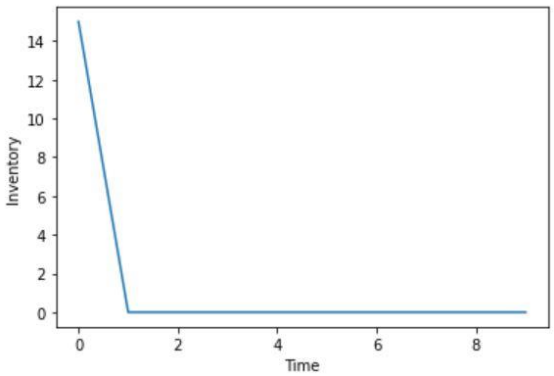
the inventory of each horizon is

[15., 0., 0., 0., 0., 0., 0., 0., 0., 0.]

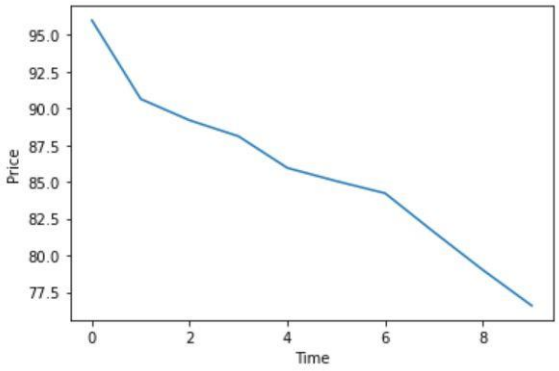
the price of each horizon is

[96. , 90.6402023 , 89.19315266, 88.10520667, 85.94932316, 85.05555251, 84.22650397, 81.58190467, 79.00481918, 76.57651099]

Inventory plot:



Price plot:



Evaluation:
Average Return:-0.024703644897371868
Standard deviation:0.014826904666457743
Sharpe Ratio :-1.6664585235735194

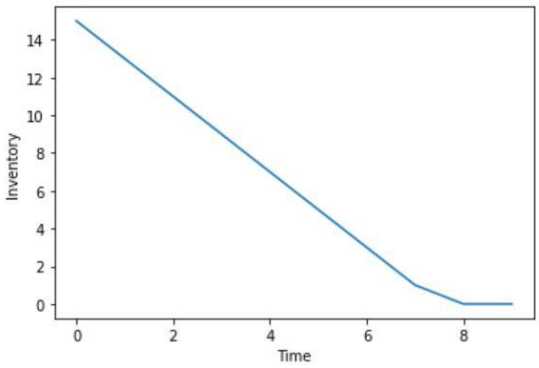
When we liquidate all the inventory at the beginning, the trend pf the price path is downward and we get Sharpe ratio -1.67, which indicates for every unit of volatility, the investor experiences about 1.67 additional amount of loss under this scenario.

c. Executing with a constant trading speed 2

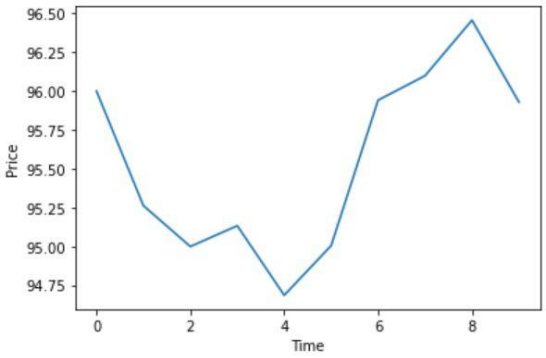
In this case, we execute 2 each time horizon until we liquidate all 15 inventories.

the action size of each horizon is
[2, 2, 2, 2, 2, 2, 2, 1, 0, 0]
the inventory of each horizon is
[15., 13., 11., 9., 7., 5., 3., 1., 0., 0.]
the price of each horizon is
[96. , 95.26153375, 94.99820961, 95.13197727, 94.68400329, 95.00409545, 95.94104442, 96.09852865, 96.45662968, 95.92882292]

Inventory plot:



Price plot:



Evaluation:
Average Return:-6.874565822696028e-05
Standard deviation:0.005546872168938272
Sharpe Ratio :-0.08991475610703054

When executing with a constant trading speed, the price path first goes down and then moves upward. We get Sharpe ratio -0.08991, which indicates for every unit of volatility, the investor experiences about 0.08991additional amount of loss under this scenario.

The second strategy is more volatile than the other 2.
Therefore, with Sharpe ratio of 0.4188, which is the highest among 3 strategies, the performance of Q learning is the best in this case. Besides, the strategy executing all inventory at the beginning performs the worst.

B. Pick 5 current mid-price

Next, instead of just look 1 single mid-price, I picked 5 current mid-price [90.2, 92.2, 94.2, 96.2, 98.2] to see the changes.

a. Q learning

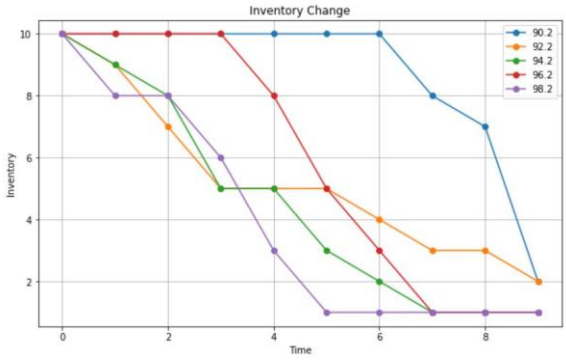
Action size table for each horizon Inventory table for each horizon

	0	1	2	3	4	5	6	7	8
90.2	0	0	1	2	0	2	1	1	2
92.2	1	1	1	1	1	2	1	0	1
94.2	1	2	1	2	0	1	2	0	0
96.2	0	0	0	2	2	2	2	1	0
98.2	2	2	3	2	0	0	0	0	0

	0	1	2	3	4	5	6	7	8	9
90.2	10	10	10	10	10	10	10	8	7	2
92.2	10	9	7	5	5	5	4	3	3	2
94.2	10	9	8	5	5	3	2	1	1	1
96.2	10	10	10	10	8	5	3	1	1	1
98.2	10	8	8	6	3	1	1	1	1	1

Price Change and Sharpe ratio

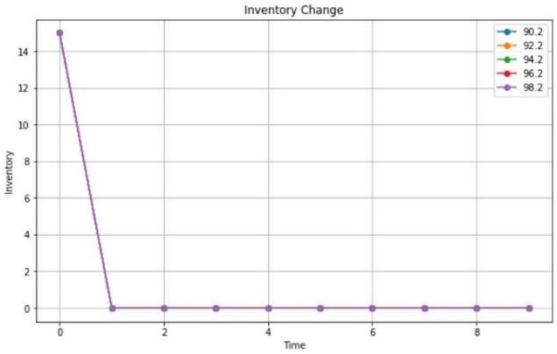
	0	1	2	3	4	5	6	7	8	9	Return_avg	Return_std	Sharpe
90.2	90.2	90.020404	90.617500	90.523143	90.216909	90.968893	90.436321	90.331797	90.653927	90.778065	0.000720	0.004684	0.061871
92.2	92.2	92.575492	93.012929	92.838764	92.973375	92.938816	93.375185	93.776324	93.624873	93.276221	0.001295	0.003289	0.263023
94.2	94.2	93.946892	93.753964	92.673740	92.749905	92.423174	92.189981	91.803042	92.437275	92.191514	-0.002382	0.004809	-0.584635
96.2	96.2	94.785663	94.760157	95.076397	94.402815	94.101501	94.144391	94.416111	94.874193	94.422535	-0.002053	0.006170	-0.402437
98.2	98.2	98.133677	98.714325	98.230344	98.149998	98.019346	97.543347	97.551410	97.266631	97.474562	-0.000818	0.003402	-0.366936

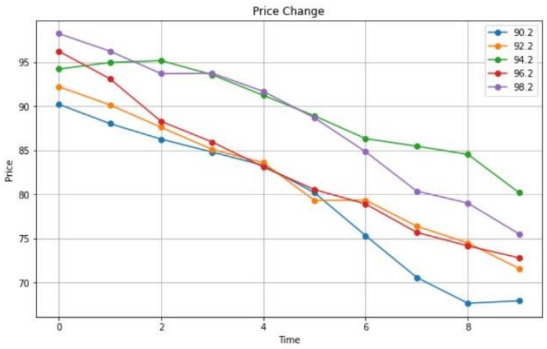


b. Executing everything at time 0

Action size [15,0,0,0,0,0,0,0,0,0] for each horizon
Inventory [15,0,0,0,0,0,0,0,0,0]for each horizon
Price Change and Sharpe ratio

	0	1	2	3	4	5	6	7	8	9	Return_avg	Return_std	Sharpe
90.2	90.2	88.011562	86.253461	84.794608	83.294299	80.190639	75.335579	70.603859	67.684057	67.952331	-0.030761	0.021781	-1.432008
92.2	92.2	90.133896	87.597934	85.092030	83.585873	79.323923	79.352005	76.401708	74.503337	71.627095	-0.027569	0.014471	-1.934877
94.2	94.2	94.933733	95.149443	93.544316	91.217069	88.891996	86.316438	85.460699	84.540830	80.209088	-0.017563	0.017755	-1.013389
96.2	96.2	93.085559	88.263508	85.935353	83.103679	80.549605	78.919024	75.701822	74.178200	72.825510	-0.030401	0.010891	-2.830921
98.2	98.2	96.215735	93.671386	93.713870	91.663797	88.668110	84.829949	80.373352	79.018559	75.535149	-0.028613	0.016383	-1.772695



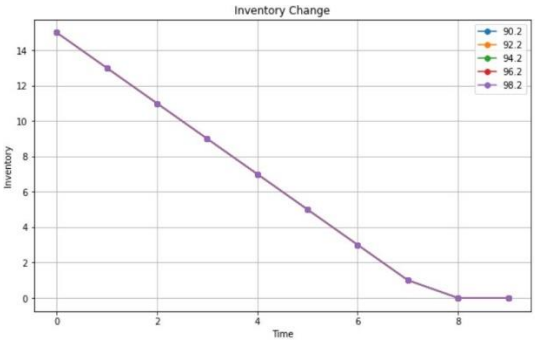


c. Executing with a constant trading speed 2

Action size[2,2,2,2,2,2,2,1,0] for each horizon
Inventory size [15,13,11,9,7,5,3,1,0,0]for each horizon

Price Change and Sharpe ratio

	0	1	2	3	4	5	6	7	8	9	Return_avg	Return_std	Sharpe
90.2	90.2	89.956287	89.820632	89.725565	90.345773	90.485102	90.878836	91.000045	91.588959	91.023304	0.001019	0.004363	0.134906
92.2	92.2	92.886035	93.726233	93.122284	94.044832	94.481607	95.091017	95.156211	94.883762	94.590392	0.002864	0.005963	0.408163
94.2	94.2	93.149383	93.673421	93.123863	92.979043	93.024953	92.884528	93.340239	93.414982	93.644612	-0.000644	0.005273	-0.203768
96.2	96.2	96.086700	97.121521	96.427917	96.435916	96.777146	97.471482	97.247294	97.125924	97.137921	0.001091	0.005345	0.123731
98.2	98.2	98.065881	97.783623	98.099442	98.167051	98.673740	98.567940	99.081679	98.940366	98.884363	0.000776	0.003020	0.114582



The second strategy attributes to a more volatile market, while Q learning and constant trading strategy is less risky.

In this case, by looking at the Sharpe ratio of 5 selected current min-prices of each strategy, executing with a constant trading speed reaches the best Sharpe ratio on average and liquidating at time 0 performs the worst. Strategy given by Q learning performs somewhere in between.

X. Analysis Based on a New Simulation Environment

A. Train new environment

The new environment is based on the current mid-price p (mid-point between the quoted Bid Price and Ask Price, action size a, and the inventory level i. The state space, action space and time horizon remain the same

a. Reward design

1.Reward for 0 to T-1 period: the reward from 0 to T-1 is determined by average executed price, X(p,a), action size a, and inventory rt(p, a,i)= X(p, a) * a - 0.02*(i-a)^2. There will be punishment if there is inventory at time t, and the punishment factor $\gamma 1 = 0.02$.

2.Reward for the last period: the reward for the last period is determined by average executed price, X(p, a) and inventory i of the last horizon. rT (p, i) = X(p, i) * i - $\gamma 2$ * i^3. There will be heavier punishment if there is inventory at time T, i has cubic influence and the punishment factor $\gamma 2 = 0.05$(same as before).

b. Assumptions

1.Action size has **cubic impact** on the execution price, and the market impact of the current action size is **0.001**.
2.The next execution price only based on the current execution price.
3.Action size cannot exceed the inventory size at each state.

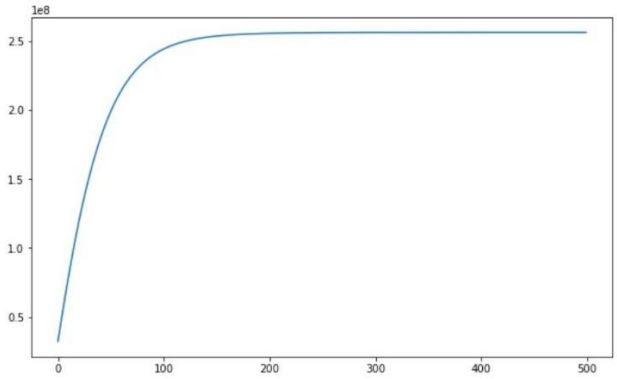
c. Settings

p: current mid-price
p_trade: the current execution price
p_next: the next execution price
a: action size, which refers to the amount to trade
r: reward that will be received by the agent after action
i: inventory at time t
where,
p_trade = p – 0.001 * a^3
p_next = p_trade + e e ~N(0,0.5)
r = p_trade * a – 0.02 * (i - a)^2

The simulator returns the reward r under each situation and the price p_next for the next state.

The new simulation environment considers punishment for inventory from time 0 to time T-1 and assumes cubic impact of the action impact on market. Therefore, agents in this environment will under the pressure of the inventory level in each phase.

B. Evidence of Convergence



The Q table converges at around the 200th iteration.

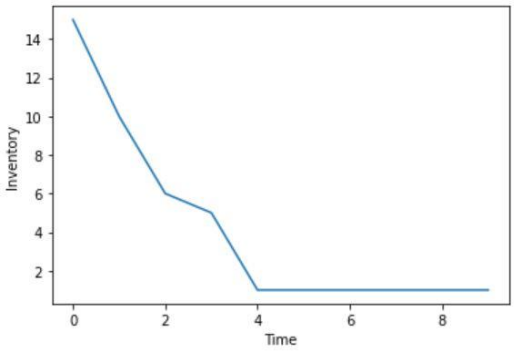
C. Compare Q learning with 2 strategies and Sharpe ratio Evaluation

a. Set current mid-price =96

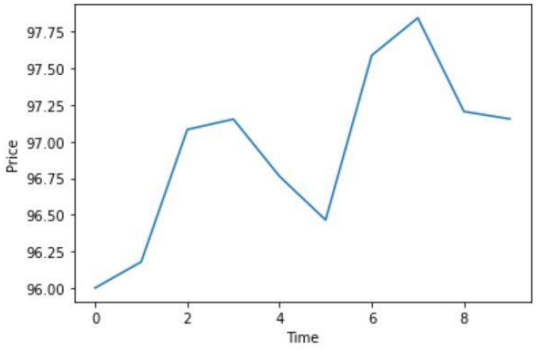
Q learning

the *action size* of each horizon suggested by Q learning is
[5, 4, 4,1,0,0,0,0,0]
the *inventory* of each horizon suggested by Q learning is
[15., 10., 6., 5., 1., 1., 1., 1., 1., 1.]
the *price* of each horizon suggested by Q learning is
[96. , 96.17833124, 97.08249897, 97.15365002, 96.7632086 , 96.46453114, 97.58907766, 97.8453307 , 97.20585809, 97.15474711]

Inventory plot:



Price plot:



Evaluation:
Average Return:0.0013453421883933775
Standard deviation:0.005995201207279716
Sharpe Ratio :0.22440317545302502

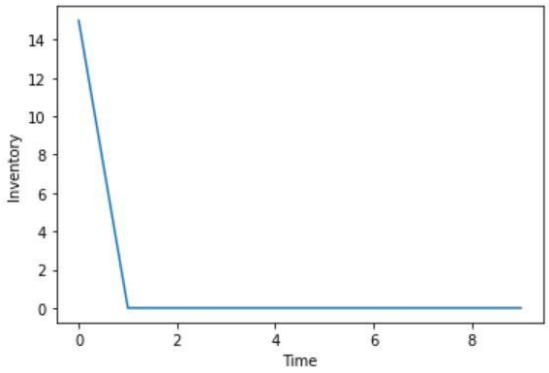
The overall trend of the price plot is upward, and the Sharpe ratio is 0.2244, which means for every unit of volatility, the investor receives 0.2244 additional amount of return

Executing everything ay time 0

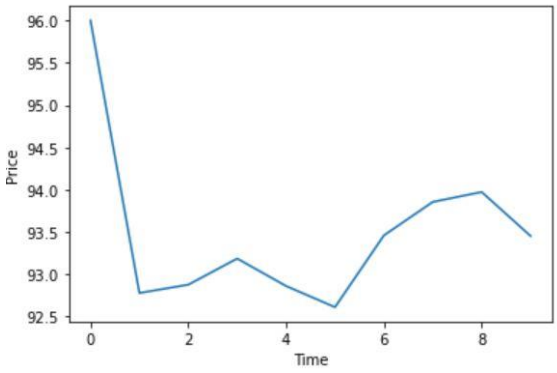
the *action size* of each horizon suggested by Q learning is
[15, 0, 0,0,0,0,0,0,0]
the *inventory* of each horizon suggested by Q learning is
[15., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
the *price* of each horizon suggested by Q learning is

[96. , 92.77674473, 92.87537063, 93.18418527, 92.85926911, 92.60947956, 93.45817445, 93.85433944, 93.97154623, 93.45102692]

Inventory plot:



Price plot:



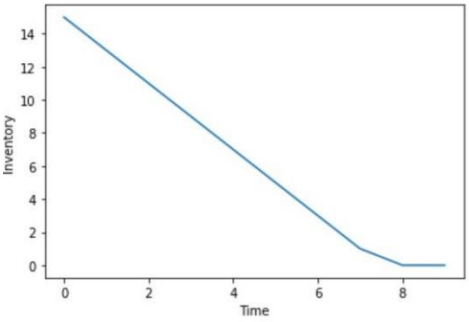
Evaluation
Average Return:-0.0029168216620322454
Standard deviation:0.012331472798607
Sharpe Ratio :-0.2365347359288453

After the initial liquidation of all stock, the price experiences a sharp decrease and fluctuates at a relatively lower price level compared to the initial mid-price. Besides, Sharpe ratio is -0.23, which indicates that investors will have 0.23 extra loss per unit volatility.

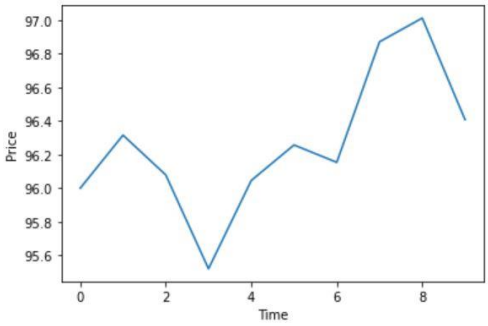
Executing with a constant trading speed 2

the *action size* of each horizon suggested by Q learning is
[2, 2, 2, 2, 2, 2, 2, 1, 0]
the *inventory* of each horizon suggested by Q learning is
[15., 13., 11., 9., 7., 5., 3., 1., 0., 0.]
the *price* of each horizon suggested by Q learning is
[96. , 96.31510956, 96.07861703, 95.5208673 , 96.04553295, 96.25675613, 96.15341324, 96.86992743, 97.01111158, 96.4074221]

Inventory plot:



Price plot:



Evaluation:
Average Return:0.0004807180571591512
Standard deviation:0.004756038256705783
Sharpe Ratio :0.10107531336220059

At time 8, liquidate everything and the price experiences drop after liquidation. Before liquidation, the price reaches lowest level at time 3 and then keeps increasing until time 8. Sharpe Ratio of 0.1 indicates that that investor will have 0.1 extra return per unit volatility.

Comparing the standard deviation of 3 scenarios, we can see that the strategy executing all at time 0 can lead to a riskier market since the std is much higher than the other 2.
With strategy suggested by Q learning, we got Sharpe ratio of 0.2244, which is the highest among 3 strategies. Besides, the strategy executing all inventory at the beginning performs the worst.

b. Pick 5 current mid-prices

mid-prices [90.2, 92.2, 94.2, 96.2, 98.2]

Q learning

Action size table for each horizon

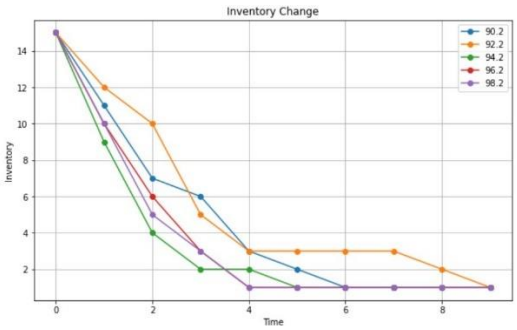
	0	1	2	3	4	5	6	7	8
90.2	4	4	1	3	1	1	0	0	0
92.2	3	2	5	2	0	0	0	1	1
94.2	6	5	2	0	1	0	0	0	0
96.2	5	4	3	2	0	0	0	0	0
98.2	5	5	2	2	0	0	0	0	0

Inventory table for each horizon

	0	1	2	3	4	5	6	7	8	9
90.2	10	10	10	10	10	10	10	8	7	2
92.2	10	9	7	5	5	5	4	3	3	2
94.2	10	9	8	5	5	3	2	1	1	1
96.2	10	10	10	10	8	5	3	1	1	1
98.2	10	8	8	6	3	1	1	1	1	1

Price Change and Sharpe ratio

	0	1	2	3	4	5	6	7	8	9	Return_avg	Return_std	Sharpe
90.2	90.2	90.690867	90.216316	90.700186	90.378744	91.012812	90.848171	90.711093	91.120008	91.382582	0.001457	0.004508	0.322196
92.2	92.2	91.560107	90.658943	90.713789	89.830531	89.614741	89.417254	89.670761	89.754645	89.005297	-0.003900	0.004906	-0.795840
94.2	94.2	94.235296	94.293124	94.111611	94.468524	94.316426	94.681248	94.733427	95.133331	95.807934	0.001886	0.003011	0.624857
96.2	96.2	95.372975	94.396447	93.633322	93.183241	93.150478	93.075944	92.352803	91.293856	92.843172	-0.003905	0.008736	-0.447519
98.2	98.2	98.507710	98.500695	98.418643	98.217590	99.024964	98.560517	98.623600	98.882243	99.307460	0.001253	0.003806	0.328001



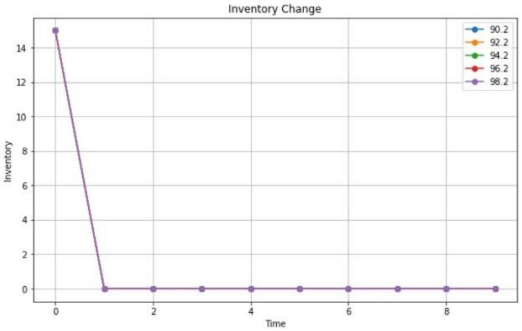
Executing everything ay time 0

Action size [15,0,0,0,0,0,0,0,0,0] for each horizon

Inventory [15,0,0,0,0,0,0,0,0,0]for each horizon

Price Change and Sharpe ratio

	0	1	2	3	4	5	6	7	8	9	Return_avg	Return_std	Sharpe
90.2	90.2	86.977394	86.443679	86.652059	86.561539	87.092443	86.802475	86.160325	86.312034	86.602773	-0.004440	0.012557	-0.353989
92.2	92.2	88.309630	87.621264	87.807329	87.417051	86.003915	85.708659	85.381755	84.905398	84.981144	-0.008935	0.013540	-0.660230
94.2	94.2	90.860472	90.985860	90.770827	91.802453	91.521089	91.586212	90.445009	89.705689	89.805321	-0.005216	0.013142	-0.397272
96.2	96.2	92.823643	92.484190	93.591421	93.953886	93.617231	94.134826	93.631920	93.339859	93.106473	-0.003547	0.013125	-0.270627
98.2	98.2	94.996318	94.767047	94.918342	94.748774	95.373844	95.793629	96.076080	95.795643	96.295894	-0.002109	0.011954	-0.176785



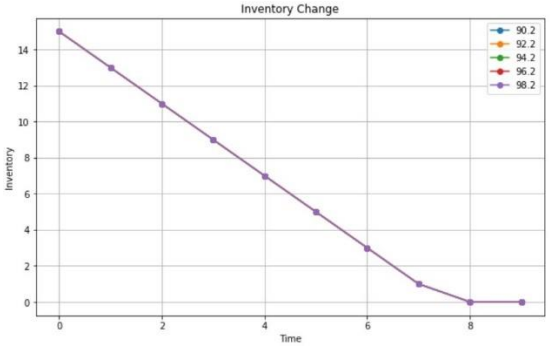
Executing with a constant trading speed

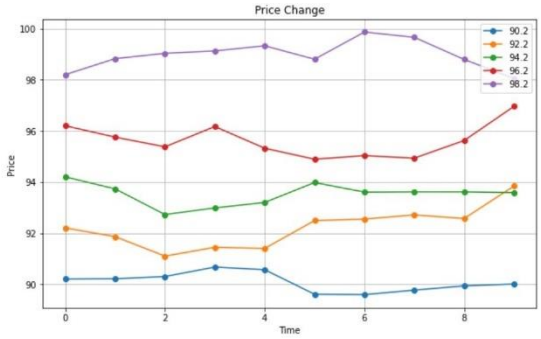
Action size[2,2,2,2,2,2,2,1,0] for each horizon

Inventory size [15,13,11,9,7,5,3,1,0,0]for each horizon

Price Change and Sharpe ratio

	0	1	2	3	4	5	6	7	8	9	Return_avg	Return_std	Sharpe
90.2	90.2	90.211836	90.299649	90.670397	90.566705	89.607389	89.592859	89.766173	89.933039	90.005691	-0.000232	0.004164	-0.056822
92.2	92.2	91.861293	91.098472	91.445239	91.398679	92.488443	92.546982	92.711027	92.569535	93.837274	0.001980	0.007049	0.280166
94.2	94.2	93.730114	92.724509	92.984208	93.197895	93.978582	93.604891	93.616272	93.615975	93.583077	-0.000717	0.005415	-0.133237
96.2	96.2	95.753572	95.374818	96.169560	95.319574	94.890771	95.031719	94.932004	95.620322	96.958255	0.000898	0.007495	0.119128
98.2	98.2	98.825885	99.034380	99.128473	99.329590	98.803857	99.867702	99.661886	98.796464	98.046211	-0.000156	0.006434	-0.024950





From perspective of volatility, we can see that liquidating at time 0 gives us highest volatility and the other 2 strategies gives us lower volatility. In this case, strategy from Q learning gives us the best Sharpe ratio on average and liquidating at time 0 performs the worst. The constant trading strategy lies somewhere in between.

XI. Almgren Framework and Reinforcement Learning

Reinforcement learning is relatively more flexible since it can take many market factors into consideration, for example, inventory, current price, policies etc. However, the more aspects and dimensions, the more time consuming for training process. Besides, the performance of reinforcement learning depends on the similarity of simulated environment and the real market, the higher the better.

In contrary to Reinforcement learning, *Almgren framework* is less flexible and is under the certain fixed framework. This makes it more efficient for us to solve optimal liquidation problem especially when we want to solve the problem based on a relatively longer time horizon. Therefore, we need to balance the tradeoff between pros and cons when choosing whether to use Almgren Framework or Reinforcement Learning.