Reinforcement Learning for Continuous-Time Optimal Execution: Actor-Critic Algorithm and Error Analysis

Boyu Wang, Xuefeng Gao, Lingfei Li

Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong

Presentation by Aric Cutuli
Department of Industrial Engineering and Operations Research
Columbia University

November 29, 2023

Introduction

- In the execution problem, an agent aims to liquidate or acquire a certain number of shares in a given time horizon
- To achieve optimal scheduling in a continuous-time setting, the agent must choose a trading rate to balance the trade-off between market impact and price uncertainty

Model-based approach: a brief history

- Almgren and Chriss (2000) derive a strategy optimizing variance-adjusted expected execution revenue under linear market impacts
- This paved the way for extensions
  - e.g. generalization of market impact assumptions, variations on price evolution, etc.
- Reliance on model-based stochastic control
  - model-based = model parameters are assumed to be known
- However, estimating market impact models through historical data is difficult (Kyle and Obizhaeva (2018))

An influx of (discrete-time) RL efforts

- Nevmyvaka et al. (2006) conducted a seminal investigation of RL applied to the execution problem using Q-learning
- Ning et al. (2021) developed a double deep Q-learning method and showcased its empirical performance on historical data
- Park and Van Roy (2015) proposed a method for simultaneous execution and learning in a market impact model
- Hambly et al. (2021) applied a policy gradient method for the linear quadratic regulator problem to the Almgren-Chriss (AC) framework
- All these papers are concerned with the discrete-time setting

Problems with discrete-time RL

- Continuous state and action spaces inspire the use of neural networks as approximators of the value function and control policy
  - Requires delicate hyperparameter tuning
  - Convergence issues
  - Interpretation difficulties

Expanding interest of continuous-time RL

- Execution is a high-frequency decision-making problem, making the continuous-time setting natural for studying execution RL algorithms
- Wang et al. (2020) pioneered a continuous-time RL framework
- Wang and Zhou (2020) developed an actor-critic algorithm for continuous-time mean-variance portfolio selection
  - Algorithm is based off an analytically formed value function and exploration distribution
  - Compares favorably with a policy gradient algorithm that relies on neural network approximations
- Developments are ever-growing

Main contributions

- Offline actor-critic algorithm based on the continuous-time AC model and the continuous-time RL framework of Wang et al. (2020)
- Main contributions are threefold
  1. Novel perspective for actor-critic algorithm design in continuous-time RL
  2. Error analysis of the algorithm
  3. Simulation and real-data study to demonstrate the algorithm's nice convergence behavior and out-of-sample performance

Classical AC model in continuous time

- Task is to liquidate $q_0 > 0$ shares within the time horizon $[0, T]$
- Trader's execution strategy is the control process $\nu = (\nu_t)_{t\in[0,T]}$
- Inventory process under ν is $q^\nu = (q^\nu_t)_{t\in[0,T]}$ and satisfies

$$dq^\nu_t = \nu_t dt, \quad t \in [0, T], \quad q^\nu_0 = q_0 \tag{1}$$

- Stock price $S^\nu = (S^\nu_t)_{t\in[0,T]}$ follows an arithmetic Brownian motion (ABM) controlled by the strategy ν through permanent impact function $k(\nu) = \kappa\nu$, where $\kappa > 0$

$$dS^\nu_t = k(\nu_t)dt + \sigma S_0 dW_t, \quad t \in [0, T], \quad S^\nu_0 = S_0 \tag{2}$$

- Cash process of the trader under ν evolves as

$$dx^\nu_t = -\nu_t(S^\nu_t + g(\nu_t))dt, \quad t \in [0, T], \quad x^\nu_0 = x_0 \tag{3}$$

with temporary impact function $g(\nu) = \eta\nu$, where $\eta > 0$

Motivating the mean-quadratic variation (MQV) objective

- Almgren and Chriss (2000) do not use any information regarding the stock price evolution after the start of trading
- The quadratic variation (QV) risk measure

$$\mathbb{E}\left[\int_0^T (q_t^\nu dS_t^\nu)^2\right] = \mathbb{E}\left[\int_0^T \sigma^2 S_0^2 (q_t^\nu)^2 dt\right] \tag{4}$$

captures the volatility path of the portfolio value process $P_t^\nu = x_t^\nu + q_t^\nu S_t^\nu$ since

$$(dP_t^\nu)^2 = (q_t^\nu dS_t^\nu)^2 \tag{5}$$

- Under the MQV objective, the stochastic control problem is time-consistent and measures risk along the entire trading path (Forsyth et al. (2012))

Solution to classical continuous-time AC

- We have the dynamic optimization problem

$$\sup_{\nu \in \mathcal{A}_0(q_0, S_0)} \mathbb{E}\left[\int_0^T \left(-\nu_t(S_t^\nu + \eta\nu_t) - \lambda\sigma^2 S_0^2 (q_t^\nu)^2\right) dt + h_T(q_T^\nu, S_T^\nu) \,\middle|\, q_0^\nu = q_0, S_0^\nu = S_0\right],$$

where $\lambda > 0$ measures risk aversion, $\mathcal{A}_0(q_0, S_0)$ is the set of admissible controls, and

$$h_T(q, S) = \begin{cases} 0, & \text{if } q = 0 \\ -\infty, & \text{otherwise} \end{cases} \tag{6}$$

penalizes inventory not liquidated by time T

- Optimal value function and optimal trading rate function are

$$V^{\text{cl}}(t, q, S) = qS - \frac{q^2}{2}(\kappa + 2\eta K \coth(K(T-t))), \quad \nu^{\text{cl}}(t, q, S) = -qK\coth(K(T-t)), \tag{7}$$

where $K = \sqrt{\frac{\lambda\sigma^2 S_0^2}{\eta}}$

Solution to classical continuous-time AC

- Optimal inventory trajectory is thus

$$q_t^{\nu^{\text{cl}}} = q_0 \frac{\sinh(K(T-t))}{\sinh(KT)}, \quad t \in [0, T] \tag{8}$$

- Subbing (8) into (7), we obtain the optimal trading rate process

$$\nu_t^{\text{cl}} = -q_0 K \frac{\cosh(K(T-t))}{\sinh(KT)}, \quad t \in [0, T],$$

which shows $\lim_{t \to T} \nu_t^{\text{cl}} = -\frac{q_0 K}{\sinh(KT)}$

Towards an RL algorithm

- The three parameters of the AC model (i.e. κ, η, σ) are difficult to estimate empirically (Kyle and Obizhaeva (2018))
- RL instead tries to learn the optimal policy by interacting with the unknown environment through exploration
- The results obtained from formulating and solving the exploratory MQV (EMQV) problem will form the basis for developing RL algorithms

Problem formulation

- To incorporate exploration, we introduce density function π_t to relax ν_t to be a probability distribution at any time t
- Using argument from Wang et al. (2020), we obtain the exploratory version of dynamics (1), (2) and (3) as

$$dq_t^\pi = \int_{\mathbb{R}} \nu \pi_t(\nu) d\nu dt, \quad t \in [0, T], \quad q_0^\pi = q_0 \tag{9}$$

$$dS_t^\pi = \kappa \int_{\mathbb{R}} \nu \pi_t(\nu) d\nu dt + \sigma S_0 dW_t, \quad t \in [0, T], \quad S_0^\pi = S_0 \tag{10}$$

$$dx_t^\pi = \int_{\mathbb{R}} -\nu(S_t^\pi + \eta\nu)\pi_t(\nu) d\nu dt, \quad t \in [0, T], \quad x_0^\pi = x_0 \tag{11}$$

- Overall information gain from exploration is quantified with accumulative Shannon differential entropy

$$\mathcal{H}(\pi) := -\int_0^T \int_{\mathbb{R}} \pi_t(\nu) \ln \pi_t(\nu) d\nu dt$$

Problem Formulation

- Introducing temperature parameter $\zeta \geq 0$, we obtain the EMQV formulation

$$\sup_{\pi \in \mathcal{A}_0(q_0, S_0)} \mathbb{E}\left[\int_0^T \int_{\mathbb{R}} \left(-\nu(S_t^\pi + \eta\nu) - \lambda\sigma^2 S_0^2 (q_t^\pi)^2 - \zeta \ln \pi_t(\nu)\right) \pi_t(\nu) d\nu dt + h_T(q_T^\pi, S_T^\pi) \,\middle|\, q_0^\pi = q_0, S_0^\pi = S_0\right] \tag{12}$$

- To solve the EMQV problem, we define the value function

$$V^\pi(t, q, S) := \mathbb{E}\left[\int_t^T \int_{\mathbb{R}} \left(-\nu(S_u^\pi + \eta\nu) - \lambda\sigma^2 S_0^2 (q_u^\pi)^2 - \zeta \ln \pi_u(\nu)\right) \pi_u(\nu) d\nu du + h_T(q_T^\pi, S_T^\pi) \,\middle|\, q_t^\pi = q, S_t^\pi = S\right] \tag{13}$$

- The optimal value function is

$$V^*(t, q, S) = \sup_{\pi \in \mathcal{A}_t(q,S)} V^\pi(t, q, S)$$

- Solutions to

$$dq_t^\pi = \nu_t^\pi dt. \quad t \in [0, T], \quad q_0^\pi = q_0, \tag{14}$$

$$dS_t^\pi = \kappa\nu_t^\pi dt + \sigma S_0 dW_t. \quad t \in [0, T], \quad S_0^\pi = S_0 \tag{15}$$

give sample trajectories of the inventory and stock price for an action sequence $\{\nu_t^\pi, t \in [0, T]\}$ generated by the control policy π

Policy evaluation for a class of control policies

- Consider a class of feedback controls of the form

$$\pi^f(\nu; t, q, S) = \mathcal{N}(\nu \mid -qf(T-t), c), \quad \forall (t, q, S) \in [0, T] \times \mathbb{R} \times \mathbb{R},$$

where $c > 0$ is constant and $f(T-t)$ is a deterministic function[1] satisfying the following conditions:

(i) f is continuous
(ii) $\lim_{t \to T} f(T-t) = \infty$
(iii) $\int_t^T f(T-u)du = \infty \ \forall t \in [0, T)$
(iv) $\lim_{t \to T} \int_t^T f(T-s)\exp(-\int_t^s f(T-u)du)ds$ is finite
(v) $\int_t^T f^2(T-s)\exp(-2\int_t^s f(T-u)du)ds < \infty \ \forall t \in [0, T)$
(vi) $\lim_{t \to T} \int_t^T f^2(T-s)\exp(-2\int_t^s f(T-u)du)ds = \infty$

- The optimal feedback control distribution for the EMQV problem is in this class

[1] Two example functions are $\coth(T-t)$ and $1/(T-t)$

Policy evaluation for a class of control policies

- Under π^f, the trader's inventory evolves deterministically with dynamics

$$dq_t^{\pi^f} = -q_t^{\pi^f} f(T-t)dt, \quad q_0^{\pi^f} = q_0,$$

which has the unique solution

$$q_t^{\pi^f} = q_0 \exp\left(-\int_0^t f(T-u)du\right), \tag{16}$$

and, from condition (iii),

$$q_T^{\pi^f} = 0 \tag{17}$$

- The stock price dynamics become

$$dS_t^{\pi^f} = -\kappa q_0 f(T-t) \exp\left(-\int_0^t f(T-u)du\right) dt + \sigma S_0 dW_t, \quad S_0^{\pi^f} = S_0$$

Proposition 3.1

The value function V^{π^f} is given by

$$V^{\pi^f}(t,q,S) = qS + \left(\zeta \ln\sqrt{2\pi e c} - \eta c\right)(T-t) - q^2\left(\frac{\kappa}{2} + \int_t^T (\lambda\sigma^2 S_0^2 + \eta f^2(T-s)) exp\left(-2\int_t^s f(T-u)du\right) ds\right)$$

for any $(t,q,S) \in [0,T] \times \mathbb{R} \times \mathbb{R}$

Optimal solution to the EMQV problem

- The optimal value function $V^*(t, q, S)$ satisfies the HJB equation

$$0 = \omega_t + \frac{\sigma^2 S_0^2}{2}\omega_{SS} - \lambda\sigma^2 S_0^2 q^2 + \sup_{\pi \in \mathcal{P}(\mathbb{R})}\left(\int_{\mathbb{R}}((\kappa\omega_S + \omega_q - S)\nu - \eta\nu^2 - \zeta \ln \pi(\nu))\pi(\nu)d\nu\right) \tag{18}$$

with terminal condition

$$\omega(T, q, S) = h_T(q, S) \tag{19}$$

Theorem 3.1

For $\zeta > 0$, (18) is equivalent to

$$0 = \omega_t + \frac{\sigma^2 S_0^2}{2}\omega_{SS} - \lambda\sigma^2 S_0^2 q^2 + \frac{(\kappa\omega_S + \omega_q - S)^2}{4\eta} + \zeta ln\sqrt{\frac{\pi\zeta}{\eta}}.$$

The solution to this PDE with terminal condition (19) is given by

$$\omega(t, q, S) = qS - \frac{q^2}{2}(\kappa + 2\eta K coth(K(T-t))) + \zeta ln\sqrt{\frac{\pi\zeta}{\eta}}(T-t), \tag{20}$$

for any $(t, q, S) \in [0, T] \times \mathbb{R} \times \mathbb{R}$, where $K = \sqrt{\frac{\lambda\sigma^2 S_0^2}{\eta}}$. The maximizer in (18) is given by

$$\pi^*(\nu; t, q, S) = \mathcal{N}\left(\nu \,\middle|\, \frac{\kappa\omega_S + \omega_q - S}{2\eta}, \frac{\zeta}{2\eta}\right) = \mathcal{N}\left(\nu \,\middle|\, -qK coth(K(T-t)), \frac{\zeta}{2\eta}\right) \tag{21}$$

Optimal solution to the EMQV problem

Theorem 3.2

$V^*(t,q,S) = \omega(t,q,S)$ and the optimal feedback control is Gaussian with density function given by $\pi^*(\nu;t,q,S)$. Furthermore, the optimal value function and optimal control of the EMQV problem converge to those of the problem without exploration and entropy regularization as $\zeta \to 0$.

- Similar to Wang and Zhou (2020), we can develop a policy improvement theorem. That is, if we let

$$\tilde{\pi}(\nu;t,q,S) := \mathcal{N}\left(\nu \,\middle|\, \frac{\kappa V_S^{\pi} + V_q^{\pi} - S}{2\eta}, \frac{\zeta}{2\eta}\right), \tag{22}$$

we can show $V^{\tilde{\pi}}(t,q,S) \geq V^{\pi}(t,q,S)$ for any admissible π.

Designing an RL algorithm

- Since we still need the AC model parameters, these analytical results are not implementable
  - Denote them as $\psi_{\text{env}} = (\kappa_{\text{env}}, \eta_{\text{env}}, \sigma^2_{\text{env}})$
- Assuming the environment is described by the AC model, we can develop an actor-critic RL algorithm to directly learn the optimal policy
- The algorithm iteratively applies a policy in the environment to collect samples and then updates the policy
- The analytical results specify a natural parameterization of policy and value function with a small number of parameters
- Convergence is guaranteed under certain conditions
- Neural network parameterizations are large and generally do not guarantee convergence

Parameterization of policy and value function

- Given the form of the optimal feedback control (21), consider the family of distributional feedback controls

$$\pi^{\Phi}(\nu; t, q, S) = \mathcal{N}(\nu \mid -q\varphi_1 \coth(\varphi_1(T-t)), \zeta\varphi_2), \tag{23}$$

which is parameterized by $\Phi := (\varphi_1, \varphi_2)$ for $\varphi_1 > 0$ and $\varphi_2 > 0$

- Calculating the integral in (16) yields

$$q_t^{\Phi} = q_0 \frac{\sinh(\varphi_1(T-t))}{\sinh(\varphi_1 T)}, \quad t \in [0, T] \tag{24}$$

- Applying Proposition 3.1, we obtain the value function of π^{Φ} as

$$\begin{aligned} V^{\pi^{\Phi}}(t,q,S) &= qS + \zeta\left(\ln\sqrt{2\pi e\zeta\varphi_2} - \eta_{\text{env}}\varphi_2\right)(T-t) \\ &- \frac{q^2}{2}\left(\kappa_{\text{env}} + \left(\eta_{\text{env}}\varphi_1 + \frac{\lambda\sigma_{\text{env}}^2 S_0^2}{\varphi_1}\right)\coth(\varphi_1(T-t)) + \left(\eta_{\text{env}}\varphi_1 - \frac{\lambda\sigma_{\text{env}}^2 S_0^2}{\varphi_1}\right)\frac{\varphi_1(T-t)}{\sinh^2(\varphi_1(T-t))}\right) \end{aligned} \tag{25}$$

Parameterization of policy and value function

- We wish to approximate V^{π^Φ} with

$$V^\Theta(t,q,S) := qS + \frac{\zeta}{2}\left(\ln(2\pi e\zeta\varphi_2) - (\theta_2+\theta_3)\frac{\varphi_2}{\varphi_1}\right)(T-t) - \frac{q^2}{2}\left(\theta_1 + \theta_2\coth(\varphi_1(T-t)) + \theta_3\frac{\varphi_1(T-t)}{\sinh^2(\varphi_1(T-t))}\right) \tag{26}$$

for any $(t,q,S)\in[0,T]\times\mathbb{R}\times\mathbb{R}$, and $\Theta := (\theta_1,\theta_2,\theta_3)\in\mathbb{R}^3$

- We want to approach the true parameter vector for the value function of π^Φ, which is

$$\Theta^*(\Phi) := \begin{bmatrix}\theta_1^*(\Phi)\\ \theta_2^*(\Phi)\\ \theta_3^*(\Phi)\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & \varphi_1 & \frac{\lambda S_0^2}{\varphi_1}\\ 0 & \varphi_1 & -\frac{\lambda S_0^2}{\varphi_1}\end{bmatrix}\begin{bmatrix}\kappa_{\text{env}}\\ \eta_{\text{env}}\\ \sigma^2_{\text{env}}\end{bmatrix} \tag{27}$$

- Any Θ, together with Φ, implies an environment parameter $\psi_{\text{imp}} = (\kappa_{\text{imp}},\eta_{\text{imp}},\sigma^2_{\text{imp}})$ through

$$\Theta = M(\Phi)\psi_{\text{imp}}(\Theta;\Phi), \tag{28}$$

where $M(\Phi)$ is the matrix in (27), and

$$\psi_{\text{imp}}(\Theta;\Phi) = M(\Phi)^{-1}\Theta \tag{29}$$

Martingale loss function for policy evaluation

- Jia and Zhou (2022ab) propose martingale loss function for policy iteration
- A feedback policy π^{Φ} of the form given by (23) has entropy

$$\mathcal{H}(\pi^{\Phi}) = -\int_{\mathbb{R}} \pi^{\Phi}(\nu)\ln\pi^{\Phi}(\nu)d\nu = \ln\sqrt{2\pi e\zeta\varphi_2}$$

- By Theorem 1 of Jia and Zhou (2022b), the process $M = (M_t^{\Phi})_{t\in[0,T]}$ is a martingale, where

$$M_t^{\Phi} := V^{\pi^{\Phi}}(t, q_t^{\Phi}, S_t^{\Phi}) + \int_0^t \left(\int_{\mathbb{R}} r_{\text{env}}(\nu)\pi_u^{\Phi}(\nu)d\nu - \lambda\sigma_{\text{env}}^2 S_0^2(q_u^{\Phi})^2 + \zeta\ln\sqrt{2\pi e\zeta\varphi_2}\right) du$$

  with $r_{\text{env}} = -\nu(S_u^{\Phi} + \eta_{\text{env}}\nu)$ being execution revenue received from the AC environment
- We define the martingale loss function $ML(\Theta;\Phi)$ for fixed policy parameter Φ as

$$\begin{aligned} ML(\Theta;\Phi) :&= \mathbb{E}\left[\int_0^T \left(M_T^{\Phi,\Theta} - M_t^{\Phi,\Theta}\right)^2 dt\right] \\ &= \mathbb{E}\Bigg[\int_0^T \Bigg(V^{\Theta}(T, q_T^{\Phi}, S_T^{\Phi}) - V^{\Theta}(t, q_t^{\Phi}, S_t^{\Phi}) \\ &\quad + \int_t^T \left(\int_{\mathbb{R}} r_{\text{env}}(\nu)\pi_u^{\Phi}(\nu)d\nu - \lambda\sigma_{\text{env}}^2 S_0^2(q_u^{\Phi})^2 + \zeta\ln\sqrt{2\pi e\zeta\varphi_2}\right) du\Bigg)^2 dt\Bigg] \end{aligned}$$

- Policy evaluation boils down to minimizing $ML(\Theta;\Phi)$ over Θ

Approximating the martingale loss function

- From (17), we deduce $V^{\Theta}(T, q_T^{\Phi}, S_T^{\Phi}) = 0$
- We approximate $ML(\Theta; \Phi)$ by

$$ML(\Theta;\Phi) \approx \mathbb{E}\left[\sum_{i=0}^{N-1}\Bigg(- V^{\Theta}(t_i, q_{t_i}^{\Phi}, S_{t_i}^{\Phi}) + \zeta(T - t_i)\ln\sqrt{2\pi e\zeta\varphi_2} + \sum_{j=i}^{N-1}\left(\int_{\mathbb{R}} r_{\text{env}}(\nu)\pi_{t_j}^{\Phi}(\nu)d\nu - \lambda\sigma_{\text{env}}^2 S_0^2 (q_{t_j}^{\Phi})^2\right)\Delta t\Bigg)^2 \Delta t\right]$$

- We use Gaussian-Hermite (GH) quadrature to approximate

$$\int_{\mathbb{R}} r_{\text{env}}(\nu)\pi_{t_j}^{\Phi}(\nu)d\nu \approx r_{t_j}^{\Phi} := \frac{1}{\sqrt{\pi}}\sum_{m=1}^{n}\omega_m^{\text{GH}} r_{\text{env}}(\mu_{t_j}^{\Phi} + \sqrt{2}\sigma_{t_j}^{\Phi} y_m^{\text{GH}}),$$

  where $\mu_{t_j}^{\Phi} = -q_{t_j}^{\Phi}\varphi_1\coth(\varphi_1(T - t_j))$, $\sigma_{t_j}^{\Phi} = \sqrt{\zeta\varphi_2}$, and ω_m^{GH} are y_m^{GH} are quadrature weights and abscissas, respectively
- Moreover, we approximate

$$\sigma_{\text{env}}^2 S_0^2 (q_{t_j}^{\Phi})^2\Delta t \approx \left(\Delta P_{t_j}^{\Phi}\right)^2, \quad \Delta P_{t_j}^{\Phi} := r_{t_j}^{\Phi}\Delta t + S_{t_j}^{\Phi}\left(q_{t_{j+1}}^{\Phi} - q_{t_j}^{\Phi}\right) + q_{t_j}^{\Phi}\left(S_{t_{j+1}}^{\Phi} - S_{t_j}^{\Phi}\right)$$

Approximating the martingale loss function

- We now approximate $ML(\Theta;\Phi)$ by

$$ML_{\Delta t}(\Theta;\Phi) := \mathbb{E}\left[\sum_{i=0}^{N-1}\left(-V_i^{\Theta} + \zeta(T-t_i)\ln\sqrt{2\pi e\zeta\varphi_2} + \sum_{j=i}^{N-1}\left(r_{t_j}^{\Phi}\Delta t - \lambda\left(\Delta P_{t_j}^{\Phi}\right)^2\right)\right)^2 \Delta t\right],$$

where $V_i^{\Theta} := V^{\Theta}(t_i, q_{t_i}^{\Phi}, S_{t_i}^{\Phi})$

- This leads to

$$\partial_{\theta_k} ML(\Theta;\Phi) \approx \partial_{\theta_k} ML_{\Delta t}(\Theta;\Phi)$$
$$= \mathbb{E}\left[-2\sum_{i=0}^{N-1}\partial_{\theta_k}V_i^{\Theta}\left(-V_i^{\Theta} + \zeta(T-t_i)\ln\sqrt{2\pi e\zeta\varphi_2} + \sum_{j=i}^{N-1}\left(r_{t_j}^{\Phi}\Delta t - \lambda\left(\Delta P_{t_j}^{\Phi}\right)^2\right)\right)\Delta t\right], \tag{30}$$

where for $i = 0, \dots, N-1$,

$$\partial_{\theta_1} V_i^{\Theta} = -\frac{(q_{t_i}^{\Phi})^2}{2}, \tag{31}$$

$$\partial_{\theta_2} V_i^{\Theta} = -\frac{(q_{t_i}^{\Phi})^2}{2}\coth(\varphi_1(T-t_i)) - \frac{\zeta\varphi_2}{2\varphi_1}(T-t_i), \tag{32}$$

$$\partial_{\theta_3} V_i^{\Theta} = -\frac{(q_{t_i}^{\Phi})^2}{2}\frac{\varphi_1(T-t_i)}{\sinh^2(\varphi_1(T-t_i))} - \frac{\zeta\varphi_2}{2\varphi_1}(T-t_i) \tag{33}$$

Approximating the policy gradient

- Let $G(\Phi) := V^{\pi^{\Phi}}(0, q_0, S_0)$ as it's given in (25). We can directly calculate

$$\nabla_{\Phi} G(\Phi) = \begin{bmatrix} \partial_{\varphi_1} G(\Phi) \\ \partial_{\varphi_2} G(\Phi) \end{bmatrix} = \begin{bmatrix} -\frac{q_0^2}{2}\left(\eta_{\text{env}}\varphi_1 - \frac{\lambda\sigma_{\text{env}}^2 S_0^2}{\varphi_1}\right) g(\varphi_1) \\ \zeta T \left(\frac{1}{2\varphi_2} - \eta_{\text{env}}\right) \end{bmatrix}, \tag{34}$$

where

$$g(\varphi_1) := \frac{\coth(\varphi_1 T)}{\varphi_1} + \frac{T}{\sinh^2(\varphi_1 T)} - \frac{2\varphi_1 T^2}{\sinh^2(\varphi_1 T)\tanh(\varphi_1 T)} \tag{35}$$

- Since (34) contains unknown environment parameters, we need to replace them with their implied counterparts to approximate the policy gradient, i.e.

$$\nabla_{\Phi} G(\Phi) \approx \nabla_{\Phi} G(\Phi; \Theta) := \begin{bmatrix} -\frac{q_0^2}{2}\left(\eta_{\text{imp}}\varphi_1 - \frac{\lambda\sigma_{\text{imp}}^2 S_0^2}{\varphi_1}\right) g(\varphi_1) \\ \zeta T \left(\frac{1}{2\varphi_2} - \eta_{\text{imp}}\right) \end{bmatrix} \overset{(29)}{=} \begin{bmatrix} -\frac{q_0^2\theta_3}{2} g(\varphi_1) \\ \zeta T \left(\frac{1}{2\varphi_2} - \frac{\theta_2+\theta_3}{2\varphi_1}\right) \end{bmatrix} \tag{36,37}$$

The EMQV algorithm

- Start with initial guesses $\Theta^{(0)}$ and $\Phi^{(0)}$

1. **PE update:** Update $\Theta^{(\ell)}$ to $\tilde{\Theta}^{(\ell)}$ by gradient descent as

$$\tilde{\theta}_k^{(\ell)} = \theta_k^{(\ell)} - \partial_{\theta_k} ML_{\Delta t}(\Theta^{(\ell)}; \Phi^{(\ell)})/\alpha_\theta^{(\ell)}, \quad k = 1, 2, 3,$$

   where $1/\alpha_\theta^{(\ell)}$ is the learning rate for iteration ℓ

2. **PG update:** Update policy parameters by gradient ascent as

$$\varphi_k^{(\ell+1)} = \varphi_k^{(\ell)} + \partial_{\varphi_k} G(\Phi^{(\ell)}; \tilde{\Theta}^{(\ell)})/\alpha_\varphi^{(\ell)}, \quad k = 1, 2, \tag{38}$$

   where $1/\alpha_\varphi^{(\ell)}$ is another learning rate

3. **Recalibration (RC):** To ensure the estimated value function moves in lockstep with the policy update, we recalibrate $\Theta^{(\ell+1)}$ via

$$\Theta^{(\ell+1)} = M(\Phi^{(\ell+1)})M(\Phi^{(\ell)})^{-1}\tilde{\Theta}^{(\ell)}$$

The EMQV algorithm

- **Collection of samples from the environment:** In iteration ℓ, generate multiple episodes by interacting with the environment to collect samples. In episode m and at $t_i = i\Delta t$, collect a sample $(t_i, q_{t_i}^{\ell,m}, S_{t_i}^{\ell,m}, r_{t_i}^{\ell,m}, \Delta P_{t_i}^{\ell,m})$ using the control $\pi_{t_i}^{\Phi^{(\ell)}} = \mathcal{N}(\cdot \mid \mu_{t_i}^{(\ell)}, (\sigma_{t_i}^{(\ell)})^2)$, where

$$\mu_{t_i}^{(\ell)} = -q_{t_i}^{\ell,m}\varphi_1^{(\ell)}\coth(\varphi_1^{(\ell)}(T - t_i)), \quad (\sigma_{t_i}^{(\ell)})^2 = \zeta\varphi_2^{(\ell)}.$$

Collect trajectories for the exploratory state process from the environment.

1. Collect exploratory execution revenue $r_{t_i}^{\ell,m}\Delta t$ by calculating trading rate $\nu_j = \mu_{t_i}^{(\ell)} + \sqrt{2}\sigma_{t_i}^{(\ell)} y_j^{\mathrm{GH}}$, sending an order of size $\nu_j\Delta t$ and receiving revenue $r_{\mathrm{env}}(\nu_j)\Delta t$, doing this n times, and calculating $r_{t_i}^{\ell,m}\Delta t = \frac{1}{\sqrt{\pi}}\sum_{j=1}^{n}\omega_j^{\mathrm{GH}} r_{\mathrm{env}}(\nu_j)\Delta t$
2. Collect quadratic variation $\Delta P_{t_i}^{\ell,m}$ by sending an order of size $\mu_{t_i}^{(\ell)}$, observing $S_{t_{t+i}}^{\ell,m}$, updating $q_{t_{i+1}}^{\ell,m} = q_{t_i}^{\ell,m} + \mu_{t_i}^{(\ell)}\Delta t$, and calculating

$$\Delta P_{t_i}^{\ell,m} = r_{t_i}^{\ell,m}\Delta t + S_{t_i}^{\ell,m}\left(q_{t_{i+1}}^{\ell,m} - q_{t_i}^{\ell,m}\right) + q_{t_i}^{\ell,m}\left(S_{t_{i+1}}^{\ell,m} - S_{t_i}^{\ell,m}\right). \quad (39)$$

Algorithm 1: EMQV algorithm for the optimal execution problem.

Input: Environment *Env*, initial price S_0, initial inventory q_0, execution horizon T, timestep Δt, risk-aversion parameter λ, temperature parameter ζ, abscissas $y_1^{\mathrm{GH}}, \ldots, y_n^{\mathrm{GH}}$ and weights $w_1^{\mathrm{GH}}, \ldots, w_n^{\mathrm{GH}}$ of the GH quadrature, number of training iterations L, number of episodes M;

Initialize $\Theta^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \theta_3^{(0)})$ and $\Phi^{(0)} = (\varphi_1^{(0)}, \varphi_2^{(0)})$;

for *training iterations* $\ell = 0, \ldots, L$ **do**
- **for** *episodes* $m = 0, \ldots, M$ **do**
  - **for** *time steps* $i = 0, \ldots, \lfloor \frac{T}{\Delta t} \rfloor$ **do**
    - Calculate $\mu_{t_i}^{(\ell)} = -q_{t_i}^{\ell,m} \varphi_1^{(\ell)} \coth(\varphi_1^{(\ell)}(T - t_i))$ and $(\sigma_{t_i}^{(\ell)})^2 = \zeta \varphi_2^{(\ell)}$;
    - **for** $j = 0, \ldots, n$ **do**
      - Calculate $\nu_j = \mu_{t_i}^{(\ell)} + \sqrt{2}\sigma_{t_i}^{(\ell)} y_j^{\mathrm{GH}}$, send $\nu_j \Delta t$, and receive $r_{\mathrm{env}}(\nu_j)$;
      - Reset *Env* to t_i;
    - **end**
    - Calculate $r_{t_i}^{\ell,m} = \frac{1}{\sqrt{\pi}} \sum_{j=1}^n \omega_j^{\mathrm{GH}} r_{\mathrm{env}}(\nu_j)$;
    - Send $\mu_{t_i}^{(\ell)}$, observe $S_{t_{i+1}}^{\ell,m}$, update $q_{t_{i+1}}^{\ell,m} = q_{t_i}^{\ell,m} + \mu_{t_i}^{(\ell)} \Delta t$, and calculate $\Delta P_{t_i}^{\ell,m}$;
  - **end**
- **end**
- **Policy evaluation:**
- Calculate $\partial_{\theta_k} ML_{\Delta t}(\Theta^{(\ell)}; \Phi^{(\ell)}) = \frac{1}{M} \sum_{m=1}^M \partial_{\theta_k} ML_{\Delta t}(\Theta^{(\ell)}; \Phi^{(\ell)})^{(m)}$ and $\tilde{\theta}_k^{(\ell)} = \theta_k^{(\ell)} - \partial_{\theta_k} ML_{\Delta t}(\Theta^{(\ell)}; \Phi^{(\ell)}) / \alpha_\theta^{(\ell)}$ for $k = 1, 2, 3$;
- **Policy gradient:**
- Calculate $\partial_{\varphi_k} G(\Phi^{(\ell)}; \tilde{\Theta}^{(\ell)}) = \frac{1}{M} \sum_{m=1}^M \partial_{\varphi_k} G(\Phi^{(\ell)}; \tilde{\Theta}^{(\ell)})^{(m)}$ and update $\varphi_k^{(\ell+1)} = \varphi_k^{(\ell)} + \partial_{\varphi_k} G(\Phi^{(\ell)}; \tilde{\Theta}^{(\ell)}) / \alpha_\varphi^{(\ell)}$ for $k = 1, 2$;
- **Recalibration:**
- Update $\Theta^{(\ell+1)} = M(\Phi^{(\ell+1)}) M(\Phi^{(\ell)})^{-1} \tilde{\Theta}^{(\ell)}$;

end

Output: Learned policy $\pi^{\Phi^{(L)}}(\nu \mid t, q, S) = \mathcal{N}(\nu \mid -q\varphi_1^{(L)} \coth(\varphi_1^{(L)}(T - t)), \zeta \varphi_2^{(L)})$

Next step: convergence analysis for EMQV

- We will be analyzing convergence for the EMQV algorithm
- First, we will obtain useful properties of the exact martingale loss function
- We will then analyze the convergence of exact PG, which will motivate convergence analysis for EMQV

Analytical formula of the martingale loss function

Lemma 5.1

The martingale loss function, its gradient and Hessian matrix are given by

$$\begin{aligned} ML(\Theta;\Phi) &= \gamma(\Phi) + \rho(\Theta;\Phi)^\intercal I(\Phi)\rho(\Theta;\Phi), \\ \nabla_\Theta ML(\Theta;\Phi) &= 2A(\Phi)^\intercal I(\Phi)\rho(\Theta;\Phi), \\ \nabla^2_\Theta ML(\Theta;\Phi) &= 2A(\Phi)^\intercal I(\Phi)A(\Phi), \end{aligned} \tag{40}$$

where $\gamma(\Phi) = \frac{\sigma_{env}^2 q_0^2 S_0^2}{\varphi_1 sinh^2(\varphi_1 T)}\left(\frac{cosh(2\varphi_1 T)-1}{8\varphi_1} - \frac{\varphi_1 T^2}{4}\right)$, $\rho(\Theta;\Phi) = A(\Phi)\Theta + b_{env}(\Phi)$ *with*

$$A(\Phi) = \begin{bmatrix} 0 & \frac{\zeta\varphi_2}{2\varphi_1} & \frac{\zeta\varphi_2}{2\varphi_1} + \frac{\varphi_1 q_0^2}{2sinh^2(\varphi_1 T)} \\ \frac{q_0^2}{2sinh^2(\varphi_1 T)} & 0 & 0 \\ 0 & \frac{q_0^2}{4sinh^2(\varphi_1 T)} & 0 \end{bmatrix}, \quad b_{env}(\Phi) = \begin{bmatrix} -\zeta\eta_{env}\varphi_2 - \frac{q_0^2(\eta_{env}\varphi_1^2 - \lambda\sigma_{env}^2 S_0^2)}{2sinh^2(\varphi_1 T)} \\ \frac{\kappa_{env} q_0^2}{2sinh^2(\varphi_1 T)} \\ \frac{q_0^2(\eta_{env}\varphi_1^2 + \lambda\sigma_{env}^2 S_0^2)}{4\varphi_1 sinh^2(\varphi_1 T)} \end{bmatrix}, \tag{41}$$

and

$$I(\Phi) = \begin{bmatrix} I_1 & I_4 & I_5 \\ I_4 & I_2 & I_6 \\ I_5 & I_6 & I_3 \end{bmatrix}, \tag{42}$$

with

$$\begin{aligned} I_1 &= \frac{1}{3}T^3, & I_2 &= \frac{sinh(4\varphi_1 T)}{32\varphi_1} - \frac{sinh(2\varphi_1 T)}{4\varphi_1} + \frac{3}{8}T, \\ I_3 &= \frac{sinh(4\varphi_1 T)}{8\varphi_1} - \frac{1}{2}T, & I_4 &= \frac{T sinh(2\varphi_1 T)}{4\varphi_1} + \frac{1 - cosh(2\varphi_1 T)}{8\varphi_1^2} - \frac{1}{4}T^2, \\ I_5 &= \frac{T cosh(2\varphi_1 T)}{2\varphi_1} - \frac{sinh(2\varphi_1 T)}{4\varphi_1^2}, & I_6 &= \frac{sinh^4(\varphi_1 T)}{2\varphi_1}. \end{aligned}$$

Furthermore, $ML(\Theta^*(\Phi);\Phi) = \gamma(\Phi)$ *and* $\nabla_\Theta ML(\Theta^*(\Phi);\Phi) = \mathbf{0}$.

Useful properties of the Hessian

- Let $S(\Phi) := A(\Phi)^\intercal I(\Phi) A(\Phi)$, $L(\Phi) := 2\|S(\Phi)\|_\infty$, and $\mu(\Phi) := 2\lambda_{\min}(S(\Phi))$

Lemma 5.2

The matrix $S(\Phi)$ is positive definite and $0 < \lambda_{min}(S(\Phi)) < \lambda_{max}(S(\Phi)) \leq \|S(\Phi)\|_\infty$. Thus, $\mu(\Phi) \in (0, L(\Phi))$.

- In the ensuing analysis, we explore how the gradient steps in PE and PG influence the performance gap between our algorithm and exact updates

Convergence of the exact PG

- Denote the optimal policy parameters by

$$\Phi^* = (\varphi_1^*, \varphi_2^*), \quad \varphi_1^* = \sqrt{\lambda \sigma_{\text{env}}^2 S_0^2 / \eta_{\text{env}}}, \quad \varphi_2^* = 1/(2\eta_{\text{env}}) \tag{43}$$

- In the exact PG update,

$$\Phi^{(\ell+1)} = \Phi^{(\ell)} + \nabla_\Phi G(\Phi^{(\ell)})/\alpha_\varphi^{(\ell)}. \tag{44}$$

Lemma 5.3

For any fixed $\underline{\varphi}_2 > 0$*, there exists a positive constant* L_G *independent of* Φ *such that for any* $\varphi_1, \varphi_1' > 0$ *and* $\varphi_2, \varphi_2' > \underline{\varphi}_2$*,*

$$-G(\Phi') \leq -G(\Phi) - \nabla_\Phi G(\Phi)^\intercal (\Phi' - \Phi) + \frac{L_G}{2} \|\Phi' - \Phi\|_2^2.$$

Lemma 5.4

For any fixed $0 < \underline{\varphi}_1 < \bar{\varphi}_1 < \infty$, $0 < \underline{\varphi}_2 < \bar{\varphi}_2 < \infty$ *such that* $\Phi^* \in \mathcal{C}_\Phi := [\underline{\varphi}_1, \underline{\varphi}_2] \times [\bar{\varphi}_1, \bar{\varphi}_2]$, $G(\Phi)$ *satisfies the local Polyak-Łojasiewicz (PŁ) condition on* $\Phi \in \mathcal{C}_\Phi$*; i.e., there exists a positive constant* μ_G *independent of* Φ *such that for any* $\Phi \in \mathcal{C}_\Phi$*,*

$$\frac{1}{2} \| - \nabla_\Phi G(\Phi)\|_2^2 \geq \mu_G (G(\Phi^*) - G(\Phi)).$$

Furthermore, we can always choose L_G *to be greater than* μ_G*.*

Estimate of the performance gap for exact PG

Theorem 5.1

For any fixed $\mathcal{C}_\Phi := [\underline{\varphi}_1, \underline{\varphi}_2] \times [\bar{\varphi}_1, \bar{\varphi}_2]$ *with* $0 < \underline{\varphi}_1 < \varphi_1^* < \bar{\varphi}_1 < \infty$ *and* $0 < \underline{\varphi}_2 < \varphi_2^* < \bar{\varphi}_2 < \infty$, *and the exact PG update scheme (44), if* $\Phi^{(0)} \in \mathcal{C}_\Phi$ *and*

$$\alpha_\varphi^{(\ell)} > max\left\{ \frac{\partial_{\varphi_1} G(\Phi^{(\ell)})}{\bar{\varphi}_1 - \varphi_1^{(\ell)}}, \frac{\partial_{\varphi_1} G(\Phi^{(\ell)})}{\underline{\varphi}_1 - \varphi_1^{(\ell)}}, \frac{\partial_{\varphi_2} G(\Phi^{(\ell)})}{\bar{\varphi}_2 - \varphi_2^{(\ell)}}, \frac{\partial_{\varphi_2} G(\Phi^{(\ell)})}{\underline{\varphi}_2 - \varphi_2^{(\ell)}} \right\} := c_{PG}(\Phi^{(\ell)}, \nabla_\Phi G(\Phi^{(\ell)})) \tag{45}$$

for any $\ell = 0, 1, \ldots$, *then* $\Phi^{(\ell)} \in \mathcal{C}_\Phi$ *for all* ℓ, *and the performance gap satisfies*

$$G(\Phi^*) - G(\Phi^{(\ell+1)}) \le \left(1 - C_{\varphi,1}^{(\ell)}\right) \left(G(\Phi^*) - G(\Phi^{(\ell)})\right),$$

where

$$C_{\varphi,1}^{(\ell)} = \mu_G \left(2\alpha_\varphi^{(\ell)} - L_G\right) / (\alpha_\varphi^{(\ell)})^2, \tag{46}$$

and L_G, μ_G *are positive constants in Lemma 5.3 and Lemma 5.4. If, in addition to (45),*

$$\alpha_\varphi^{(\ell)} > L_G/2, \tag{47}$$

then

$$0 < C_{\varphi,1}^{(\ell)} \le \mu_G / L_G < 1, \tag{48}$$

and hence the linear convergence of the exact PG iterations.

Error analysis of one-step PE

Lemma 5.5

For $\Phi^{(\ell)}$ and $\Theta^{(\ell)}$, after the one-step PE update

$$\tilde{\Theta}^{(\ell)} = \Theta^{(\ell)} - \nabla_{\Theta} ML(\Theta^{(\ell)}; \Phi^{(\ell)})/\alpha_\theta^{(\ell)}, \tag{49}$$

with $\alpha_\theta^{(\ell)} > L(\Phi^{(\ell)})/2$, we have

$$\|\tilde{\Theta}^{(\ell)} - \Theta^*(\Phi^{(\ell)})\|_2 \leq \Lambda(\alpha_\theta^{(\ell)}, \Phi^{(\ell)})\|\Theta^{(\ell)} - \Theta^*(\Phi^{(\ell)})\|_2, \tag{50}$$

where

$$\Lambda(\alpha_\theta, \Phi) = \begin{cases} 2\lambda_{max}(S(\Phi))/\alpha_\theta - 1 & \text{if } L(\Phi)/2 < \alpha_\theta < \lambda_{max}(S(\Phi)) + \lambda_{min}(S(\Phi)), \\ 1 - 2\lambda_{min}(S(\Phi))/\alpha_\theta & \text{if } \alpha_\theta \geq \lambda_{max}(S(\Phi)) + \lambda_{min}(S(\Phi)). \end{cases} \tag{51}$$

We also have

$$0 < \Lambda(\alpha_\theta, \Phi) \leq 1 - \varepsilon_\Lambda \quad \text{for any } \Phi \in \mathcal{C}_\Phi, \tag{52}$$

where $\varepsilon_\Lambda \in (0,1)$ is a constant independent of Φ but dependent of the chosen $\mathcal{C}_\Phi$.

Error analysis of one-step PG

Lemma 5.6

Suppose $\Phi^*, \Phi^{(\ell)} \in \mathcal{C}_\Phi$. After one-step PG (38) using the approximate policy gradient $\nabla_\Phi G(\Phi^{(\ell)}; \tilde{\Theta}^{(\ell)})$, if

$$\alpha_\varphi^{(\ell)} > max\left\{ c_{PG}(\Phi^{(\ell)}, \nabla_\Phi G(\Phi^{(\ell)}; \tilde{\Theta}^{(\ell)})), L_G/2 \right\},$$

we have $\Phi^{(\ell+1)} \in \mathcal{C}_\Phi$ and

$$G(\Phi^*) - G(\Phi^{(\ell+1)}) \leq \left(1 - C_{\varphi,1}^{(\ell)}\right)\left(G(\Phi^*) - G(\Phi^{(\ell)})\right) + C_{\varphi,2}\|\Delta\tilde{\Theta}^{(\ell)}(\Phi^{(\ell)})\|_2^2 + C_{\varphi,3}\|\Delta\tilde{\Theta}^{(\ell)}(\Phi^{(\ell)})\|_2,$$

where $\Delta\tilde{\Theta}^{(\ell)}(\Phi^{(\ell)}) := \tilde{\Theta}^{(\ell)} - \Theta^*(\Phi^{(\ell)})$, $C_{\varphi,1}^{(\ell)}$ is defined in (46) and satisfies (48), and $C_{\varphi,2}$ and $C_{\varphi,3}$ are positive constants independent of ℓ.

Error analysis of RC

Lemma 5.7

After the RC step, if

$$\alpha_{\varphi}^{(\ell)} > max\left\{ c_{RC}(\varepsilon, \Phi^{(\ell)}, \tilde{\Theta}^{(\ell)}), c_{PG}(\Phi^{(\ell)}, \nabla_{\Phi} G(\Phi^{(\ell)}, \tilde{\Theta}^{(\ell)})) \right\}$$

and

$$c_{RC}(\varepsilon, \Phi^{(\ell)}, \tilde{\Theta}^{(\ell)}) := \frac{\Lambda(\alpha_{\theta}^{(\ell)}, \Phi^{(\ell)}) \mathbb{1}_{\{\tilde{\theta}_3^{(\ell)} \le 0\}} - (1-\varepsilon) \mathbb{1}_{\{\tilde{\theta}_3^{(\ell)} > 0\}}}{\varphi_1^{(\ell)} \left(1 - \varepsilon - \Lambda(\alpha_{\theta}^{(\ell)}, \Phi^{(\ell)}) \right)} \partial_{\varphi_1} G(\Phi^{(\ell)}, \tilde{\Theta}^{(\ell)}) \tag{53}$$

for any fixed $\varepsilon \in (0, \varepsilon_{\Lambda})$, *we have*

$$\|\Theta^{(\ell+1)} - \Theta^*(\Phi^{(\ell+1)})\|_2 \le d(\ell) \|\Theta^{(\ell)} - \Theta^*(\Phi^{(\ell)})\|_2, \tag{54}$$

where

$$d(\ell) := max\left\{ 1, \varphi_1^{(\ell+1)}/\varphi_1^{(\ell)}, \varphi_1^{(\ell)}/\varphi_1^{(\ell+1)} \right\} \Lambda(\alpha_{\theta}^{(\ell)}, \Phi^{(\ell)}) \in (0, 1-\varepsilon). \tag{55}$$

Error bound for the performance gap of the EMQV algorithm

Theorem 5.2

Suppose $\Phi^*, \Phi^{(0)} \in \mathcal{C}_\Phi$ *and assume the following condition is satisfied for* $\ell = 0, 1, \ldots$.

Condition 1: $\alpha_\theta^{(\ell)} > L(\Phi^{(\ell)})/2$, *and for any fixed* $0 < \varepsilon < \varepsilon_\Lambda$,

$$\alpha_\varphi^{(\ell)} > max\left\{c_{PG}(\Phi^{(\ell)}, \nabla_\Phi G(\Phi^{(\ell)}, \tilde{\Theta}^{(\ell)})), L_G/2, c_{RC}(\varepsilon, \Phi^{(\ell)}, \tilde{\Theta}^{(\ell)})\right\}. \tag{56}$$

Then $\Phi^{(\ell)} \in \mathcal{C}_\Phi$ *for all* ℓ *and*

$$G(\Phi^*) - G(\Phi^{(\ell+1)}) \leq (G(\Phi^*) - G(\Phi^{(0)}))E(\ell+1) + C_{\varphi,2}(\Delta\Theta_0)^2(E \circledast D^2)(\ell) + C_{\varphi,3}\Delta\Theta_0(E \circledast D)(\ell), \tag{57}$$

where

$$E(\varrho) = \prod_{\iota=0}^{\varrho-1}\left(1 - C_{\varphi,1}^{(\ell-\iota)}\right), \quad \varrho = 0, 1, \ldots, \ell+1, \tag{58}$$

$$D(\ell) = \Lambda(\alpha_\theta^{(\ell)}, \Phi^{(\ell)}) \prod_{\iota=0}^{\ell-1} d(\iota) \in (0, (1-\varepsilon)^\ell), \tag{59}$$

with $(E \circledast D)(\ell) = \sum_{\varrho=0}^{\ell} E(\ell-\varrho)D(\varrho)$ *and* $\Delta\Theta_0 := \|\Theta^{(0)} - \Theta^*(\Phi^{(0)})\|_2$.

Interpretation of the performance gap

- The upper bound on the performance gap is the sum of three parts
  - The first is due to PG error, which converges to 0
  - The last two are due to PE error, and the summands coverge to 0
- However, the PE error accumulates with respect to ℓ
- To obtain convergence of the algorithm, we must ensure $\alpha_{\varphi}^{(\ell)}$ is also small enough to overcome the cumulative impact of PE error

Convergence of the EMQV algorithm

Theorem 5.3

Assume the following condition holds in addition to Condition 1.
Condition 2: *For any fixed ε and $\bar{\varepsilon}$ such that $0 < \bar{\varepsilon} < \varepsilon < \varepsilon_\Lambda < 1$,*

$$\alpha_\varphi^{(\ell)} > c_{CVG}(\varepsilon, \bar{\varepsilon}) := \frac{\mu_G + \sqrt{\mu_G(\mu_G - \beta_{\varepsilon,\bar{\varepsilon}} L_G)}}{\beta_{\varepsilon,\bar{\varepsilon}}} \mathbb{1}_{\{\mu_G > \beta_{\varepsilon,\bar{\varepsilon}} L_G\}}, \tag{60}$$

where $\beta_{\varepsilon,\bar{\varepsilon}} := (\varepsilon - \bar{\varepsilon})/(1 - \bar{\varepsilon}) \in (0, 1)$.
Then, the sequence of performance gaps $\{G(\Phi^{(\ell)}) - G(\Phi^*),\ \ell = 0, 1, \ldots\}$ exhibits linear convergence to zero. Furthermore, $\lim_{\ell\to\infty} \Phi^{(\ell)} = \Phi^*$ and $\lim_{\ell\to\infty} \Theta^{(\ell)} = \Theta^*(\Phi^*)$.

- (60) implies that for any ℓ,

$$0 < (1 - \varepsilon)/(1 - C_{\varphi,1}^{(\ell)}) < 1 - \bar{\varepsilon} < 1. \tag{61}$$

Using an AC simulator to verify convergence

- We verify the convergence of the EMQV algorithm and analyze the effect of recalibration and the choice of the state process on the algorithm's convergence
- We also compare EMQV with the soft actor-critic (SAC) algorithm (Haarnoja et al. (2018)), which uses deep learning
- We choose 3 different levels of risk aversion:

$$\lambda^{\text{Low}} = \eta_{\text{env}}, \quad \lambda^{\text{Mid}} = 10^3 \times \eta_{\text{env}}, \quad \lambda^{\text{High}} = 10^4 \times \eta_{\text{env}}$$

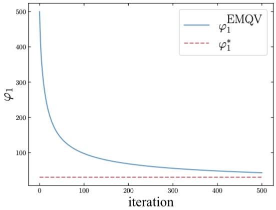
- We specify AC model parameters

S_0	q_0	T (day)	κ_{env}	η_{env}	σ_{env}
100	5×10^5	1	2.5×10^{-7}	2.5×10^{-6}	30%

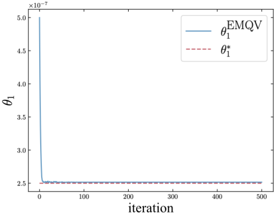
- Optimal parameters should be given by

$$\varphi_1^* = \sqrt{\lambda \sigma_{\text{env}}^2 S_0^2 / \eta_{\text{env}}}, \quad \theta_1^* = \kappa_{\text{env}}, \quad \theta_2^* = \varphi_1^* \eta_{\text{env}} + \lambda S_0^2 \sigma_{\text{env}}^2 / \varphi_1^*, \quad \theta_3^* = 0$$

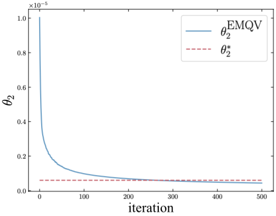
Training results of EMQV for low risk-aversion λ^{Low}



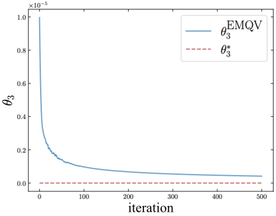
(a) Learning path of φ_1



(b) Learning path of θ_1

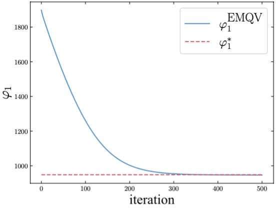


(c) Learning path of θ_2

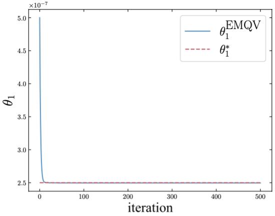


(d) Learning path of θ_3

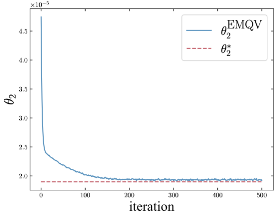
Training results of EMQV for medium risk-aversion λ^{Mid}



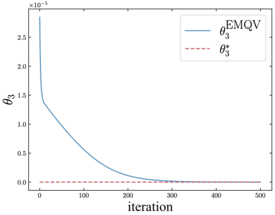
(a) Learning path of φ_1



(b) Learning path of θ_1

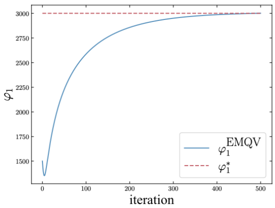


(c) Learning path of θ_2

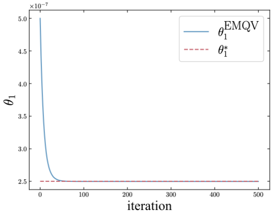


(d) Learning path of θ_3

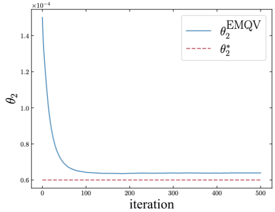
Training results of EMQV for high risk-aversion λ^{High}



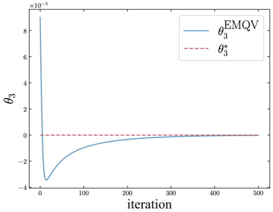
(a) Learning path of φ_1



(b) Learning path of θ_1

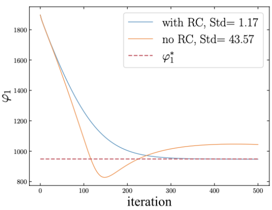


(c) Learning path of θ_2

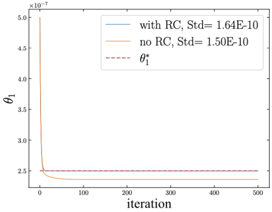


(d) Learning path of θ_3

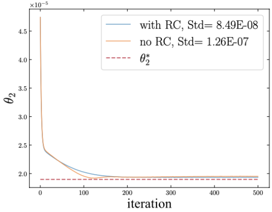
Training results of EMQV with and without recalibration for λ^{Mid}



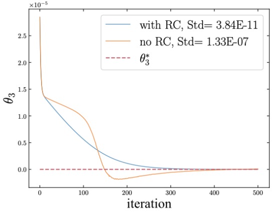
(a) Average learning path of φ_1



(b) Average learning path of θ_1

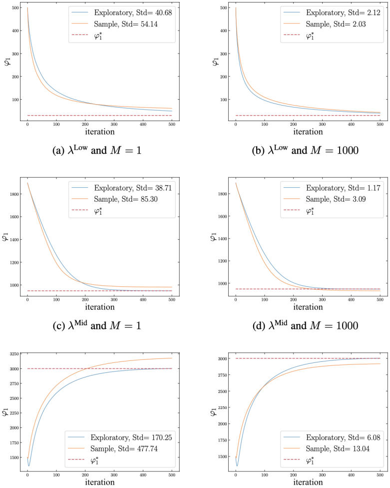


(c) Average learning path of θ_2



(d) Average learning path of θ_3

Learning paths of φ_1 from sample and exploratory trajectories



(a) λ^{Low} and $M = 1$ (b) λ^{Low} and $M = 1000$

(c) λ^{Mid} and $M = 1$ (d) λ^{Mid} and $M = 1000$

(e) λ^{High} and $M = 1$ (f) λ^{High} and $M = 1000$

Performance gaps of EMQV and SAC

- We compare out-of-sample testing performance measured by QV-adjusted PnL of the policy π learned by the algorithm

$$\mathrm{PnL}^{\pi} := \mathbb{E}[x_T^{\pi}] - \lambda \mathbb{E}\left[\int_0^T (q_t^{\pi} dS_t^{\pi})^2\right]$$

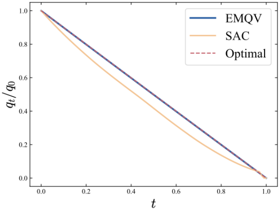
- The performance gap (in basis points) between policy π and the optimal one is

$$\Delta \mathrm{PnL}^{\pi} := \frac{\mathrm{PnL}^{\pi} - \mathrm{PnL}^{*}}{\mathrm{PnL}^{*}} \times 10^4$$

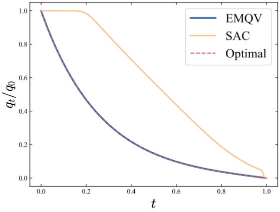
- Average performance gaps over 10^5 episodes

	λ^{Low}	λ^{Mid}	λ^{High}
$\Delta\mathrm{PnL}^{\mathrm{EMQV}}$	-7.490×10^{-4}	-1.022×10^{-3}	-3.075×10^{-4}
$\Delta\mathrm{PnL}^{\mathrm{SAC}}$	-19.311	-169.754	-185.475

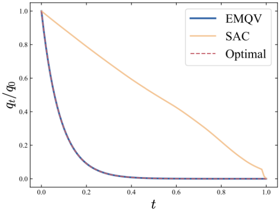
Inventory processes of EMQV and SAC



(a) Low risk-aversion λ^{Low}



(b) Medium risk-aversion λ^{Mid}



(c) High risk-aversion λ^{High}

Performance in market simulation

- We train the EMQV algorithm on two market simulators built using real data
  1. HD: Uses historical limit order book and order flow data without any model assumption
  2. CST: Stochastic model of order book dynamics (Cont et al. (2010))
- An algorithm's performance is measured by its improvement over TWAP
- Let AC-EC denote the classical control approach

Out-of-sample test results

	AAPL	BMY	CVX	DIS	FB	MSFT	PG
λ^{Low}							
$\Delta\text{PnL}^{\text{EMQV}}$ (HD)	−0.020	**0.017**	**0.015**	−0.032	**0.046**	−0.019	**0.001**
	(0.01)	(0.10)	(0.03)	(0.02)	(0.11)	(0.01)	(0.05)
$\Delta\text{PnL}^{\text{EMQV}}$ (CST)	−0.011	−0.000	−0.003	**−0.000**	−0.010	−0.005	0.000
	(0.01)	(0.00)	(0.00)	(0.00)	(0.01)	(0.00)	(0.00)
$\Delta\text{PnL}^{\text{SAC}}$ (HD)	−1.16E3	−4.958	−12.566	−6.212	−790.351	−1.884	−2.094
	(56.55)	(26.90)	(22.19)	(16.27)	(69.85)	(15.68)	(14.71)
$\Delta\text{PnL}^{\text{SAC}}$ (CST)	−120.740	−2.752	−2.766	−2.691	−297.374	−1.516	−1.895
	(34.58)	(26.68)	(22.32)	(16.26)	(48.26)	(15.60)	(14.70)
$\Delta\text{PnL}^{\text{AC-EC}}$	**−0.001**	−0.000	0.000	−0.000	−0.001	**−0.001**	−0.000
	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)	(0.00)
λ^{Mid}							
$\Delta\text{PnL}^{\text{EMQV}}$ (HD)	**15.369**	2.339	1.766	0.506	7.994	5.202	0.281
	(2.83)	(0.64)	(0.31)	(0.29)	(1.77)	(0.88)	(0.09)
$\Delta\text{PnL}^{\text{EMQV}}$ (CST)	14.661	0.161	1.689	0.023	**18.954**	7.445	0.894
	(2.31)	(0.03)	(0.29)	(0.01)	(3.74)	(1.29)	(0.33)
$\Delta\text{PnL}^{\text{SAC}}$ (HD)	−1.25E5	−0.367	−44.855	−36.189	−9.80E3	−27.604	−0.095
	(5813.98)	(29.31)	(35.70)	(26.52)	(1192.00)	(20.10)	(14.73)
$\Delta\text{PnL}^{\text{SAC}}$ (CST)	−6.91E3	**4.260**	**7.360**	**1.397**	−9.20E3	**7.751**	**1.169**
	(772.96)	(27.31)	(22.54)	(16.36)	(1083.18)	(22.34)	(14.72)
$\Delta\text{PnL}^{\text{AC-EC}}$	6.459	0.339	0.212	0.054	12.522	2.611	0.043
	(1.21)	(0.11)	(0.04)	(0.02)	(3.16)	(0.53)	(0.02)
λ^{High}							
$\Delta\text{PnL}^{\text{EMQV}}$ (HD)	102.285	59.241	**38.113**	**19.704**	75.322	85.983	**15.829**
	(58.73)	(12.26)	(3.78)	(2.28)	(19.02)	(37.38)	(1.37)
$\Delta\text{PnL}^{\text{EMQV}}$ (CST)	**164.837**	2.111	27.062	0.476	**241.802**	92.267	7.842
	(53.06)	(0.31)	(2.18)	(0.05)	(64.09)	(24.48)	(0.59)
$\Delta\text{PnL}^{\text{SAC}}$ (HD)	−68.872	27.832	−79.031	−44.397	−517.230	3.143	−33.383
	(306.17)	(66.95)	(32.93)	(21.61)	(500.27)	(127.99)	(16.25)
$\Delta\text{PnL}^{\text{SAC}}$ (CST)	−396.266	**107.468**	−22.186	2.302	−215.508	**245.135**	15.083
	(278.97)	(62.99)	(31.61)	(20.14)	(390.73)	(130.87)	(15.86)
$\Delta\text{PnL}^{\text{AC-EC}}$	−905.742	25.328	20.470	6.731	−529.620	−203.392	4.885
	(121.44)	(5.54)	(1.80)	(0.76)	(160.46)	(59.75)	(0.70)

Observations & Discussion

- Empirically, EMQV demonstrates some performance advantages over SAC
- The AC model is relatively simplistic and does not consider potentially useful microstructural features
- SAC is developed without the AC model and can incorporate many features but can be problematic to train
- EMQV is an easy-to-train algorithm that delivers significant improvement over TWAP that is far more stable than that of SAC

Conclusions

- Our analytical solutions to the exploratory MQV problem under the AC model provide natural parameterizations of the value function and control policy for learning
- We introduce a recalibration step to the actor-critic algorithm which facilitates convergence
- A finite-time error analysis shows our algorithm converges linearly to the global optimum in the AC model given proper learning rate choices
- Simulation and empirical studies demonstrate the algorithm's effectiveness

References

- B. Wang, X. Gao, and L. Li. Reinforcement learning for continuous-time optimal execution: actor-critic algorithm and error analysis. *Available at SSRN: https://ssrn.com/abstract=4378950*, 2023.

- R. Almgren and N. Chriss. Optimal execution of portfolio transactions. *Journal of Risk*, 3(2):5–39, 2000.
- P. Forsyth, J. Kennedy, S. T. Tse, and H. Windcliff. Optimal trade execution: A mean quadratic variation approach. *Journal of Economic Dynamics and Control*, 36(12):1971–1991, 2012.
- T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290v2*, 2018.
- B. Hambly, R. Xu, and H. Yang. Policy gradient methods for the noisy linear quadratic regulator over a finite horizon. *SIAM Journal on Control and Optimization*, 59(5):3359–3391, 2021.
- Y. Jia and X. Y. Zhou. Policy evaluation and temporal-difference learning in continuous time and space: A martingale approach. *Journal of Machine Learning Research*, 23(154):1–55, 2022a.
- Y. Jia and X. Y. Zhou. Policy gradient and actor–critic learning in continuous time and space: Theory and algorithms. *Journal of Machine Learning Research*, 23(275):1–50, 2022b.
- A. S. Kyle and A. A. Obizhaeva. The market impact puzzle. *Available at SSRN: https://ssrn.com/abstract=3124502*, 2018.
- Y. Nevmyvaka, Y. Feng, and M. Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd International Conference on Machine Learning*, page 673–680, 2006.
- B. Ning, F. H. T. Lin, and S. Jaimungal. Double deep Q-learning for optimal execution. *Applied Mathematical Finance*, 28(4):361–380, 2021.
- B. Park and B. Van Roy. Adaptive execution: Exploration and learning of price impact. *Operations Research*, 63(5):1058–1076, 2015.
- H. Wang and X. Y. Zhou. Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Mathematical Finance*, 30(4):1273–1308, 2020.
- H. Wang, T. Zariphopoulou, and X. Y. Zhou. Reinforcement learning in continuous time and space: A stochastic control approach. *Journal of Machine Learning Research*, 21(198):1–34, 2020.