Optimal Execution

Dr. Tamás Nagy

Dr. Tamás Nagy

tamas@thelatent.space

Skeleton

Abstract

#8 Optimal Execution Under Nonlinear Market Impact.

This paper presents 38 machine-verified theorems building on 0 established facts and 62 hypotheses. All results are formally verified in the Platonic proof kernel (230 verification units, 58 proved statements) and exportable to Lean 4.

---

1. Introduction

2. Further Results

Theorem (perm_cost_positive). *Perm Cost Positive.* [Platonic: perm_cost_positive, domain: optimal_execution]

Theorem (perm_cost_quadratic). *Perm Cost Quadratic.* [Platonic: perm_cost_quadratic, domain: optimal_execution]

Theorem (temp_cost_positive). *Temp Cost Positive.* [Platonic: temp_cost_positive, domain: optimal_execution]

Theorem (patience_reduces_temp_cost). *Patience Reduces Temp Cost.* [Platonic: patience_reduces_temp_cost domain: optimal_execution]

Theorem (timing_risk_positive). *Timing Risk Positive.* [Platonic: timing_risk_positive, domain: optimal_execution]

Theorem (more_time_more_risk). *More Time More Risk.* [Platonic: more_time_more_risk, domain: optimal_execution]

Theorem (ra_cost_ge_expected). *Ra Cost Ge Expected.* [Platonic: ra_cost_ge_expected, domain: optimal_execution]

Theorem (urgency_positive). *Urgency Positive.* [Platonic: urgency_positive, domain: optimal_execution]

Theorem (risk_aversion_increases_urgency). *Risk Aversion Increases Urgency.* [Platonic: risk_aversion_increases_urgency, domain: optimal_execution]

Theorem (volatility_increases_urgency). *Volatility Increases Urgency.* [Platonic: volatility_increases_urgency, domain: optimal_execution]

Theorem (concave_impact_lower_per_unit). *Concave Impact Lower Per Unit.* [Platonic: concave_impact_lower_per_unit, domain: optimal_execution]

Theorem (front_load_reduces_concave_cost). *Front Load Reduces Concave Cost.* [Platonic: front_load_reduces_concave_cost, domain: optimal_execution]

Theorem (more_modes_tighter_trajectory). *More Modes Tighter Trajectory.* [Platonic: more_modes_tighter_trajectory, domain: optimal_execution]

Theorem (cost_error_from_trajectory). *Cost Error From Trajectory.* [Platonic: cost_error_from_trajectory, domain: optimal_execution]

Theorem (shortfall_improvement_ratio). *Shortfall Improvement Ratio.* [Platonic: shortfall_improvement_ratio, domain: optimal_execution]

Theorem (shortfall_quadratic_scaling). *Shortfall Quadratic Scaling.* [Platonic: shortfall_quadratic_scaling, domain: optimal_execution]

Theorem (cross_impact_increases_cost). *Cross Impact Increases Cost.* [Platonic: cross_impact_increases_cost, domain: optimal_execution]

Theorem (eigenmode_cost_decomposition). *Eigenmode Cost Decomposition.* [Platonic: eigenmode_cost_decomposition, domain: optimal_execution]

Theorem (diagonalization_speedup). *Diagonalization Speedup.* [Platonic: diagonalization_speedup, domain: optimal_execution]

Theorem (deep_adaptive_le_static). *Deep Adaptive Le Static.* [Platonic: deep_adaptive_le_static, domain: optimal_execution]

Theorem (deep_adaptation_benefit_nonneg). *Deep Adaptation Benefit Nonneg.* [Platonic: deep_adaptation_benefit_nonneg, domain: optimal_execution]

Theorem (deep_high_vol_more_benefit). *Deep High Vol More Benefit.* [Platonic: deep_high_vol_more_benefit, domain: optimal_execution]

Theorem (deep_unfilled_positive). *Deep Unfilled Positive.* [Platonic: deep_unfilled_positive, domain: optimal_execution]

Theorem (deep_better_fill_lower_cost). *Deep Better Fill Lower Cost.* [Platonic: deep_better_fill_lower_cost, domain: optimal_execution]

Theorem (deep_spread_increases_cost). *Deep Spread Increases Cost.* [Platonic: deep_spread_increases_cost, domain: optimal_execution]

Theorem (deep_wider_spread_higher_cost). *Deep Wider Spread Higher Cost.* [Platonic: deep_wider_spread_higher_cost, domain: optimal_execution]

Theorem (deep_info_leak_increases_impact). *Deep Info Leak Increases Impact.* [Platonic: deep_info_leak_increases_impact, domain: optimal_execution]

Theorem (deep_splitting_reduces_per_child). *Deep Splitting Reduces Per Child.* [Platonic: deep_splitting_reduces_per_child, domain: optimal_execution]

Theorem (deep_more_slices_less_leak). *Deep More Slices Less Leak.* [Platonic: deep_more_slices_less_leak, domain: optimal_execution]

Theorem (deep_bellman_optimality). *Deep Bellman Optimality.* [Platonic: deep_bellman_optimality, domain: optimal_execution]

3. Bounds and Estimates

Theorem (trajectory_error_bounded). *Trajectory Error Bounded.* [Platonic: trajectory_error_bounded, domain: optimal_execution]

Theorem (cross_impact_bounded). *Cross Impact Bounded.* [Platonic: cross_impact_bounded, domain: optimal_execution]

Theorem (deep_dark_fraction_bounded). *Deep Dark Fraction Bounded.* [Platonic: deep_dark_fraction_bounded, domain: optimal_execution]

Theorem (deep_policy_error_bounded). *Deep Policy Error Bounded.* [Platonic: deep_policy_error_bounded, domain: optimal_execution]

4. Convergence Results

Theorem (linear_limit_no_frontload). *Linear Limit No Frontload.* [Platonic: linear_limit_no_frontload, domain: optimal_execution]

Theorem (spectral_cost_convergence). *Spectral Cost Convergence.* [Platonic: spectral_cost_convergence, domain: optimal_execution]

Theorem (deep_value_function_convergence). *Deep Value Function Convergence.* [Platonic: deep_value_function_convergence, domain: optimal_execution]

5. Spectral Theory

Theorem (spectral_lower_shortfall). *Spectral Lower Shortfall.* [Platonic: spectral_lower_shortfall, domain: optimal_execution]

6. Formal Framework

Hypotheses

- H_X_pos: X Pos
- H_T_pos: T Pos
- H_gamma_pos: Gamma Pos
- H_eta_pos: Eta Pos
- H_sigma_pos: Sigma Pos
- H_lam_pos: Lam Pos
- H_delta_pos: Delta Pos
- H_delta_le1: Delta Le1
- H_rho_gt1: Rho Gt1
- H_N_pos: N Pos
- H_C_pos: C Pos
- H_X2_pos: X2 Pos

- H_X2_gt_X: X2 Gt X
- H_T2_pos: T2 Pos
- H_T2_gt_T: T2 Gt T
- H_lam2_pos: Lam2 Pos
- H_lam2_gt: Lam2 Gt
- H_sigma2_pos: Sigma2 Pos
- H_sigma2_gt: Sigma2 Gt
- H_rate_gt1: Rate Gt1
- H_vnow_pos: Vnow Pos
- H_vlater_pos: Vlater Pos
- H_front_load: Front Load
- H_traj_bound: Traj Bound
- H_traj_nn: Traj Nn
- H_spec_bound_pos: Spec Bound Pos
- H_log_rho_pos: Log Rho Pos
- H_IS_pos: Is Pos
- H_ISs_pos: Iss Pos
- H_ISn_pos: Isn Pos
- H_spectral_better: Spectral Better
- H_na_pos: Na Pos
- H_ci_pos: Ci Pos
- H_ci_lt_self: Ci Lt Self
- H_cm1_nn: Cm1 Nn
- H_cm2_nn: Cm2 Nn
- H_cst_pos: Cst Pos
- H_cad_pos: Cad Pos
- H_adaptive_better: Adaptive Better
- H_blv_nn: Blv Nn
- H_bhv_nn: Bhv Nn
- H_vol_helps: Vol Helps
- H_fp_pos: Fp Pos
- H_fp_lt1: Fp Lt1
- H_fp2_pos: Fp2 Pos
- H_fp2_gt: Fp2 Gt
- H_fp2_lt1: Fp2 Lt1
- H_df_pos: Df Pos
- H_df_lt1: Df Lt1
- H_spread_pos: Spread Pos
- H_s2_pos: S2 Pos
- H_s2_gt: S2 Gt
- H_il_pos: Il Pos
- H_nc_gt1: Nc Gt1
- H_Vo_pos: Vo Pos
- H_Vs_pos: Vs Pos
- H_opt_le: Opt Le
- H_VN_pos: Vn Pos
- H_ve_nn: Ve Nn
- H_ve_bound: Ve Bound

- H_pe_nn: Pe Nn
- H_pe_bound: Pe Bound

7. Proof Architecture

All proofs are implemented in the Platonic kernel (elysium/fields/optimal_execution/).

File	Role
nonlinear_impact_proof.py	

8. Discussion

References