Algebraic Reduction and Numerical Optimization Methods for Solving Quantum Control Sum of Squares Problems

Alexander Leong

JuliaCon 2026

Mainz, Germany

The Problem

Quantum control: find $E(t) = \sum_k x_k\, t^k$ that drives a system to a target unitary U^*

$$\min \ \ \mathrm{Tr}\left(M^\dagger M\right), \quad M = U(T) - U^* \approx e^{\Lambda_n/2} - e^{-\Lambda_n/2} U^*$$

Unitary, $U(T)$ approximated via **Magnus expansion** (ODE solution) + **Chebyshev expansion** (matrix exponential):

$$e^{\alpha \Lambda_n} \approx J_0(\alpha)\mathbf{1} + 2\sum_{k=1}^{p} J_k\,(\alpha) T_k$$

Result: the objective becomes a polynomial in x_k — solvable globally via Sum of Squares programming.

The Problem

Quantum control: find $E(t) = \sum_k x_k \, t^k$ that drives a system to a target unitary U^* given control Hamiltonian, V and drift Hamiltonian, H_0

$$\begin{aligned} \text{minimize} \quad & \mathrm{Tr}\left(M^\dagger M\right) \\ \text{subject to} \quad & \partial_t U(t) = A(t)U(t) \\ & U(0) = \mathbb{1} \\ & A(t) = -i(H_0 + E(t)V) \end{aligned}$$

$$M \approx e^{\Lambda_n/2} - e^{-\Lambda_n/2} U^*$$

Result: the objective becomes a polynomial in x_k — solvable globally via Sum of Squares programming.

The Problem

- This can be written as a sum of squares program solved using semidefinite programming!

$$\begin{aligned} \text{minimize} \quad & [x]_d^T Q [x]_d \\ \text{subject to} \quad & Q \in S_+^d \end{aligned}$$

- We solve for the polynomial control coefficients, Q and choose an appropriate basis, $[x]_d$
- Even for small quantum systems (e.g. 3x3), SDPs become intractable quickly!

Symmetry Reduction:
`SymbolicWedderburn.jl`

Block-diagonalize the SDP via Wedderburn decomposition over the Symmetric Group:

$$V = \bigoplus_{i=1}^{r} V_i \quad \Rightarrow \quad \text{4 independent, smaller SDPs}$$

Affine constraints become per-block:

$$A^i = \sum_j V_i^j, \qquad b^i = \alpha \cdot V_i^j$$

Reconstruct the full solution as: $\sum_i V_i \, X V_i^T$

27× variable reduction before the solver even starts — and better numerical conditioning as a bonus.

Face Reduction

ConicSolveFR.jl

Symmetry reduction alone leaves **structural degeneracy** — the solution lives on a lower-dimensional face of the cone. Interior point solvers stall without this.

Algorithm (Permenter 2017): iteratively expose the minimal face via TSVD:

```
while SDP(*) feasible:
    solve SDP(*) → get X̃
    update face: X̃ = U_r Λ_r U_r^T   (truncated SVD)
return minimal face F
```

Solve the reduced problem, then invert: $X_{n-1} = UX_kU^T$

⚠️ TSVD threshold η_λ is the key tuning parameter — too small: stalls; too large: diverges from original problem.

The QCSOS Pipeline

ConicSolve.jl — symmetric self-dual interior point solver: - Euclidean-Jordan algebra structure; no autodiff (AD) needed - QR + Cholesky for KKT system, equilibration + regularization - Native support for symmetry & face reduction inputs

```
""" * `ϵ`: absolute tolerance
* `η_eps`: absolute tolerance to determine exposed face (using duality gap)
* `η_lambda`: absolute tolerance to remove near redundant constraints
* `p`: order of Chebyshev expansion for approximating matrix exponential """
function run_test(ϵ = 1e-2, η_eps = 1e-3, η_lambda = 1e-3, p = 2)
    # define optimization problem
    U_target, problem = get_qc_problem()
    problem = QuantumUnitaryFixedTimeProblem(U_target, problem, p)
    qcsos_solver = QCSOS_Solver(p, problem, ϵ, η_eps, η_lambda)
    solve!(qcsos_solver)

    # get solution
    solution = get_solution(problem.program)

    # evaluate solution
    coefficients = get_control(problem, solution)
    H_result = get_unitary_from_control(problem, coefficients)
    infidelity = get_infidelity(U_target, H_result)
    return U_target, H_result, problem, solution
end
```

```
function QuantumUnitaryFixedTimeProblem(U_target, problem, order=2)
    exp½Ω = est_unitary(problem, order)
    A = exp½Ω' *  U_target - exp½Ω
    f = get_hilbert_schmidt_inner_product(A)
    n = problem.n

    cone_qp = ConeQP()
    program = define_program(cone_qp,
          minimize(f),
          f ∈ ConicSolve.SymmetricGroup(n))

    build_program(program)
    problem.program = program

    return problem
end
```

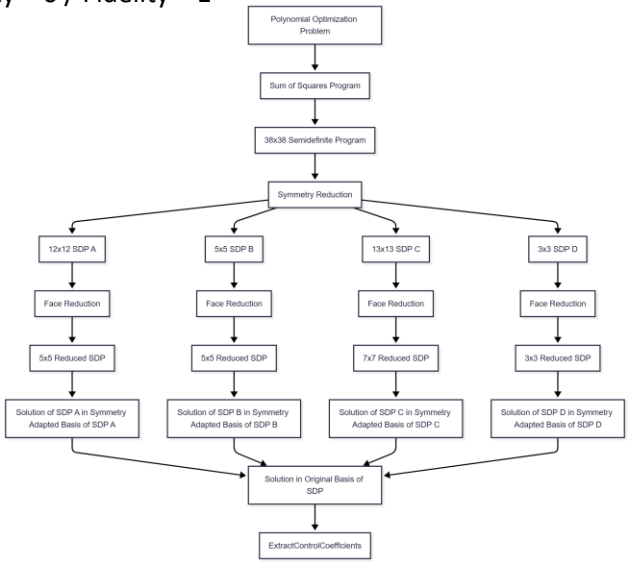
QCSOS Pipeline In Action

What do we want?

(Dual, Primal, Cent.) Res. ≈ 0

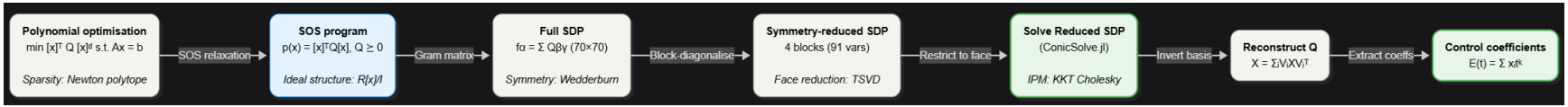
Duality gap ≈ 0

Infidelity ≈ 0 / Fidelity ≈ 1




```
Info: Performing face reduction on cone 5 of 5
Info: Subproblem constructed of size 91
Info: Iter. |      Dual Res. |    Primal Res. |     Cent. Res. |      Dual. Gap |    Primal Obj. |      Dual Obj. |      Step size |             b'y |
Info:     10 |        1.22222 |     2.7607e-16 |        166.072 |        0.114093 |            0.0 |    -1.14808e-9 |      0.0296663 |     1.09035e-9 |
Info: Constraint already feasible, nothing to reduce
Info: Nothing reduced
Info: Subproblem feasible
Info: Subproblem reduced
Info: Optimize called
Info:               Solver parameters
Info:              Objective limit:  -Inf
Info:                Limit solution:  0.0
Info:                 Max iterations:  10
Info:              Num constraints:  140
Info:                    Num threads:  1
Info:             Preferred device:  CPU
Info: Iterative Refinement Max Iterations:  1
Info: Iterative Refinement Trigger Mode:  NO_ITERATIVE_REFINEMENT
Info:  Presolve Regularization Method: none
Info: Presolve Scaling (Equilibration) Method: ruiz
Info:       System Solver KKT method:  QR and Cholesky
Info:              Time limit (seconds):  1000000
Info:          Tolerance gap absolute:  0.0001
Info:          Tolerance gap relative:  0.0001
Info:           Tolerance optimality:  0.01
Info: Performing Ruiz Equilibration
Info: Solving for primal dual starting points
Info: Executing main optimization loop
Info: Iter. |      Dual Res. |    Primal Res. |     Cent. Res. |      Dual. Gap |    Primal Obj. |      Dual Obj. |      Step size |
Info:      1 |        3.60555 |4.082479999999999e-31 |        7.69185e-16 |                |            1.0 |            0.0 |     3.29242e-18 |     0.000328976 |
Info:      2 |        1.01933 |     7.89854e-16 |        1.36866 |        0.282705 |            0.0 |    -1.30152e-9 |       0.732622 |
Info: Slow progress...
Info:      3 |        1.34468 |     7.65259e-16 |        271.485 |         0.28123 |            0.0 |    -1.38242e-9 |     0.00821025 |
Info:      4 |        1.44543 |     6.23237e-16 |        25.8466 |        0.276814 |            0.0 |     -1.7389e-9 |      0.182095 |
Info: Slow progress...
Info:      5 |        1.52599 |     6.33852e-16 |       1.2927e6 |        0.276575 |            0.0 |    -4.5288e-11 |      4.96022e-5 |
Info:      6 |        1.63113 |     6.18324e-16 |        338.699 |        0.272053 |            0.0 |   -2.17993e-10 |      0.0102004 |
Info:      7 |        1.67295 |      4.0018e-16 |         1.3818 |        0.188906 |            0.0 |   -8.44292e-10 |       0.484499 |
Info:      8 |        1.09964 |     2.59768e-16 |       0.723022 |        0.116047 |            0.0 |    -1.10313e-9 |       0.467991 |
Info: Slow progress...
Info:      9 |        1.11866 |     2.71525e-16 |        3694.3 |        0.116005 |            0.0 |     -1.0981e-9 |     0.00375714 |
Info:     10 |        1.14028 |     2.88731e-16 |        163.137 |        0.111972 |            0.0 |    -1.19215e-9 |      0.0321183 |
Info: Solver finished
Exit status: ITERATION_LIMIT
Primal objective value: 0.0
Number of iterations: 10
Time elapsed: 0.028846816000000004
Info: Iter. |      Dual Res. |    Primal Res. |     Cent. Res. |      Dual. Gap |    Primal Obj. |      Dual Obj. |      Step size |             b'y |
Info:     10 |        1.14028 |     2.88731e-16 |        163.137 |        0.111972 |            0.0 |    -1.19215e-9 |      0.0321183 |     1.05874e-9 |
Info: Solved reduced problem
Info: Reproject iterate
Info: Reproject iterate
Info: Reproject iterate
Info: Reproject iterate
Info: Reproject iterate
Info: Face reduction completed
Info: WARNING: no result from subprogram, skipping
Info: Numerical rank of solution (ideally 1) with tolerance 0.0001 is 93
Info: Extracted control coefficients
Info: Infidelity: 0.10637362880070156
```

The QCSOS method computational pipeline

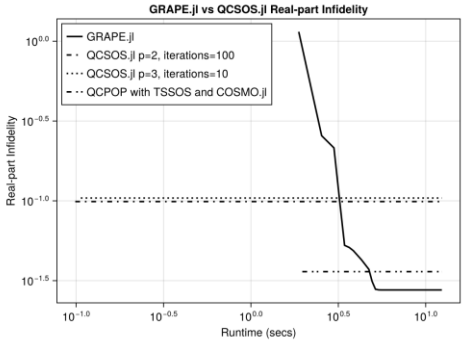


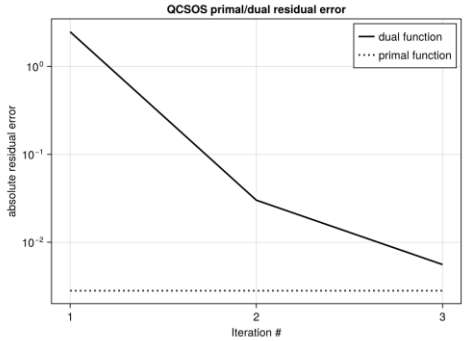
Achieve global convergence via
Structural exploitation -> better numerical stability and efficiency -> faster convergence and tighter numerical guarantees

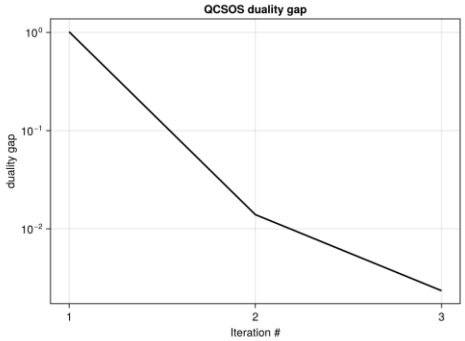
Results: QCSOS vs GRAPE.jl and QCPOP

For a Three-level system

Performance Measure	GRAPE.jl	QCPOP with TSSOS + COSMO.jl	QCSOS.jl
Infidelity	0.0277	0.036	**0.099**
Runtime (seconds)	53.2	1.97	**0.1**







🚀 **QCSOS is ~500× faster at ~1e-1 infidelity compared to GRAPE.jl**

GRAPE plateaus at iteration ~10. QCSOS reaches its solution **in single-digit iterations** — no local minima to escape.

Discussion & Future Work

What's working: - Layered algebraic reduction without relaxing the original problem - Composable Julia packages — drop-in preprocessing for existing solvers

Current limitations: - Numerically brittle — careful parameter tuning required - Residuals plateau at physics model ceiling (Magnus/Chebyshev truncation)

Next steps: - Alternate polynomial bases (e.g. Chebyshev) for better numerical stability - Partial face reduction to reduce overhead of multiple face-exposing SDPs - Scaling to larger quantum systems for safety-critical applications

Thank You!

Step	Julia Package	Contribution
Symmetry reduction	`SymbolicWedderburn.jl`	2485 → 91 variables
SDP solve + face reduction	`ConicSolve.jl`	Degeneracy-robust IPM
Full pipeline	`QCSOS.jl`	215× speedup, 7× infidelity

📦 github.com/alexander-leong/QCSOS.jl

📧 toshibaalexander@gmail.com

Thanks to QNumerics 2025 for inspiring this work.

Link to Paper =>