Exploiting Symmetry in Large-Scale Optimization and Control

Claus Danielson
Assistant Professor
Department of Mechanical Engineering
University of New Mexico

Outline

- Personal Overview
  - Background
  - Research Interests
- Symmetry
  - Model Predictive Control (MPC)
  - Symmetric MPC
  - Symmetric Explicit MPC
  - Symmetric Implicit MPC
    - Symmetric Alternating Direction Method of Multipliers (ADMM)
  - Example – Symmetric HVAC

Education

Doctorate: University of California, Berkeley

- Advisor: Francesco Borrelli
- Model Predictive Control Lab

Masters: Rensselaer Polytechnic Institute

- Advisor: John Wen
- Center for Automation Technology and Systems

Bachelors: University of Washington

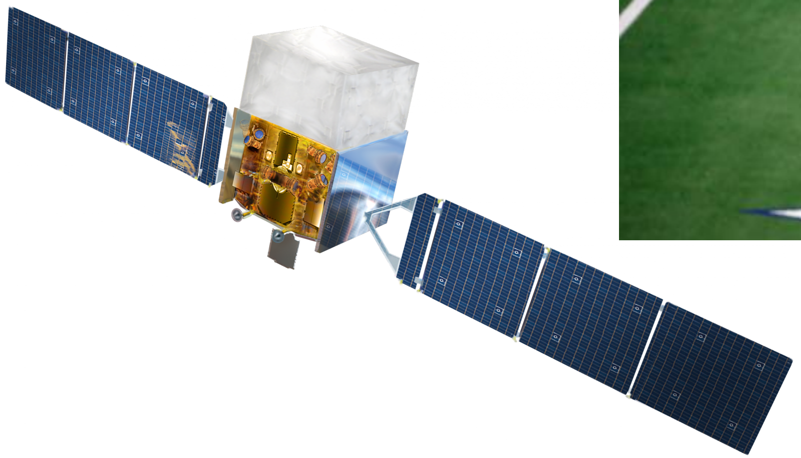
Industrial Experience

Mitsubishi Electric Research Laboratories

- Autonomous driving
- Heating, Ventilation, and Air-Conditioning
- Advanced Manufacturing, Spacecraft (JAXA)

Sequoia Technologies

- Local start-up
- Robotics for television broadcast

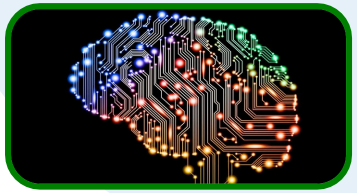


Air Force Research Laboratories

General Dynamics

Research Interests

Precision motion control
Factory automation



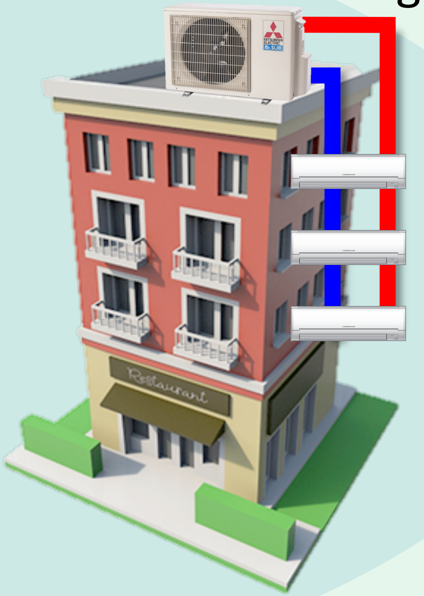
Learning-based control

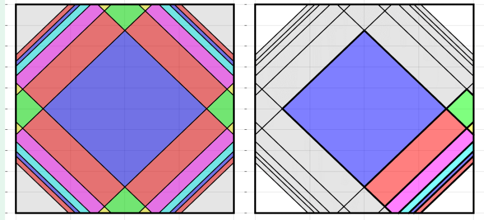
Motion planning

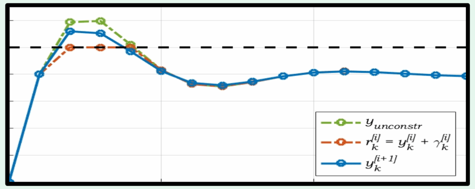
- Autonomous driving
- Drones and robotics
- Spacecraft rendezvous

Control

Optimization and Control for Energy Efficiency







Real-time optimization
Computational geometry
Symmetry

Optimization

Outline

- Personal Overview
  - Background
  - Research Interests
- Symmetry
  - Model Predictive Control (MPC)
  - Symmetric MPC
  - Symmetric Explicit MPC
  - Symmetric Implicit MPC
    - Symmetric Alternating Direction Method of Multipliers (ADMM)
  - Example – Symmetric HVAC

Symmetry: Intuition and Applications

Motivation: Model predictive control for large-scale systems

- High-dimensional problems
- Limited computation

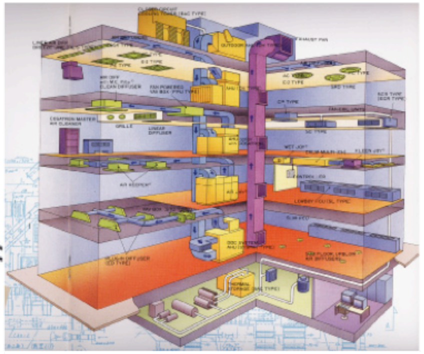
Large-scale systems comprised of repeated components connected in regular patterns

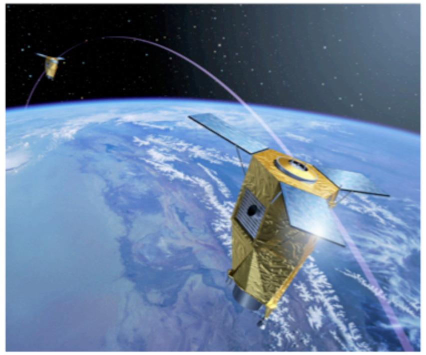
- Patterns called symmetries
- Also called invariance or equivariance

Benefit:

- Symmetric systems are simpler
- More symmetries ➔ simpler system

Question: How do we use symmetry to simplify MPC?

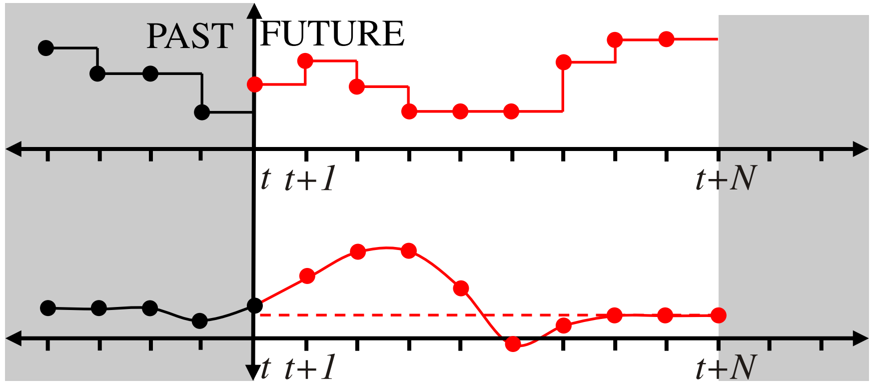




Model Predictive Control

Receding Horizon Control

1. Determine current situation
   - Look at the chess board
   - Measuring sensors
2. Formulate a plan of future actions
   - Think about strategy
   - Solve optimal control problem
3. Apply first-step of plan
   - Move your pieces
   - Command the actuators



Finite-Time Optimal Control Problem

$$\min_{u_0,\dots,u_{N-1}} \quad p(x_N) + \sum_{k=1}^{N} q(x_k, u_k)$$
$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$
$$x_k \in \mathcal{X}, u_k \in \mathcal{U}$$

Model Predictive Control:

- Obtain control input by solving constrained finite-time optimal control (CFTOC) problem

Model Predictive Control

Receding Horizon Control

1. Determine current situation
   - Look at the chess board
   - Measuring sensors
2. Formulate a plan of future actions
   - Think about strategy
   - Solve optimal control problem
3. Apply first-step of plan
   - Move your pieces
   - Command the actuators

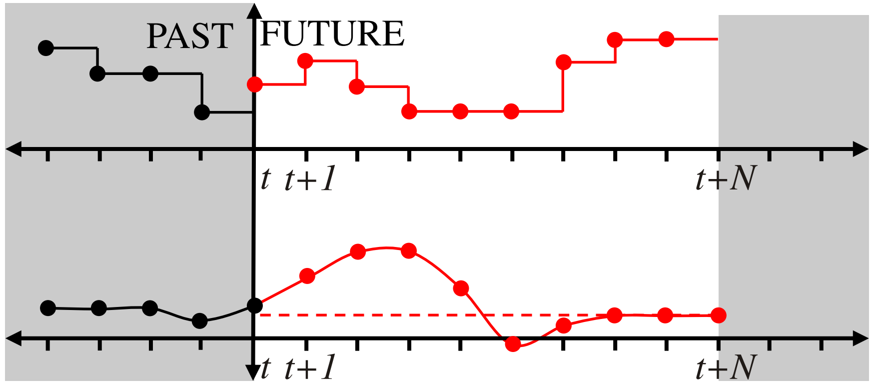
Model Predictive Control:

- Obtain control input by solving constrained finite-time optimal control (CFTOC) problem

Model Predictive Control

Design Considerations for Model Predictive Control:

- Stability: Closed-loop stability not guaranteed
- Persistent Feasibility: Optimization problem
- may become infeasible
- Robustness: Controller must continue to function in presence of model uncertainty
- Performance: Optimization problem only considers cost over finite-horizon
- Real-time Implementation: Need to solve optimization problem in real-time



Finite-Time Optimal Control Problem

$$\min_{u_0,\ldots,u_{N-1}} \quad p(x_N) + \sum_{k=1}^{N} q(x_k, u_k)$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k)$$

$$x_k \in \mathcal{X}, u_k \in \mathcal{U}$$

Model Predictive Control

Model Predictive Control:

- Obtain control input by solving constrained finite-time optimal control (CFTOC) problem

Explicit Model Predictive Control:

- Replace CFTOC with pre-solved look-up table
- If CFTOC is QP or LP then look-up table is piecewise affine on polyhedral partition

Finite-Time Optimal Control Problem

$$\begin{aligned} \min_{u_0,\ldots,u_{N-1}} \quad & p(x_N) + \sum_{k=1}^{N} q(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k) \\ & x_k \in \mathcal{X}, u_k \in \mathcal{U} \end{aligned}$$

Explicit Model Predictive Control

$$u_0^\star(x) = \begin{cases} F_1 x + G_1 & \text{for } x \in \mathcal{R}_1 \\ \vdots & \\ F_r x + G_r & \text{for } x \in \mathcal{R}_r. \end{cases}$$

Outline

- Personal Overview
    - Background
    - Research Interests
- Symmetry
    - Model Predictive Control (MPC)
    - Symmetric MPC
    - Symmetric Explicit MPC
    - Symmetric Implicit MPC
        - Symmetric Alternating Direction Method of Multipliers (ADMM)
- Example – Symmetric HVAC

Symmetric Constrained Optimal Control

A symmetry of the optimal control problem is a transformation that preserves:

- Constraints

$$\Theta \mathcal{X} = \mathcal{X}$$

$$\Omega \mathcal{U} = \mathcal{U}$$

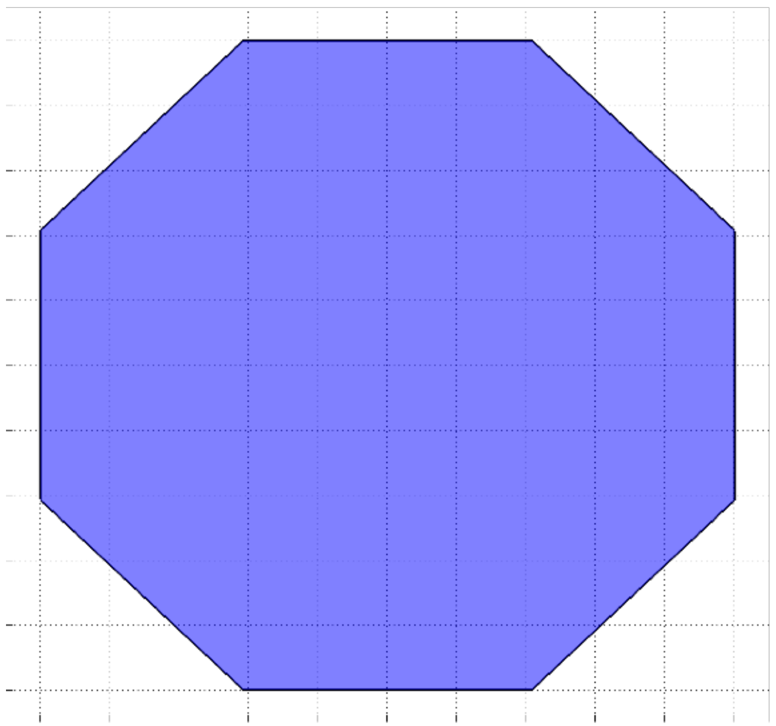
- Dynamics

$$\Theta f(x, u) = f(\Theta x, \Omega u)$$

- Cost

$$p(\Theta x) = p(x)$$

$$q(\Theta x, \Omega u) = q(x, u)$$



Symmetric Constrained Optimal Control

A symmetry of the optimal control problem is a transformation that preserves:

- Constraints

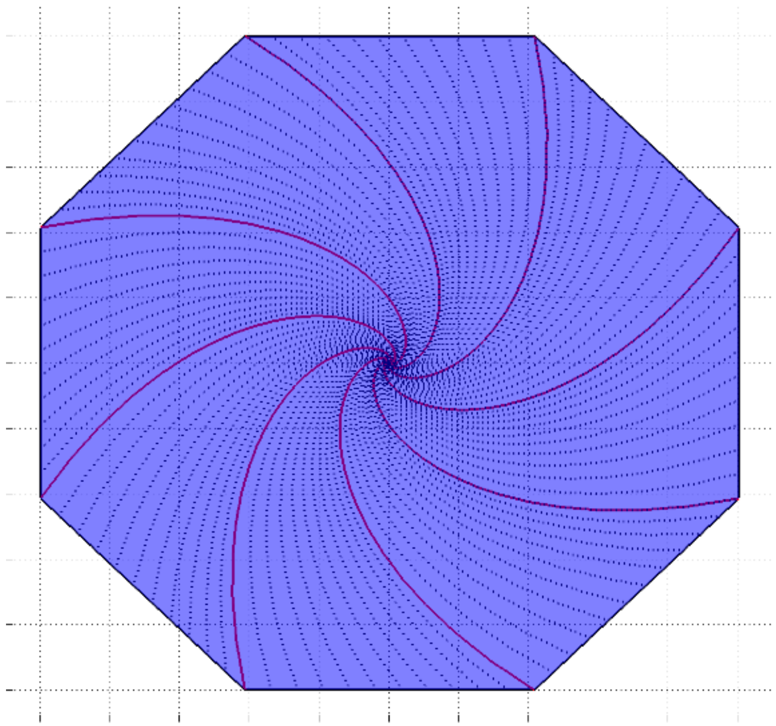
$$\Theta \mathcal{X} = \mathcal{X}$$
$$\Omega \mathcal{U} = \mathcal{U}$$

- Dynamics

$$\Theta f(x,u) = f(\Theta x, \Omega u)$$

- Cost

$$p(\Theta x) = p(x)$$
$$q(\Theta x, \Omega u) = q(x,u)$$



Symmetric Constrained Optimal Control

Theorem: Symmetric MPC

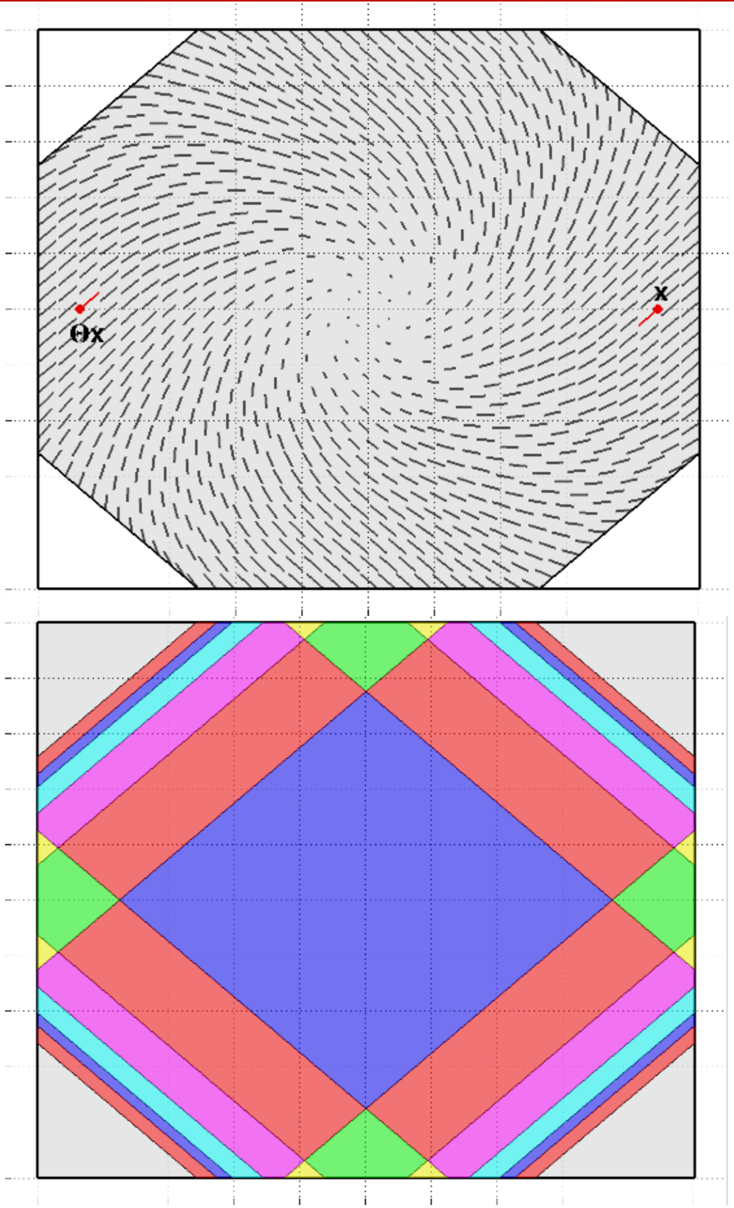
If the optimal control problem is symmetric and convex it has a symmetric controller

$$\Omega u_0^\star(x) = u_0^\star(\Theta x)$$

Corollary: Symmetric Explicit MPC

If the symmetric MPC is an LP or QP then the symmetries permute the controller pieces

$$\Omega F_i = F_j \Theta$$
$$\Omega G_i = G_j$$
$$\Theta \mathcal{R}_i = \mathcal{R}_j$$



Symmetric Constrained Optimal Control

Theorem: Symmetric MPC

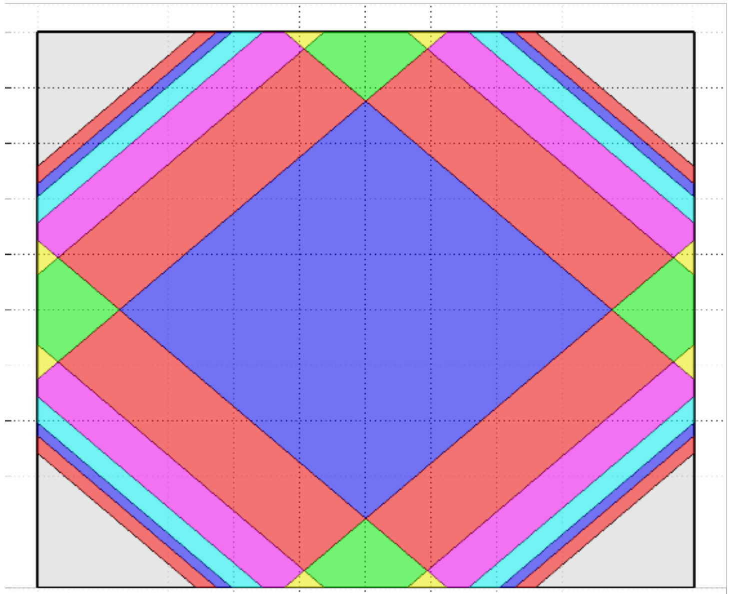
If the optimal control problem is symmetric and convex it has a symmetric controller

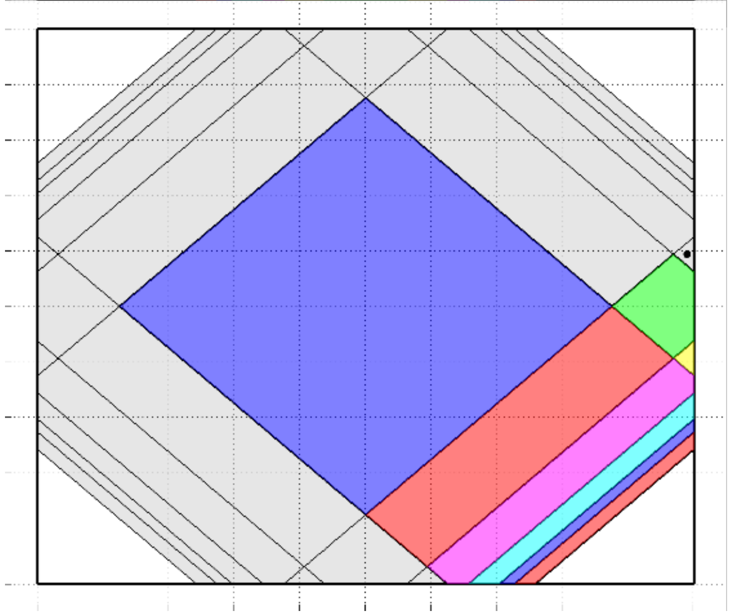
$$\Omega u_0^\star(x) = u_0^\star(\Theta x)$$

Corollary: Symmetric Explicit MPC

If the symmetric MPC is an LP or QP then the symmetries permute the controller pieces

$$\Omega F_i = F_j \Theta$$
$$\Omega G_i = G_j$$
$$\Theta \mathcal{R}_i = \mathcal{R}_j$$





Example: Quadrotor

Symmetry group: Dihedral-4 group $\times$ reflection group

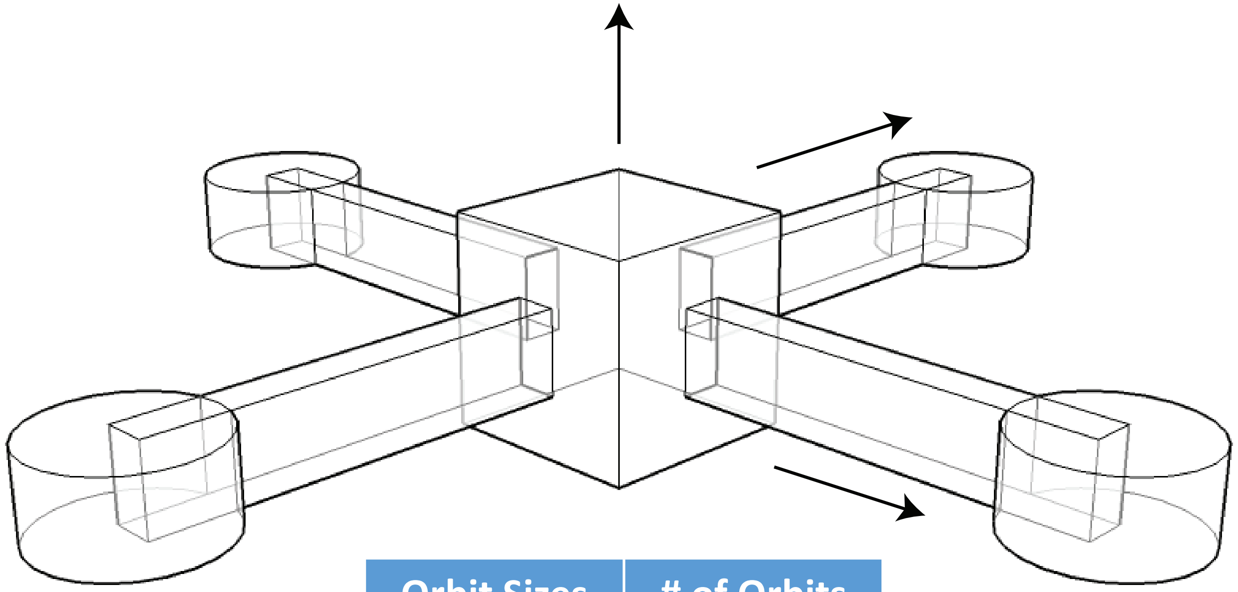
- 16 symmetries

Without symmetry:

- 10,173 controller pieces
- 53.7 megabytes

With symmetry:

- 772 controller pieces
- 4.2 megabytes



Orbit Sizes	# of Orbits
1	1
2	6
4	30
8	215
16	520

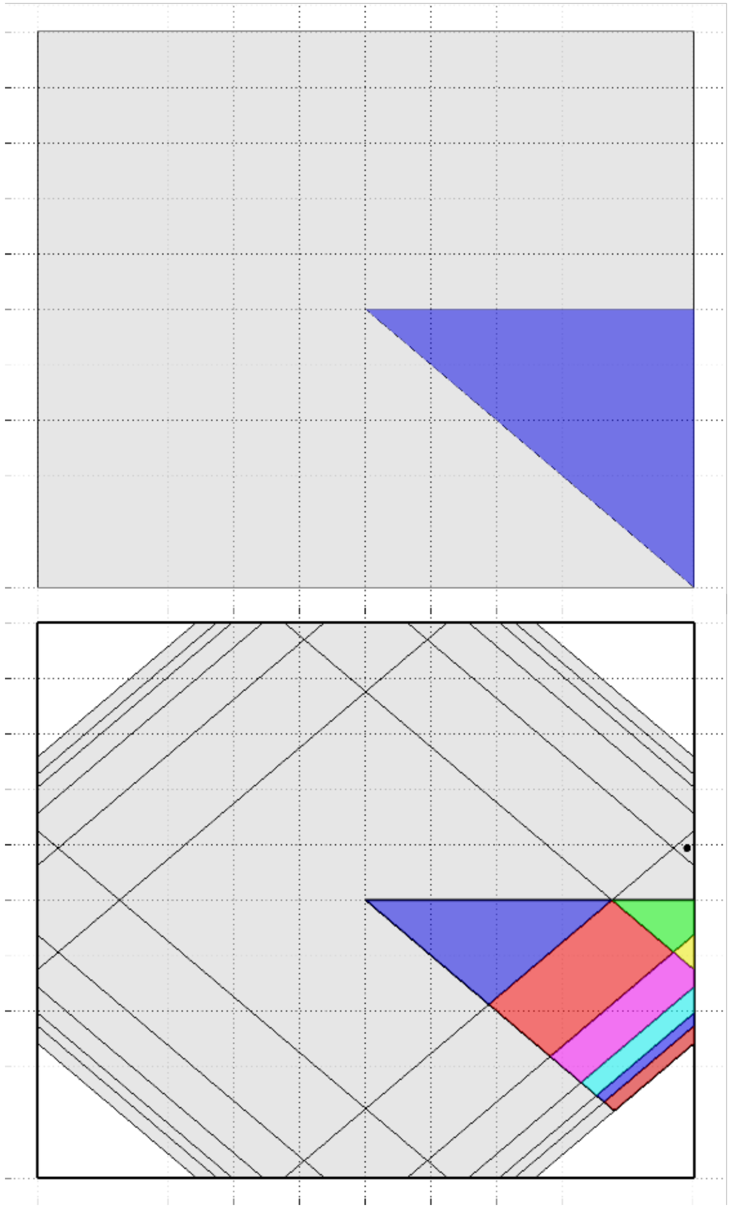
Fundamental Domain Controller

Fundamental Domain Controller

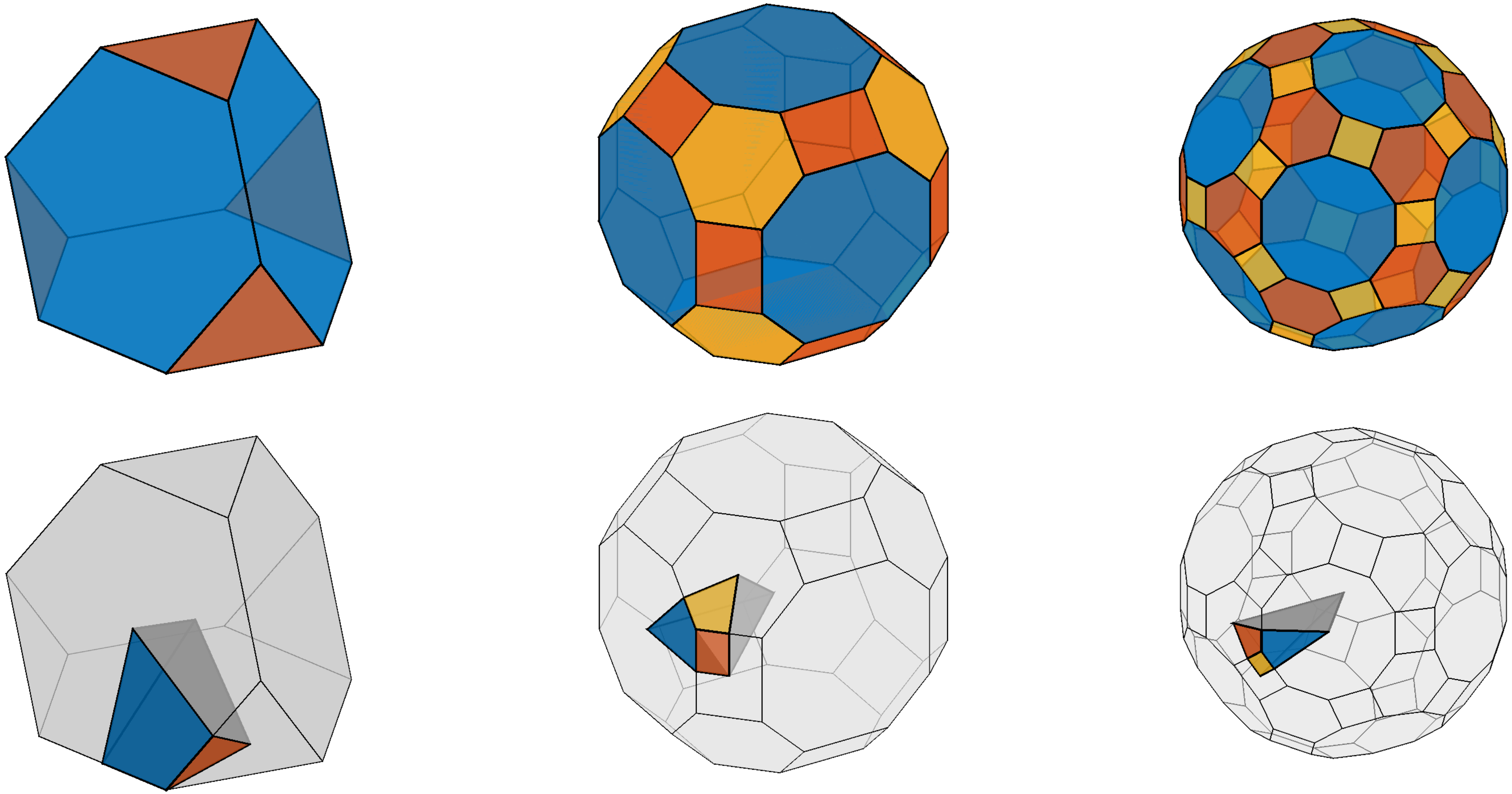
- Orbit controller requires solving original constrained optimal control problem and compressing the result.
- Fundamental domain controller solves a smaller constrained optimal control problem.

Linear-complexity algorithms

- Constructing fundamental domain
- Searching for transformation into fundamental domain



Fundamental Domains: Archimedean Solids



Outline

- Personal Overview
  - Background
  - Research Interests
- Symmetry
  - Model Predictive Control (MPC)
  - Symmetric MPC
  - Symmetric Explicit MPC
  - Symmetric Implicit MPC
    - Symmetric Alternating Direction Method of Multipliers (ADMM)
  - Example – Symmetric HVAC

Alternating Direction Method of Multipliers Algorithm

Problem:

- Generic linear MPC problem
- Outputs are quantities to be constrained
- Output constraints include input constraints
- Box constraints on outputs

$$
\begin{aligned}
\min \sum_{k=0}^{N-1} & \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} Q & S \\ S & R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\
\text{s.t.} \quad x_{k+1} &= Ax_k + Bu_k \\
y_k &= Cx_k + Du_k \\
y_k &\in \mathcal{Y}
\end{aligned}
$$

- Terminal cost and constraints can be added

Alternating Direction Method of Multipliers Algorithm

ADMM: Augmented Lagrangian QP

- Split variable with inequality constraints

$$
\begin{aligned}
\min \sum_{k=0}^{N-1} & \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} Q & S \\ S & R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\
\text{s.t. } \; x_{k+1} &= Ax_k + Bu_k \\
y_k &= Cx_k + Du_k \\
y_k &\in \mathcal{Y}
\end{aligned}
$$

$$\Downarrow$$

$$
\begin{aligned}
\min \sum_{k=0}^{N-1} & \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} Q & S \\ S & R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\
\text{s.t. } \; x_{k+1} &= Ax_k + Bu_k \\
v_k &= Cx_k + Du_k \\
v_k &= y_k \\
y_k &\in \mathcal{Y}
\end{aligned}
$$

Alternating Direction Method of Multipliers Algorithm

- Split variable with inequality constraints
- Add equality constraint to cost function
  - Lagrange multiplier
  - Quadratic regularization

$$
\begin{aligned}
\min \; & \sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} Q & S \\ S & R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\
\text{s.t.} \; & x_{k+1} = Ax_k + Bu_k \\
& v_k = Cx_k + Du_k \\
& v_k = y_k \\
& y_k \in \mathcal{Y}
\end{aligned}
$$

$$
\begin{aligned}
\min \; & \sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} Q & S \\ S & R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \frac{\rho}{2} \| v_k - y_k - \gamma_k \|_2^2 \\
\text{s.t.} \; & x_{k+1} = Ax_k + Bu_k \\
& v_k = Cx_k + Du_k \\
& y_k \in \mathcal{Y}
\end{aligned}
$$

Alternating Direction Method of Multipliers Algorithm

Iteratively solve Augmented Lagrangian QP

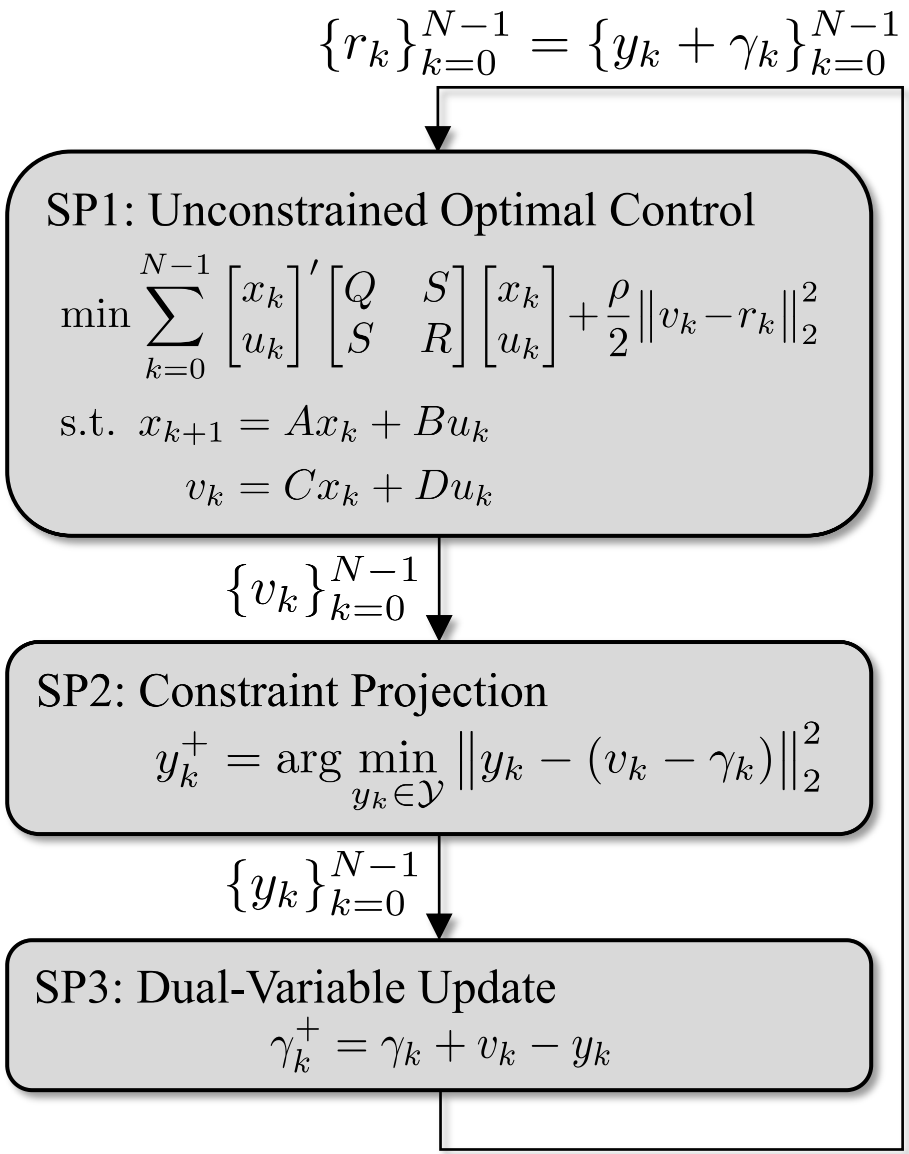
1. Solve for states, inputs, and unconstrained outputs
   x_k, u_k, v_k
2. Solve for constrained outputs y_k
3. Solve for dual-variables γ_k

Intuition:

- Sub-problem 1 trades-off tracking the unconstrained optimal and reference that avoids constraints

$$\{r_k\}_{k=0}^{N-1} = \{y_k + \gamma_k\}_{k=0}^{N-1}$$

- Sub-problems 2+3 provide integral-action on the reference to avoid constrain violations



Alternating Direction Method of Multipliers Algorithm

Iteratively solve Augmented Lagrangian QP

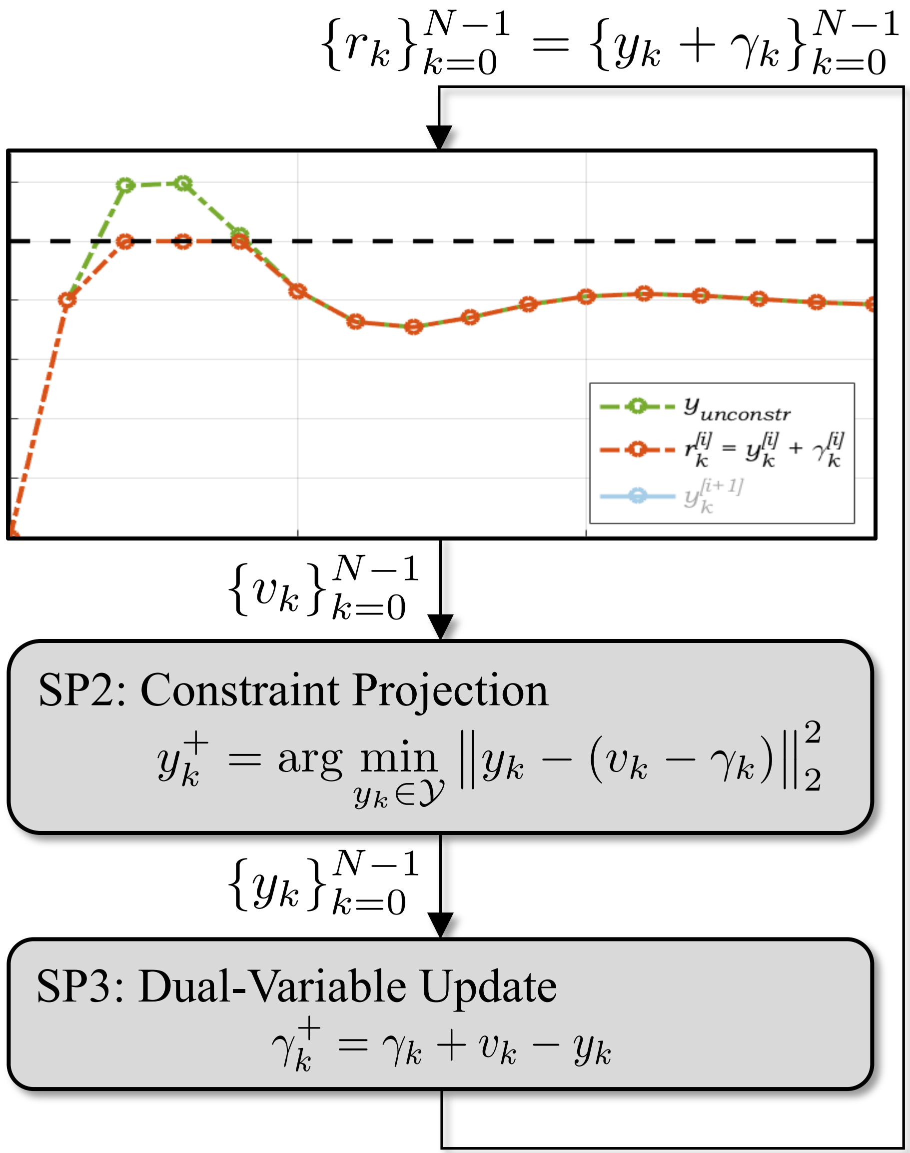
1. Solve for states, inputs, and unconstrained outputs
   x_k, u_k, v_k
2. Solve for constrained outputs y_k
3. Solve for dual-variables γ_k

Intuition:

- Sub-problem 1 trades-off tracking the unconstrained optimal and reference that avoids constraints

$$\{r_k\}_{k=0}^{N-1} = \{y_k + \gamma_k\}_{k=0}^{N-1}$$

- Sub-problems 2+3 provide integral-action on the reference to avoid constrain violations



Alternating Direction Method of Multipliers Algorithm

Iteratively solve Augmented Lagrangian QP

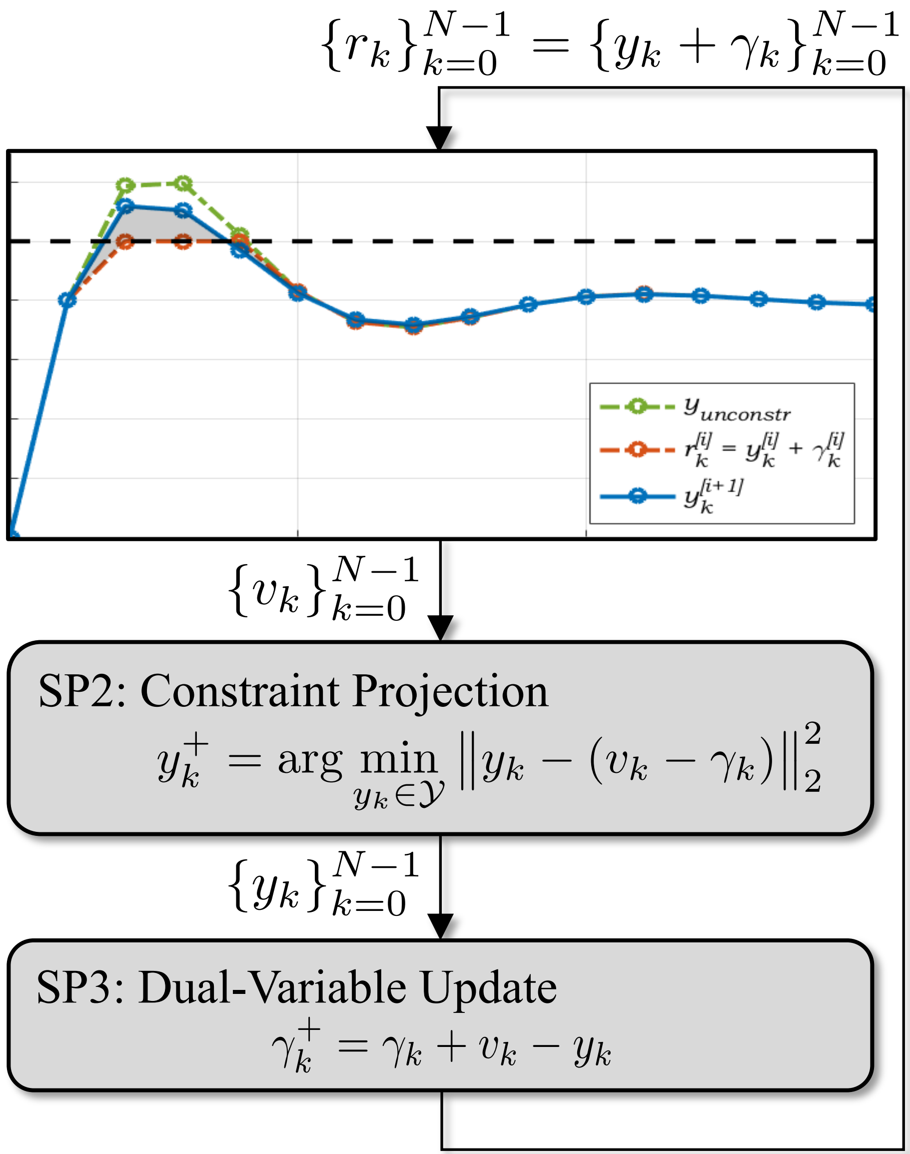
1. Solve for states, inputs, and unconstrained outputs
   x_k, u_k, v_k
2. Solve for constrained outputs y_k
3. Solve for dual-variables γ_k

Intuition:

- Sub-problem 1 trades-off tracking the unconstrained optimal and reference that avoids constraints

$$\{r_k\}_{k=0}^{N-1} = \{y_k + \gamma_k\}_{k=0}^{N-1}$$

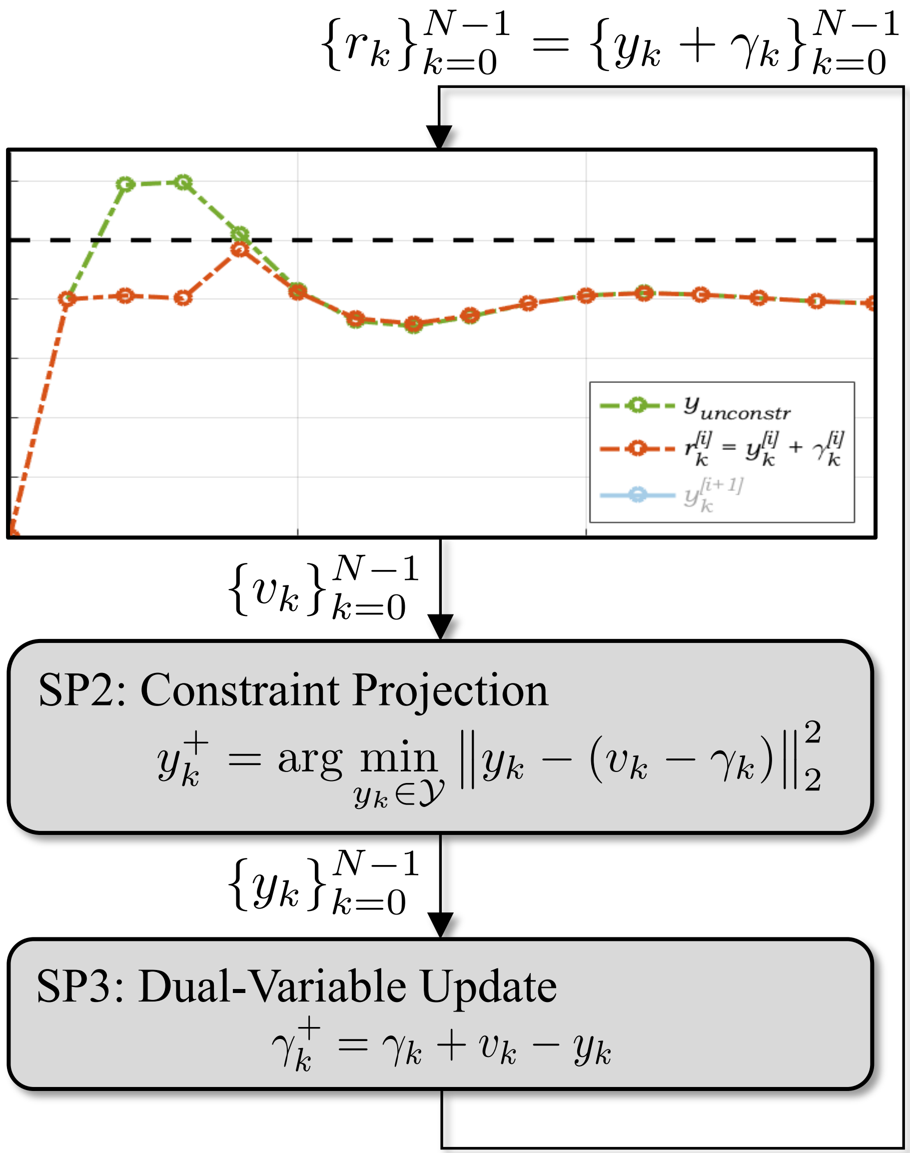
- Sub-problems 2+3 provide integral-action on the reference to avoid constrain violations



Alternating Direction Method of Multipliers Algorithm

Iteratively solve Augmented Lagrangian QP

1. Solve for states, inputs, and unconstrained outputs x_k, u_k, v_k
2. Solve for constrained outputs y_k
3. Solve for dual-variables γ_k



Intuition:

- Sub-problem 1 trades-off tracking the unconstrained optimal and reference that avoids constraints

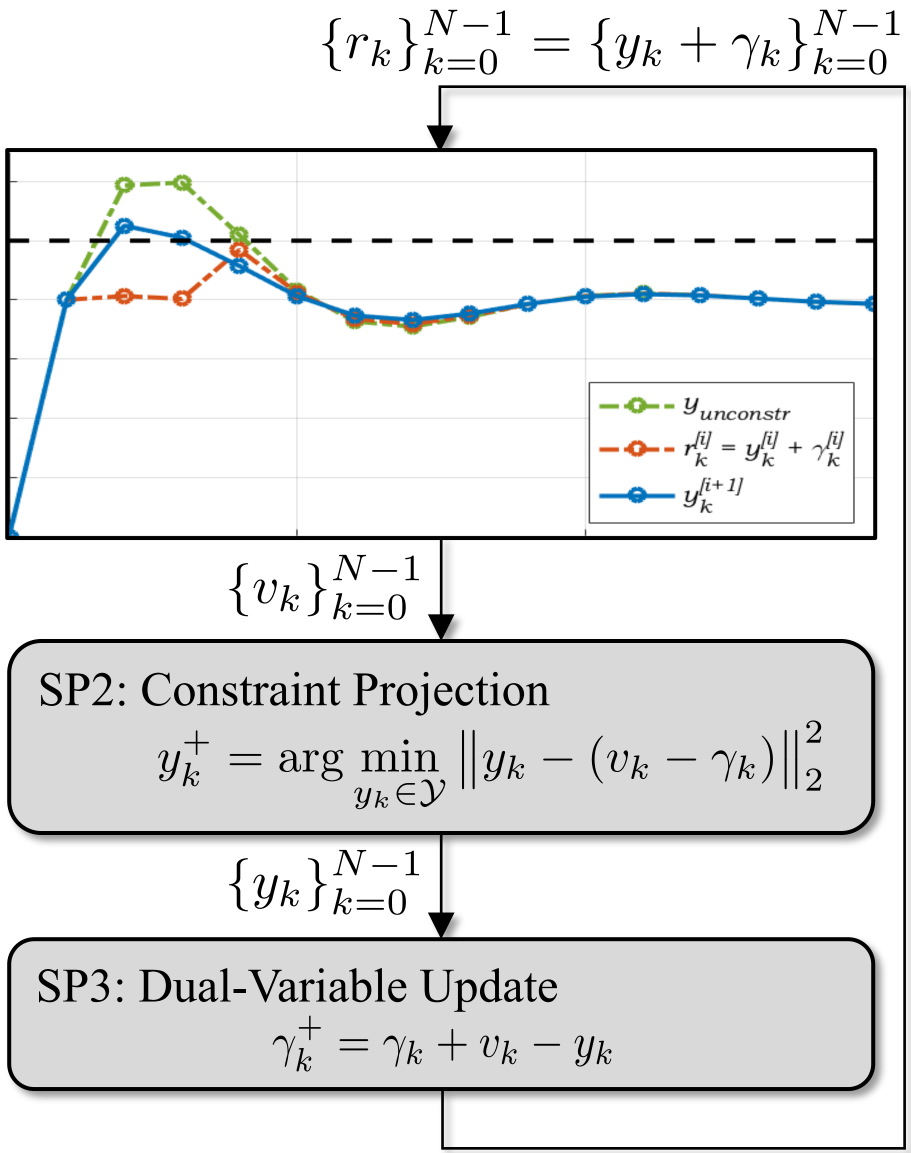
$$\{r_k\}_{k=0}^{N-1} = \{y_k + \gamma_k\}_{k=0}^{N-1}$$

- Sub-problems 2+3 provide integral-action on the reference to avoid constrain violations

Alternating Direction Method of Multipliers Algorithm

Iteratively solve Augmented Lagrangian QP

1. Solve for states, inputs, and unconstrained outputs
   x_k, u_k, v_k
2. Solve for constrained outputs y_k
3. Solve for dual-variables γ_k



Intuition:

- Sub-problem 1 trades-off tracking the unconstrained optimal and reference that avoids constraints

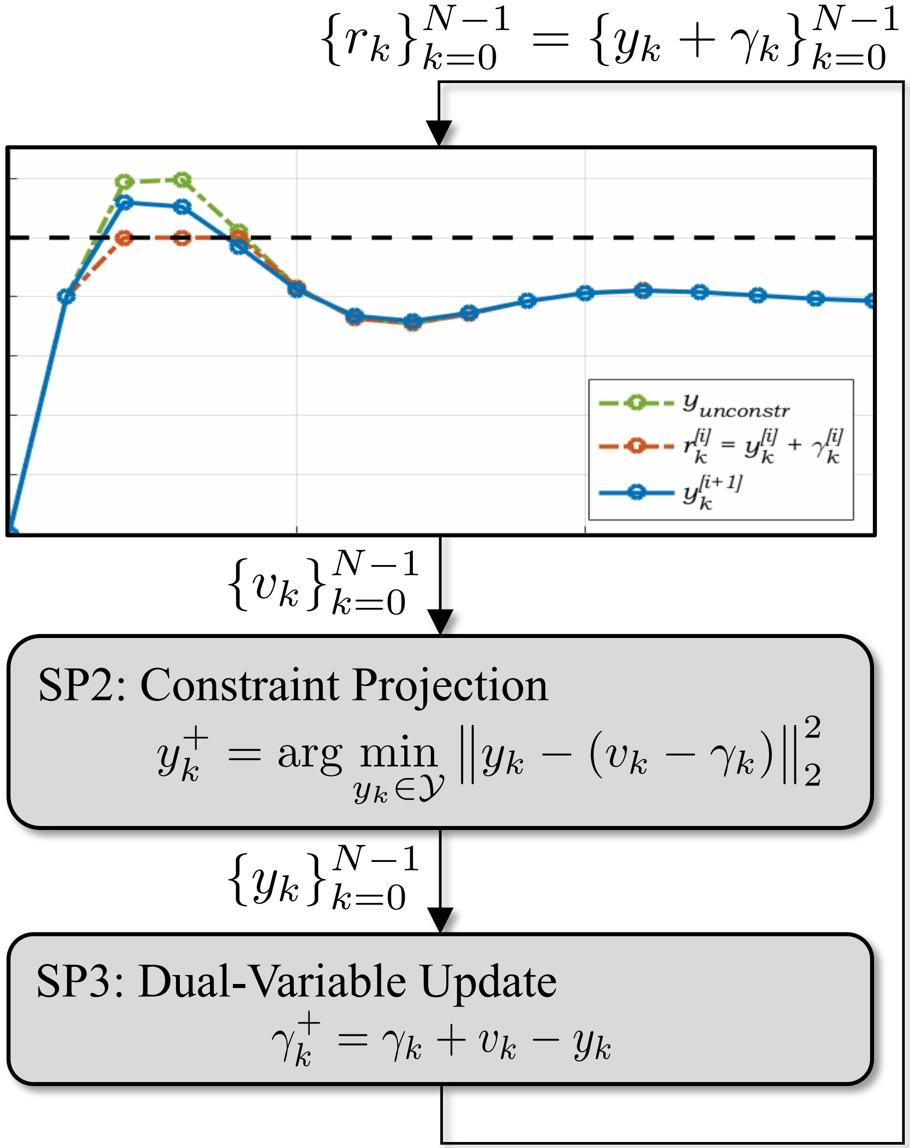
$$\{r_k\}_{k=0}^{N-1} = \{y_k + \gamma_k\}_{k=0}^{N-1}$$

- Sub-problems 2+3 provide integral-action on the reference to avoid constrain violations

Alternating Direction Method of Multipliers Algorithm

Iteratively solve Augmented Lagrangian QP

1. Solve for states, inputs, and unconstrained outputs
   x_k, u_k, v_k
2. Solve for constrained outputs y_k
3. Solve for dual-variables γ_k



Intuition:

- Sub-problem 1 trades-off tracking the unconstrained optimal and reference that avoids constraints

$$\{r_k\}_{k=0}^{N-1} = \{y_k + \gamma_k\}_{k=0}^{N-1}$$

- Sub-problems 2+3 provide integral-action on the reference to avoid constrain violations

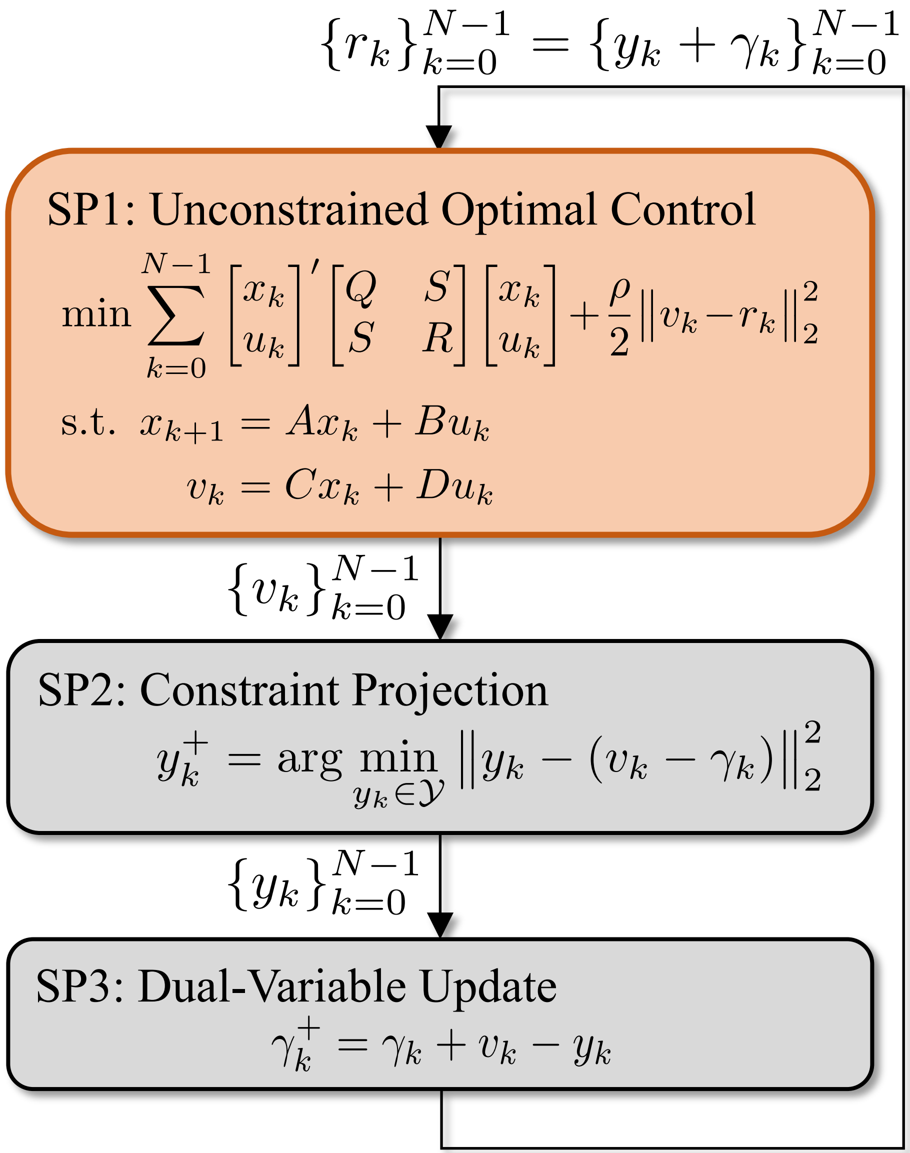
Symmetric ADMM

Computational bottleneck is solving the unconstrained optimal control problem

- SP1: $O(n^2)$ / $O(n^3)$
- SP2 & SP3: $O(n)$

Exploit symmetry to reduce the computational complexity

- Symmetric Decomposition



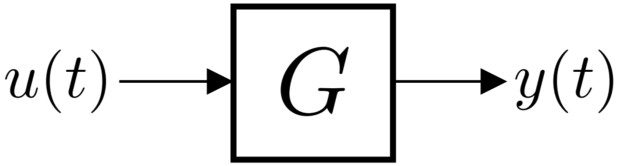
Symmetric Decomposition

Use Schur's lemma to decompose linear systems

Schur's Lemma

Symmetric matrices have input subspaces that only affect the corresponding output subspace

- From representation theory of linear groups



$$u(t) \in \mathbb{U}_1 \Rightarrow y(t) \in \mathbb{Y}_1$$

$$u(t) \in \mathbb{U}_1^\perp \Rightarrow y(t) \in \mathbb{Y}_1^\perp$$

Decompose dynamics of symmetric systems

- Symmetric cost decompose similarly
- Decomposition is both numerically and dynamically robust

$$\Phi_{y,i}^* G \Phi_{u,j} = \begin{cases} \hat{G}_{ii} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Symmetric Decomposition: Example

Example: 2-masses (4 states, 2 inputs/outputs)

- Reflective symmetry
- In-phase forces will produce in-phase displacements

$$u_1(t) = u_2(t) \Rightarrow y_1(t) = y_2(t)$$

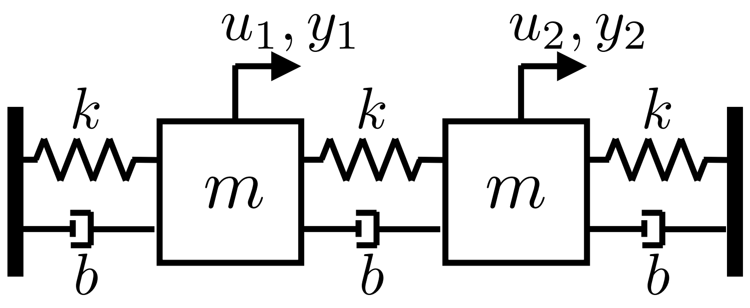
$$\Phi_y^1 = \Phi_u^1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

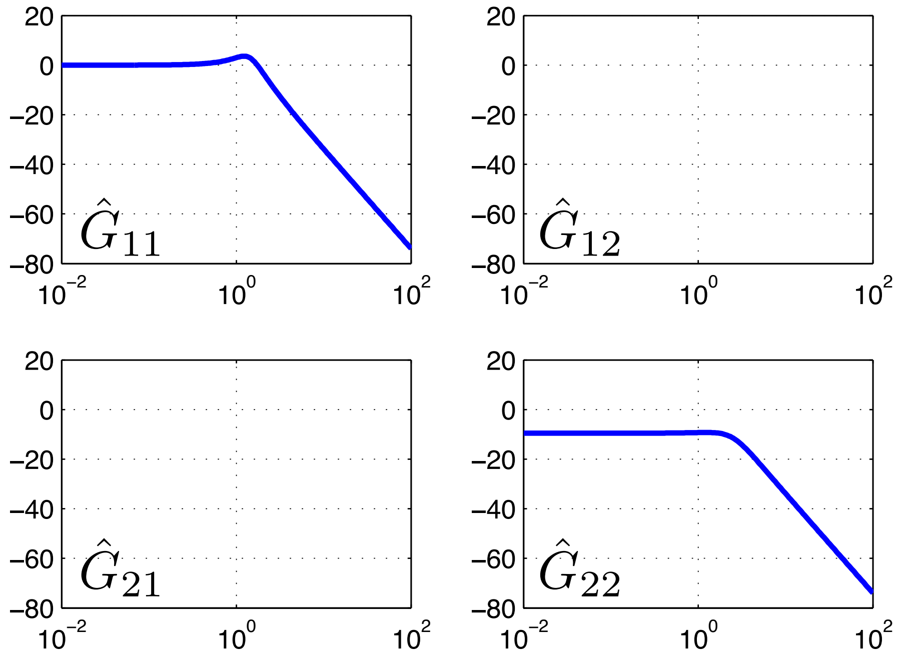
- Out-of-phase forces will produce out-of-phase displacements

$$u_1(t) = -u_2(t) \Rightarrow y_1(t) = -y_2(t)$$

$$\Phi_y^2 = \Phi_u^2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\Phi_{y,i}^* G \Phi_{u,j} = \begin{cases} \hat{G}_{ii} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$





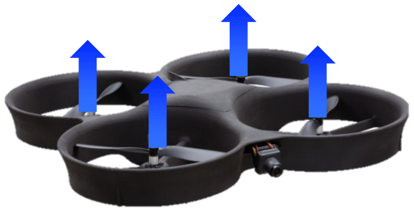
Symmetric Decomposition: Example

Example: Quadrotor (12 states, 4 inputs)

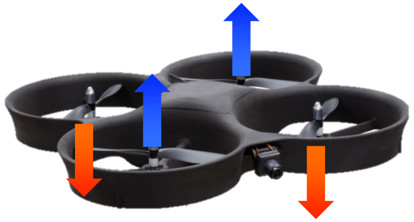
- Dihedral-4 symmetry
- Decomposition into 4 subsystems:
  - *z*-cartesian dynamics: 2 states, 1 input

$$\Phi_{y,i}^{*} G \Phi_{u,j} = \begin{cases} \hat{G}_{ii} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\Phi_u^1 = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

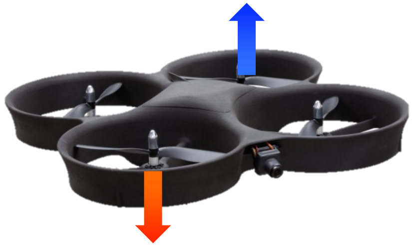


$$\Phi_u^2 = \frac{1}{2} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$



$$\Phi_u^3 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\Phi_u^4 = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ 0 \\ -1 \end{bmatrix}$$



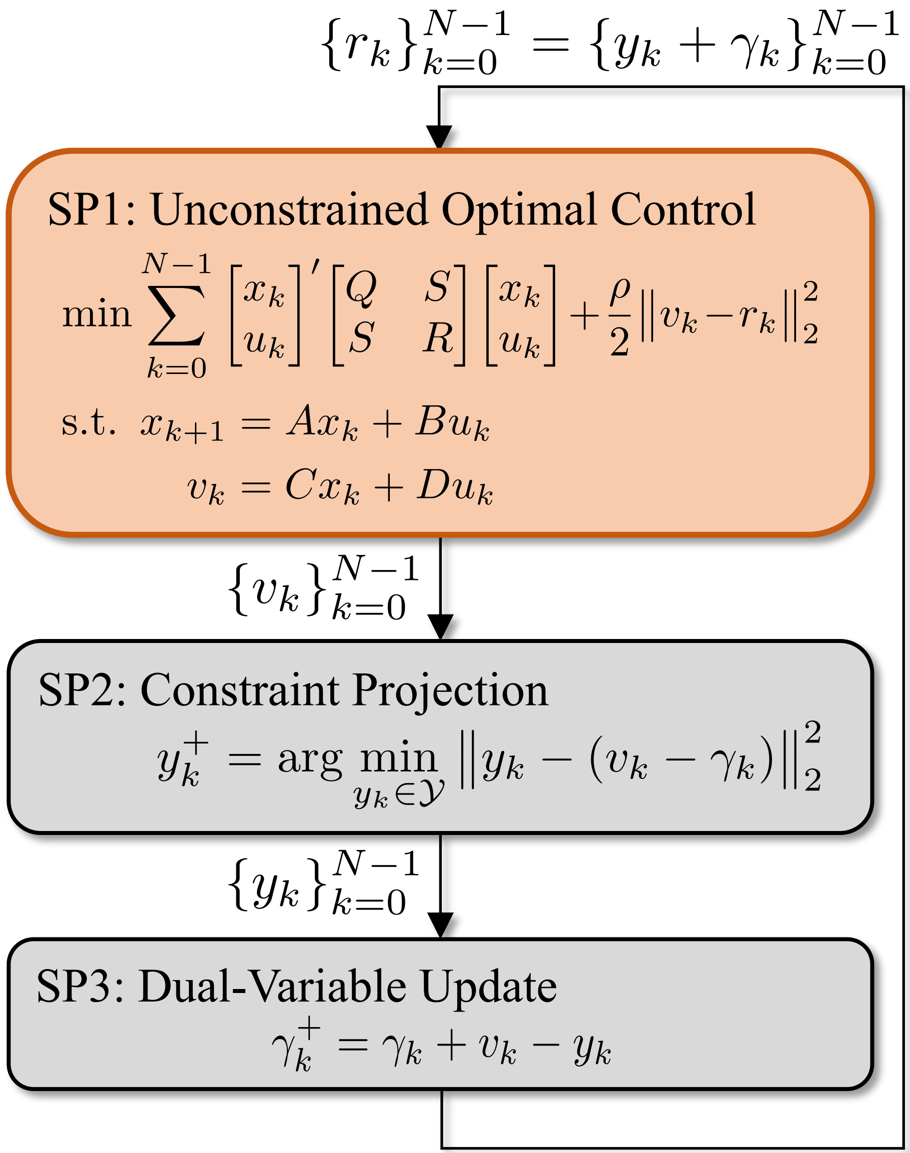
Symmetric ADMM

Use symmetric decomposition to decompose unconstrained optimal control problem

$$\begin{bmatrix} \Phi_x^i & 0 \\ 0 & \Phi_y^i \end{bmatrix}' \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \begin{bmatrix} \Phi_x^j & 0 \\ 0 & \Phi_u^j \end{bmatrix} = \begin{cases} \left[\begin{array}{c|c} \hat{A}_{ii} & \hat{B}_{ii} \\ \hline \hat{C}_{ii} & \hat{D}_{ii} \end{array}\right] \\ 0 \end{cases}$$

$$\begin{bmatrix} \Phi_x^i & 0 \\ 0 & \Phi_u^i \end{bmatrix}' \begin{bmatrix} Q & S \\ S' & R \end{bmatrix} \begin{bmatrix} \Phi_x^j & 0 \\ 0 & \Phi_u^j \end{bmatrix} = \begin{cases} \begin{bmatrix} \hat{Q}_{ii} & \hat{S}_{ii} \\ \hat{S}'_{ii} & \hat{R}_{ii} \end{bmatrix} \\ 0 \end{cases}$$

- Decompose dynamics and constraints
- Completely decoupled unconstrained optimal control problems
- Decoupled sub-problems can be solve sequentially or in parallel



Symmetric ADMM

Use symmetric decomposition to decompose unconstrained optimal control problem

$$\begin{bmatrix} \Phi_x^i & 0 \\ 0 & \Phi_y^i \end{bmatrix}' \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \begin{bmatrix} \Phi_x^j & 0 \\ 0 & \Phi_u^j \end{bmatrix} = \begin{cases} \left[\begin{array}{c|c} \hat{A}_{ii} & \hat{B}_{ii} \\ \hline \hat{C}_{ii} & \hat{D}_{ii} \end{array}\right] \\ 0 \end{cases}$$

$$\begin{bmatrix} \Phi_x^i & 0 \\ 0 & \Phi_u^i \end{bmatrix}' \begin{bmatrix} Q & S \\ S' & R \end{bmatrix} \begin{bmatrix} \Phi_x^j & 0 \\ 0 & \Phi_u^j \end{bmatrix} = \begin{cases} \begin{bmatrix} \hat{Q}_{ii} & \hat{S}_{ii} \\ \hat{S}'_{ii} & \hat{R}_{ii} \end{bmatrix} \\ 0 \end{cases}$$

- Decompose dynamics and constraints
- Completely decoupled unconstrained optimal control problems
- Decoupled sub-problems can be solve sequentially or in parallel

$$\begin{aligned} \min \sum_{k=0}^{N-1} & \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} Q & S \\ S & R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \frac{\rho}{2} \| v_k - r_k \|_2^2 \\ \text{s.t. } & x_{k+1} = A x_k + B u_k \\ & v_k = C x_k + D u_k \end{aligned}$$

$$\Downarrow$$

$$\begin{aligned} \min \sum_{k=0}^{N-1} & \begin{bmatrix} \hat{x}_{ik} \\ \hat{u}_{ik} \end{bmatrix}' \begin{bmatrix} \hat{Q}_{ii} & \hat{S}_{ii} \\ \hat{S}_{ii} & \hat{R}_{ii} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \frac{\rho}{2} \| \hat{v}_{ik} - \hat{r}_{ik} \|_2^2 \\ \text{s.t. } & \hat{x}_{ik+1} = \hat{A}_{ii} \hat{x}_{ik} + \hat{B}_{ii} \hat{u}_{ik} \\ & \hat{v}_{ik} = \hat{C}_{ii} \hat{x}_{ik} + \hat{D}_{ii} \hat{u}_{ik} \end{aligned}$$

$$\vdots \times m$$

$$\begin{aligned} \min \sum_{k=0}^{N-1} & \begin{bmatrix} \hat{x}_{ik} \\ \hat{u}_{ik} \end{bmatrix}' \begin{bmatrix} \hat{Q}_{ii} & \hat{S}_{ii} \\ \hat{S}_{ii} & \hat{R}_{ii} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \frac{\rho}{2} \| \hat{v}_{ik} - \hat{r}_{ik} \|_2^2 \\ \text{s.t. } & \hat{x}_{ik+1} = \hat{A}_{ii} \hat{x}_{ik} + \hat{B}_{ii} \hat{u}_{ik} \\ & \hat{v}_{ik} = \hat{C}_{ii} \hat{x}_{ik} + \hat{D}_{ii} \hat{u}_{ik} \end{aligned}$$

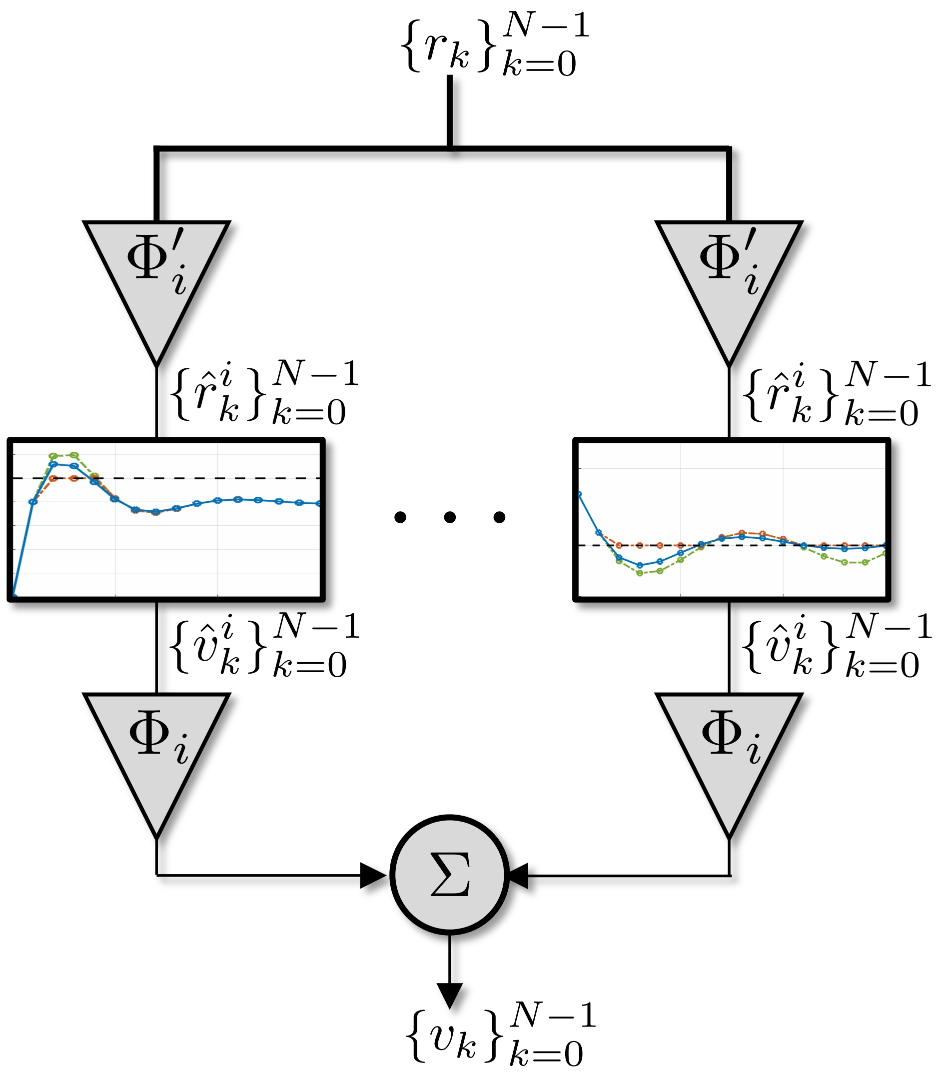
Symmetric ADMM

Decomposition of Sub-Problem 1: Unconstrained Optimal Control Problem

1. Project reference onto subspaces
2. Solve optimal control problem
3. Lift and combine outputs

Sub-Problems 2&3 solved normally

- Dual-variable updates can also be decomposed (w/ minimal benefit)



Symmetric ADMM

Computational Complexity

- Decomposing Sub-Problem 1 does not change convergence rate
  - Transformations are orthogonal
  - # of iterations for symmetric ADMM = # of iterations for baseline ADMM

Symmetric ADMM

Computational Complexity

- Decomposing Sub-Problem 1 does not change convergence rate
    - Transformations are orthogonal
    - # of iterations for symmetric ADMM = # of iterations for baseline ADMM

+Decomposing Sub-Problem 1 does reduce cost of solving optimal control problem
    - Sequential: reduction by m
    - Parallel: reduction by m^2
    - $m \leq n$ is related to size of symmetry group

Symmetric ADMM

Computational Complexity

- Decomposing Sub-Problem 1 does not change convergence rate
  - Transformations are orthogonal
  - # of iterations for symmetric ADMM = # of iterations for baseline ADMM

+ Decomposing Sub-Problem 1 does reduce cost of solving optimal control problem
  - Sequential: reduction by m
  - Parallel: reduction by m^2
  - $m \leq n$ is related to size of symmetry group

– Addition cost required to project/lift Sub-Problem 1
  - Overall reduction if cost of performing transformations is sub-quadratic $< O(m^2)$
  - Generally, not true e.g. singular value decomposition $O(n^2) \geq O(m^2)$
  - Cyclic-m/dihedral-m groups: $O(m \log m)$ (Fast Fourier Transform)
  - Symmetric-m/hyperoctohedral-m groups: $O(m)$
  - 4 symmetry groups (cyclic/dihedral/symmetric/hyperoctohedral) cover most applications

Outline

- Personal Overview
  - Background
  - Research Interests
- Symmetry
  - Model Predictive Control (MPC)
  - Symmetric MPC
  - Symmetric Explicit MPC
  - Symmetric Implicit MPC
    - Symmetric Alternating Direction Method of Multipliers (ADMM)
  - Example – Symmetric HVAC

Application: Symmetric HVAC

Heating ventilation and air-conditioning (HVAC)

- Cost Function:
  - Room temperature tracking
  - Energy consumption
- Dynamics:
  - Thermodynamics
  - Fluid-dynamics
  - Heat-transfer
- Constraints:
  - Maintain sub-cooling/super-heating in indoor units
  - Ensure gas entering compressor
  - Limits on valve and compressor

Constrained Optimal Control Problem

$$\min \quad p(x_{N|t}) + \sum_{k=0}^{N-1} q(x_{k|t}, u_{k|t})$$

$$\text{s.t.} \quad x_{k+1|t} = f(x_{k|t}, u_{k|t})$$

$$x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}$$

$$x_{N|t} \in \mathcal{C}$$

Application: Symmetric HVAC

Symmetry:

- Due to repeated components
  - Same heat-exchangers, fans, sensors in each room
  - Refrigerant flows through pipes of similar diameter
  - Rooms have different thermal masses, pipes have different lengths

- Behavioral symmetry
  - Indoor units have similar dynamics and costs

Constrained Optimal Control Problem

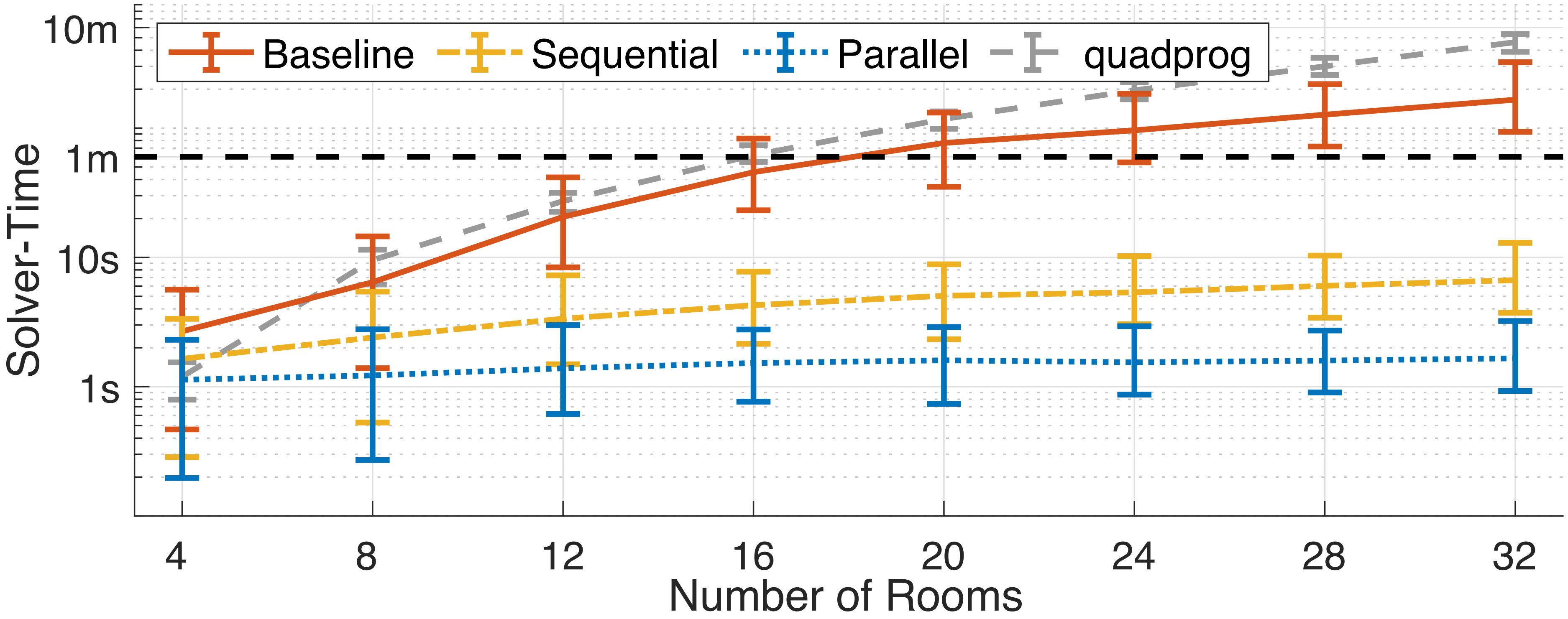
$$\min \quad p(x_{N|t}) + \textstyle\sum_{k=0}^{N-1} q(x_{k|t}, u_{k|t})$$

$$\text{s.t.} \quad x_{k+1|t} = f(x_{k|t}, u_{k|t})$$

$$x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}$$

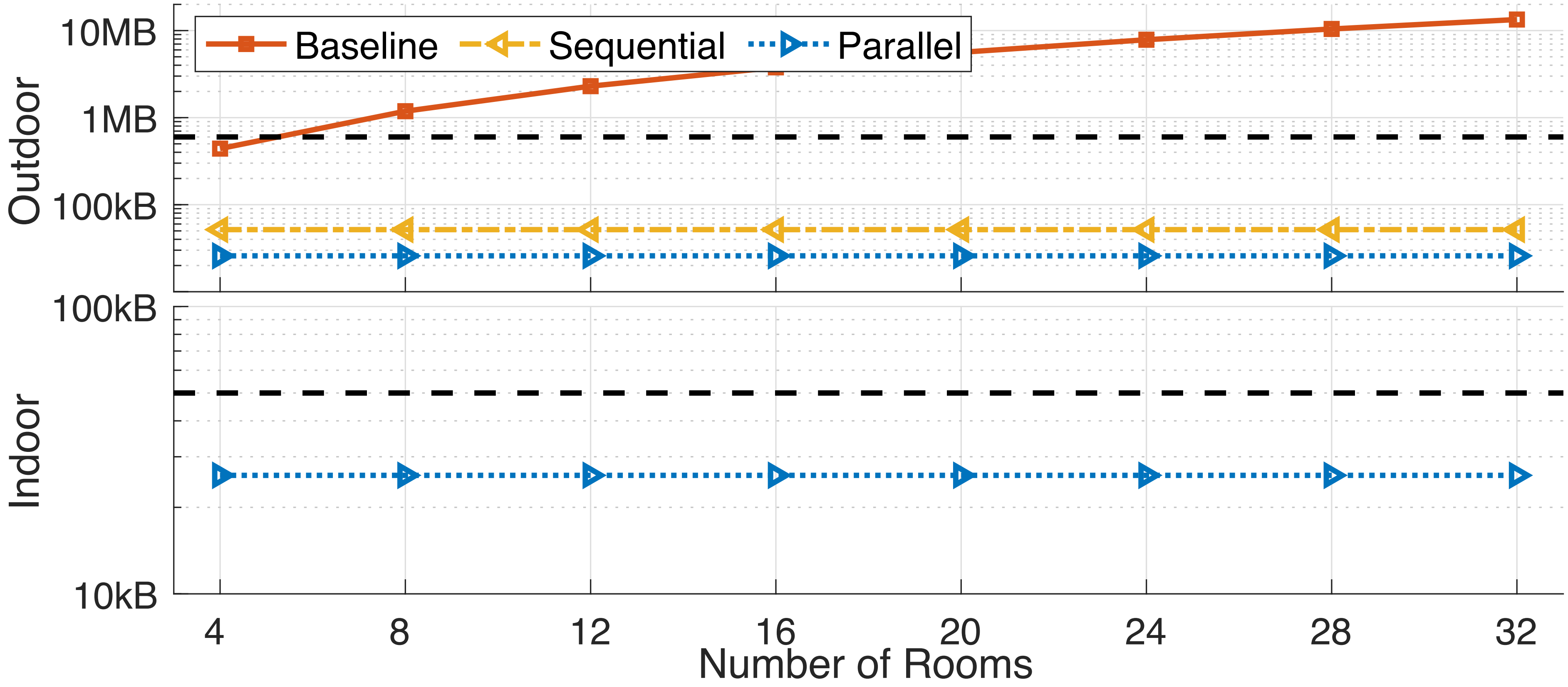
$$x_{N|t} \in \mathcal{C}$$

Application: Symmetric HVAC



- Test emulates embedded processor performance
    - Disabled multi-threading, code acceleration, brute-force matrix computation
- Optimization problem always solved within the allotted time

Application: Symmetric HVAC



- Memory requires below allotted space

Thanks for your attention

Questions?

cdanielson@unm.edu