Identification of the Symmetries of Linear Systems with Polytopic Constraints

Claus Danielson and Francesco Borrelli

Abstract—In this paper we consider the problem of finding all the state-space and input-space transformations that preserve the parameters of a constrained linear system. Such transformations are called symmetries. For systems constrained by bounded polytopes the set of all symmetries is a finite group and requires techniques from discrete mathematics to find. We transform the problem of finding the symmetries of a constrained linear system into the problem of finding the symmetries of a vertex colored graph. The symmetries of a vertex colored graph can be found efficiently using graph automorphism software. We demonstrate our symmetry identification procedure on a quadcopter example.

I. INTRODUCTION

This paper presents a procedure for identifying the symmetries of linear systems with bounded polytopic constraints. Symmetries of linear systems are state-space and input-space transformation that preserve the system parameters. In particular the state-space matrices and constraint sets are mapped to themselves.

Symmetry has been used extensively in numerous fields to reduce computational complexity. In recent years symmetry has been applied to optimization to solve linear-programs [3], semi-definite programs [12], and integer-programs [4]. In [5] symmetry was used to find the transition probabilities for Markov-Chains that produce the fastest convergence. In [8] symmetry was exploited to reduce the computational complexity of $H_2$ and $H_{\infty}$ control design. In [17] symmetry was used to simplify the analysis and control design for large-scale distributed parameter system.

In previous works the systems considered were unconstrained. For unconstrained systems linear algebra techniques can be used to find the system symmetries. However when the system is constrained the problem is more difficult. In this case discrete-mathematics techniques must be used to find the symmetries. In this paper we consider systems with bounded polytopic constraints. We show how the problem of finding the symmetries of the constrained system can be transformed into the problem of finding the symmetries of a vertex-colored graph. Graph symmetries can be efficiently found using graph automorphism software [16], [11], [14].

We begin this paper by motivating our work with an application to model predictive control. In Section II we formally define the symmetry identification problem for constrained systems and compare with the unconstrained case. In Section III we show that the symmetry identification problem is equivalent to finding the set of permutation matrices that commute with a specifically designed set of matrices. In Section IV we show how the set of all permutation matrices that commute with a set of matrices can be identified using graph theory. Finally in Section V we summarize our results and present an example to demonstrate our software.

A. Motivation: Symmetric Explicit Model Predictive Control

In [9] we presented a method for exploiting the symmetries of a constrained dynamic system to reduce the memory requirement for explicit model predictive control. In this section we briefly summarize our results from that paper.

Consider the following model predictive control problem for a constrained linear time-invariant system with quadratic cost

\[
\begin{align*}
\min_{u_0, \ldots, u_{N-1}} & \quad x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \\
\text{subject to:} & \quad x_{k+1} = A x_k + B u_k \\
& \quad x_{k+1} \in X, u_k \in U \quad \text{for} \quad k = 0, \ldots, N
\end{align*}
\]

where $x_0 = x(t) \in \mathbb{R}^n_x$ is the measured state, $x_k \in \mathbb{R}^n_x$ is the predicted state under control action $u_k \in \mathbb{R}^n_u$ over the horizon $N$, $X$ and $U$ are the polytopic state and input constraint sets, and $P = P^T \succeq 0$, $Q = Q^T \succeq 0$, $R = R^T \succ 0$.

In [1] it was shown that (1) can be solved parametrically to produce a continuous piecewise affine on polytope controller

\[
u_k = \kappa(x(t)) = \begin{cases} 
F_1 x(t) + G_1 & \text{for } x(t) \in R_1 \\
\vdots & \\
F_p x(t) + G_p & \text{for } x(t) \in R_p
\end{cases}
\]

where $R_i \subseteq X$ are polytopes called critical regions, $F_i, G_i \in \mathbb{R}^{n_x \times n_x}$ are the feedback matrices and $G_i \in \mathbb{R}^{n_x}$ are the feedforward vectors. The triple $(F_i, G_i, R_i)$ is called the $i$-th controller piece and $I = \{1, \ldots, p\}$ is the set of all controller pieces.

A symmetry of the explicit model predictive controller (2) is a state-space $\Theta \in \mathbb{R}^{n_x \times n_x}$ and input-space $\Omega \in \mathbb{R}^{n_u \times n_u}$ transformation that permutes the controller pieces $I$. The transformation pair $(\Theta, \Omega)$ preserves the control law $\Omega^{-1} \Theta \kappa(x) = \kappa(x)$. Definition 1: A pair of invertible matrices $(\Theta, \Omega)$ is a symmetry of (2) if for every $i \in I$ there exists a $j \in I$ such that

\[
\begin{align*}
\Omega F_i \Theta^{-1} & = F_j \\
\Omega G_i & = G_j \\
\Theta \circ R_i & = R_j
\end{align*}
\]

The set of all symmetries of the controller (2) form a group called the symmetry group $\text{Aut}(\kappa)$ of the controller.

In [9] we showed how controller symmetry can be used to reduce the storage requirements for explicit model predictive control by eliminating symmetrically redundant controller pieces. In addition we proved that the controller (2) is symmetry with respect to the intersection of the symmetry groups of the cost function (1a), dynamics (1b), and constraints (1c), of the model predictive control problem. In this paper we show how to find the symmetry groups of the dynamics and constraints. Moreover we provide methods for both finding the symmetry of the cost function (1a) or for modifying the cost function (1a) so that it is symmetric with respect to the symmetry group of the dynamics (1b), and constraints (1c).

B. Mathematical Background

A half-space is the set $\{x \in \mathbb{R}^n : h x \leq k\}$ where $h^T \in \mathbb{R}^n$ and $k \in \mathbb{R}$. A polytope $P \subseteq \mathbb{R}^n$ is the intersection of a finite number of half-spaces $P = \{x \in \mathbb{R}^n : h_i x \leq k_i, i = 1, \ldots, m\}$ where $m$ is the number of half-spaces. The half-spaces parameters $h_i$ and $k_i$ for $i = 1, \ldots, m$ can be collected in the matrix $H \in \mathbb{R}^{m \times n}$ and vector $K \in \mathbb{R}^m$. A half-space is redundant if removing it from the description of $P$ does not change the set. If the polytope $P$
contains the origin in its interior then it can be normalized \( P = \{ x \in \mathbb{R}^n : Hx \leq 1 \} \) where \( K = 1 \in \mathbb{R}^m \) is the vector of ones. Let \( M \in \mathbb{R}^{m \times n} \) and \( \mathcal{P} \subseteq \mathbb{R}^n \) then \( M \circ \mathcal{P} = \{ Mx : x \in \mathcal{P} \} \).

Let \( \mathcal{N} = \{ 1, \ldots, n \} \subset \mathbb{N} \) be a finite set. A permutation \( \sigma \) is a bijective function from \( \mathcal{N} \) to itself. Permutations will be denoted using the Cayley notation. For instance the permutation \( \sigma \) on \( \mathcal{N} = \{ 1, 2, 3 \} \) such that \( \sigma(1) = 2, \sigma(2) = 3, \) and \( \sigma(3) = 1 \) will be denoted
\[
\sigma = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}.
\]
A permutation matrix \( P \in \mathbb{R}^{n \times n} \) is a square binary matrix that has exactly one 1 in each row and column. Each permutation \( \sigma \) corresponds to a permutation matrix \( P \in \mathbb{R}^{n \times n} \) given by
\[
P_{\sigma} = \begin{cases} 1 & \text{if } i = \sigma(j) \\ 0 & \text{otherwise.} \end{cases}
\]

A group \( (G, +) \) is a set \( G \) along with a binary operator \( + \) such that the operator \( + \) is associative, the set \( G \) is closed under the operation \( + \), the set \( G \) includes an identity element \( e \in G \), and the inverse \( g^{-1} \in G \) of each element \( g \in G \). A group is called trivial if it contains only the identity element \( e \). A group is called finite if it has a finite number of elements \( |G| \). A set \( S \subseteq G \) is called a generating set if every element of \( G \) can be expressed as a product of the elements of \( S \). The group generated by the set \( S \) is denoted by \( \langle S \rangle \).

Let \( G \) be a group. A group homomorphism \( f : G \rightarrow H \) is a function from \( G \) to \( H \) satisfying \( f(g + h) = f(g) \circ f(h) \) for all \( g, h \in G \). A group isomorphism is a bijective group homomorphism. Groups \( G \) and \( H \) are isomorphic if there exists an isomorphism between them. Let \( S \subseteq G \) be a generating for \( G \) and \( f \) a group isomorphism from \( G \) to \( H \) then \( f(S) \) is a generating set for \( H \).

In this paper we consider two types of groups: matrix groups and permutation groups. In matrix groups the elements of \( G \) are matrices. With abuse of terminology the elements of a permutation group \( G \) can be permutations or the corresponding permutation matrices.

An undirected graph \( \Gamma = (\mathcal{N}, \mathcal{E}) \) is a set of vertices \( \mathcal{N} \) together with a list of unordered pairs \( \mathcal{E} \subset \mathcal{N} \times \mathcal{N} \) called edges. A graph coloring \( C : \mathcal{N} \rightarrow \mathcal{C} \) assigns a color \( C(i) \in \mathcal{C} \) to each vertex \( i \in \mathcal{N} \) of the graph. A graph symmetry is a permutation \( \sigma \) of the vertex set \( \mathcal{N} \) that preserves the edge set \( \mathcal{E} \) and vertex coloring \( C(\sigma(i)) = C(i) \) where \( \sigma((i, j)) = \{ \sigma(i), \sigma(j) \} \).

II. PROBLEM STATEMENT

We consider a constrained dynamic system \( \Sigma \) modeled by
\[
\begin{align*}
x(t+1) &= Ax(t) + Bu(t) \\
x(t) &\in \mathcal{X}, \ u(t) \in \mathcal{U}
\end{align*}
\]
where \( A \in \mathbb{R}^{n_x \times n_x} \) is the dynamics matrix, \( B \in \mathbb{R}^{n_x \times n_u} \) is the input matrix, \( \mathcal{X} \subseteq \mathbb{R}^{n_x} \) is the state constraint set and \( \mathcal{U} \subseteq \mathbb{R}^{n_u} \) is the input constraint set. The constraint sets are full-dimensional, bounded polytopes containing the origin in their interior
\[
\begin{align*}
\mathcal{X} &= \{ x \in \mathbb{R}^{n_x} : H_x x \leq 1 \} \\
\mathcal{U} &= \{ u \in \mathbb{R}^{n_u} : H_u u \leq 1 \}
\end{align*}
\]
where \( H_x \in \mathbb{R}^{m_x \times n_x} \) and \( H_u \in \mathbb{R}^{m_u \times n_u} \). We assume that the half-spaces that define \( \mathcal{X} \) and \( \mathcal{U} \) are not redundant.

The symmetry group of a constrained dynamic system is the intersection of the symmetry group of the dynamics and constraints. These symmetry groups are defined below.

**Definition 2:** The symmetry group \( \text{Aut}(A, B) \) of the dynamics (6a) is the set of all pairs of invertible matrices \( (\Theta, \Omega) \) satisfying
\[
\begin{align*}
\Theta A &= A \Theta \\
\Theta B &= B \Omega
\end{align*}
\]

**Definition 3:** The symmetry group \( \text{Aut}(\mathcal{X}, \mathcal{U}) \) of the constraints (6b) is the set of all pairs of invertible matrices \( (\Theta, \Omega) \) satisfying
\[
\Theta \circ \mathcal{X} = \mathcal{X}, \quad \Omega \circ \mathcal{U} = \mathcal{U}
\]

The symmetry group \( \text{Aut}(\Sigma) = \text{Aut}(A, B) \cap \text{Aut}(\mathcal{X}, \mathcal{U}) \) of the constrained dynamic system (6) is the intersection of the symmetry groups of the dynamics \( \text{Aut}(A, B) \) and constraints \( \text{Aut}(\mathcal{X}, \mathcal{U}) \). The symmetry group \( \text{Aut}(\Sigma) \) is the set of all transformations of the state-space \( \Theta \) and input-space \( \Omega \) that preserve the system parameters \( A, B, \mathcal{X}, \) and \( \mathcal{U} \). The symmetry identification problem is the problem of finding a set of generators for the symmetry group \( \text{Aut}(\Sigma) \). We formally define this problem below.

**Problem 1 (Symmetry Identification):** Let \( \text{Aut}(\Sigma) \) be the set of all pairs of matrices \( (\Theta, \Omega) \) satisfying (8) and (9). Find a set \( \text{Gen}(\text{Aut}(\Sigma)) \) that generates the symmetry group \( \text{Aut}(\Sigma) \).

The related problem of finding the symmetry group \( \text{Aut}(A, B) \) for the unconstrained linear system (6a) is much simpler since the closure of \( \text{Aut}(A, B) \) is a vector-space [13]. On the other hand the symmetry group \( \text{Aut}(\mathcal{X}, \mathcal{U}) \), and therefore \( \text{Aut}(\Sigma) \subseteq \text{Aut}(\mathcal{X}, \mathcal{U}) \), is typically discrete. Therefore we cannot use linear algebra to find the symmetry group \( \text{Aut}(\Sigma) \). In this case methods from discrete-mathematics must be used to find the symmetry group \( \text{Aut}(\Sigma) \).

**Proposition 1:** Let \( \mathcal{X} \) and \( \mathcal{U} \) be full-dimensional and bounded sets. Then \( \text{Aut}(\Sigma) \) is a finite group.

**Proof:** See [10]

In the case where the symmetry group \( \text{Aut}(\Sigma) \) is finite we use discrete mathematics techniques to find the group \( \text{Aut}(\Sigma) \).

III. SYMMETRY IDENTIFICATION

In this section we present our methodology for solving Problem 1 (Symmetry Identification).

The symmetry group \( \text{Aut}(\Sigma) \) of the constrained dynamic system (6) is a subgroup of the symmetry group \( \text{Aut}(\mathcal{X}, \mathcal{U}) \) of the constraint sets \( \mathcal{X} \) and \( \mathcal{U} \). For bounded sets \( \mathcal{X} \) and \( \mathcal{U} \) the symmetry group \( \text{Aut}(\mathcal{X}, \mathcal{U}) \) is isomorphic to a permutation group acting on the non-redundant half-spaces of \( \mathcal{X} \) and \( \mathcal{U} \). In other words \( \Theta \) and \( \Omega \) permute the facets of \( \mathcal{X} \) and \( \mathcal{U} \) respectively. We will denote this permutation group by \( \text{Perm}(\mathcal{X}, \mathcal{U}) \). The symmetries \( \Theta, \Omega \in \text{Aut}(\mathcal{X}, \mathcal{U}) \) and \( (P_x, P_u) \in \text{Perm}(\mathcal{X}, \mathcal{U}) \) are related by the expression
\[
\begin{bmatrix}
P_x & 0 \\
0 & P_u
\end{bmatrix}
\begin{bmatrix}
H_x \\
H_u
\end{bmatrix}
= 
\begin{bmatrix}
H_x \\
H_u
\end{bmatrix}
\begin{bmatrix}
\Theta \\
0 \\
0 \Omega
\end{bmatrix}
\]
where the rows of \( H_x \) and \( H_u \) are the half-spaces of \( \mathcal{X} \) and \( \mathcal{U} \) respectively, and \( P_x \in \mathbb{R}^{m_x \times m_x} \) and \( P_u \in \mathbb{R}^{m_u \times m_u} \). Taking the pseudo-inverse of \( H_x \) and \( H_u \) we can express \( \Theta \) and \( \Omega \) in terms of \( P_x \) and \( P_u \) by
\[
\Theta = (H_x^T H_x)^{-1} H_x^T P_x H_x
\]
and
\[
\Omega = (H_u^T H_u)^{-1} H_u^T P_u H_u
\]
In [7] it was shown that the permutation group Perm(X,U) is
the set of all matrices $P_x$ and $P_u$ that satisfy the commutative relation
\[
\begin{bmatrix}
P_x & 0 \\
0 & P_u
\end{bmatrix}
\begin{bmatrix}
H_x(H_x^TH_x)^{-1}H_x^T & 0 \\
0 & H_u(H_u^TH_u)^{-1}H_u^T
\end{bmatrix}
= 
\begin{bmatrix}
H_x(H_x^TH_x)^{-1}H_x^T & 0 \\
0 & H_u(H_u^TH_u)^{-1}H_u^T
\end{bmatrix}
\begin{bmatrix}
P_x & 0 \\
0 & P_u
\end{bmatrix}
\] (12)

In Section IV we will describe the procedure for finding all permutation matrices $P_x$ and $P_u$ that commute with some matrix.

For the matrices $(\Theta, \Omega) \in \text{Aut}(X,U)$ to be elements of the symmetry group $\text{Aut}(\Sigma) \subseteq \text{Aut}(X,U)$ they must also map the dynamics matrices $A$ and $B$ to themselves. This suggests that we can construct $\text{Aut}(\Sigma)$ by selecting the elements of $\text{Aut}(X,U)$ that commute with $P_x$ and $P_u$. Theorem 1: Let $\text{Perm}(\Sigma)$ be the set of all permutation matrices $P_x \in \mathbb{R}^{m_x \times m_x}$ and $P_u \in \mathbb{R}^{m_u \times m_u}$ that satisfy (12) and (17). Then the group $\text{Perm}(\Sigma)$ is isomorphic to the symmetry group $\text{Aut}(\Sigma)$ of the constrained dynamic system (6) with isomorphism (11).

\textbf{Proof:} See [10]

Our procedure for identifying the symmetry group $\text{Aut}(\Sigma)$ (Problem 1) can be summarized by Procedure 1. In the next section we will describe the procedure for finding the group $\text{Perm}(\Sigma)$.

\begin{enumerate}
\item Find a generating set $\text{Gen}(\text{Perm}(\Sigma))$ for the group $\text{Perm}(\Sigma)$
\item Calculate the generators $\text{Gen}(\text{Aut}(\Sigma))$ for the symmetry group $\text{Aut}(\Sigma)$ from $\text{Gen}(\text{Perm}(\Sigma))$ using the isomorphism (11)
\end{enumerate}

A. Symmetry of the Cost

The symmetry group $\text{Aut}(\text{MPC})$ of the model predictive control problem (1) is the intersection of the symmetry groups of the cost (1a), dynamics (1b), and constraints (1c). So far we have shown how to find the symmetry group $\text{Aut}(\Sigma) \supseteq \text{Aut}(\text{MPC})$ of the dynamics and constraints. There are two ways we can address the symmetry of the cost function; we can find the symmetries $\text{Aut}(J)$ of the cost function or we can modify the cost function so that it is symmetric with respect to the symmetries $\text{Aut}(\Sigma)$ of the dynamics and constraints.

Definition 4: The symmetry group $\text{Aut}(J)$ of the quadratic cost function (1a) is the set of all pairs of invertible matrices $(\Theta, \Omega)$ satisfying
\[
\Theta^T P \Theta = P \quad (18a)
\]
\[
\Theta^T Q \Theta = Q \quad (18b)
\]
\[
\Omega^T R \Omega = R \quad (18c)
\]
The symmetry group $\text{Aut}(\text{MPC}) = \text{Aut}(\Sigma) \cap \text{Aut}(J)$ can be found by finding the set of all permutations $(P_x, P_u) \in \text{Perm}(\Sigma)$ that satisfy the commutative relations
\[
\begin{bmatrix}
P_x & 0 \\
0 & P_u
\end{bmatrix}
\begin{bmatrix}
H_x & 0 \\
0 & H_u
\end{bmatrix}
= 
\begin{bmatrix}
H_x & 0 \\
0 & H_u
\end{bmatrix}
\begin{bmatrix}
P_x & 0 \\
0 & P_u
\end{bmatrix}
\] (19)
and
\[
\begin{bmatrix}
P_x & 0 \\
0 & P_u
\end{bmatrix}
\begin{bmatrix}
0 & H_x R_h \Theta \\
0 & 0
\end{bmatrix}
= 
\begin{bmatrix}
0 & H_x R_h \Theta \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
P_x & 0 \\
0 & P_u
\end{bmatrix}
\] (20)
where $H_x = H_x(H_x^TH_x)^{-1}$ and $H_u = H_u(H_u^TH_u)^{-1}$.

Corollary 1: Let $\text{Perm}(\text{MPC})$ be the set of all permutation matrices $P_x$ and $P_u$ that satisfy (12), (17), (19), and (20). Then the group $\text{Perm}(\text{MPC})$ is isomorphic to the symmetry group $\text{Aut}(\text{MPC}) = \text{Aut}(J) \cap \text{Aut}(\Sigma)$ of the model predictive control problem (1) with isomorphism (11).

Alternatively we can modify the cost function (1a) so that it shares the symmetries of the system $\text{Aut}(\Sigma)$. The cost (1a) can be made symmetric with respect to $\text{Aut}(\Sigma)$ by taking its barycenter
\[
J(X,U) = x(N)^T P_x x(N) + \sum_{k=0}^{N-1} x(k)^T Q x(k) + u(k)^T R u(k)
\]
where $U = \{u(k)\}_{k=0}^{N-1}$ is the input trajectory, $X = \{x(k)\}_{k=0}^{N}$ is the state trajectory,
\[
P = \frac{1}{|\Omega|} \sum_{\Theta \in \Omega^T P \Theta}
\]
and likewise for $Q$ and $R$, and $\Omega = \text{Aut}(\Sigma)$.

4220
Proposition 2: The cost function (21) is symmetric with respect to the symmetries $\text{Aut}(\Sigma)$ of the dynamics and constraints.

Proof: Follows directly from the closure of groups [5], [4] i.e. $(\Theta, \Omega) \circ \text{Aut}(\Sigma) = \text{Aut}(\Sigma)$ for all $(\Theta, \Omega) \in \text{Aut}(\Sigma)$.

IV. MATRIX PERMUTATION GROUPS

In this section we present the procedure for finding the set of all block structured permutation matrices that commute with a set of matrices $M_1, \ldots, M_r$. We reformulate this problem as a graph symmetry problem which can be efficiently solved using software packages such Nauty [16], Saucy [11], and Bliss [14].

This procedure can be used to find the generators of the permutation groups $\text{Perm}(\Sigma)$, $\text{Perm}(X,H)$ and $\text{Perm}(\text{MPC})$ which are used to find the generators of the symmetry groups $\text{Aut}(\Sigma)$, $\text{Aut}(X,H)$ and $\text{Aut}(\text{MPC})$ respectively.

A. Permutations that Commute with a Single Matrix

We begin by presenting the procedure for finding the set of all permutation matrices that commute with a single matrix $M$. This problem was first addressed in [16]. Our method is taken from [4].

The group $\text{Perm}(M)$ is defined as the set of all permutation matrices that commute with a matrix $M \in \mathbb{R}^{m \times m}$

$$\text{PM} = MP. \quad (23)$$

We find the group $\text{Perm}(M)$ by constructing a graph $\Gamma_m(M)$ whose symmetry group $\text{Aut}(\Gamma_m(M))$ is isomorphic to the permutation group $\text{Perm}(M)$. The generators of $\text{Aut}(\Gamma_m(M))$ can then mapped to generators for $\text{Perm}(M)$.

Procedure 2 produces a vertex colored graph $\Gamma_m(M)$ whose symmetry group $\text{Aut}(\Gamma_m(M))$ is isomorphic to $\text{Perm}(M)$. The generators of the group $\text{Aut}(\Gamma_m(M))$ are found using graph automorphism software [16], [11], [14].

The graph $\Gamma_m(M)$ contains a vertex for each row $r_i$ for $i = 1, \ldots, m$, each column $c_j$ for $j = 1, \ldots, m$, and each element $e_{ij}$ for $i, j = 1, \ldots, m$ of the matrix $M$. For each unique value $s \in \{M_{ij} : i, j = 1, \ldots, m\}$ of the matrix we create a vertex $s$. The rows, columns, and element vertices are different colors. Each value vertex $s \in \{M_{ij} : i, j = 1, \ldots, m\}$ has a different color. The graph $\Gamma_m(M)$ contains edges that connect each element vertex $e_{ij}$ to the row $r_i$ and column $c_j$ vertices and the value vertex $s = M_{ij}$. If $M_{ij} = M_{kl}$ then the element vertices $e_{ij}$ and $e_{kl}$ are both connected to the same value vertex $s \in \{M_{ij} : i, j = 1, \ldots, m\}$.

The graph $\Gamma_m(M) = (N, E)$ has $|N| = m^2 + 2m + m^2$ vertices where $m_s = |\{M_{ij} : i, j = 1, \ldots, m\}|$ is the number of unique values in the matrix $M$. There are $3 + m^2$ vertex colors. The graph $\Gamma_m(M)$ has $|E| = 3m^2$ edges; each element vertex $e_{ij}$ is neighbors with the row vertex $r_i$, column vertex $c_j$, and the value $s = M_{ij}$ vertex.

The graph automorphism software searches for permutations $\sigma \in \text{Aut}(\Gamma_m(M))$ of the vertex set $N$ which preserves the edge list $E$ and vertex coloring. The graph automorphism software produces a set $\text{Gen(}\text{Aut}(\Gamma_m(M))\text{)}$ of at most $m - 1$ permutations that generate the symmetry group $\text{Aut}(\Gamma_m(M))$.

The graph $\Gamma_m(M)$ is constructed so that the symmetries $\sigma \in \text{Aut}(\Gamma_m(M))$ have a particular structure. The graph automorphism software will not find permutations $\sigma$ that transpose row vertices with non-row vertices since they have different colors. Therefore each $\sigma \in \text{Aut}(\Gamma_m(M))$ will map row vertices $r_i$ to row vertices. Likewise each $\sigma \in \text{Aut}(\Gamma_m(M))$ maps column vertices $c_j$ to column vertices and element vertices $e_{ij}$ to element vertices. The value vertices $s \in \{M_{ij} : i, j = 1, \ldots, m\}$ can only be mapped to themselves since each has a different color.

Let $\sigma_m$ denote the restriction of the permutations $\sigma \in \text{Aut}(\Gamma_m(M))$ to the set of row vertices $N_r = \{r_1, \ldots, r_m\} \subset N$. The edges of the graph $\Gamma_m(M)$ are constructed such that $\sigma_m$ is also the restriction of $\sigma \in \text{Aut}(\Gamma_m(M))$ to the column vertices. The element vertices $e_{ij}$ are permuted according to the expression $\sigma(e_{ij}) = e_{\sigma_m(i)\sigma_m(j)}$. And the value vertices $s \in \{M_{ij} : i, j = 1, \ldots, n\}$ are not permuted by any $\sigma \in \text{Aut}(\Gamma_m(M))$. Therefore each row permutation $\sigma_m$ uniquely defines the permutation $\sigma \in \text{Aut}(\Gamma_m(M))$. Thus without loss of generality we can define the following isomorphism from $\text{Aut}(\Gamma_m(M))$ to $\text{Perm}(M)$ by

$$P_{ij} = \begin{cases} 1 & \text{for } r_j = \sigma_m(r_i) \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

where $\sigma_m \in \text{Aut}(\Gamma_m(M))$. This result is formally stated in the following theorem.

Theorem 2: The symmetry group $\text{Aut}(\Gamma_m(M))$ of the vertex colored graph $\Gamma_m(M)$ is isomorphic to the permutation group $\text{Perm}(M)$ of the square matrix $M$ with isomorphism (24).

Proof: See [4].

This procedure is demonstrated by the following example.

Example 2: The graph $\Gamma_2(I)$ for the $2 \times 2$ identity matrix $I \in \mathbb{R}^{2 \times 2}$ is shown in Figure 1. The green diamonds represent the row vertices $r_1$ and $r_2$ and the blue squares represent the column vertices $c_1$ and $c_2$. The white circles represent the matrix element vertices $e_{ij}$. The purple and yellow hexagons represent the unique matrix values $I_{11} = I_{22} = 1$ and $I_{12} = I_{21} = 0$ respectively.

The graph automorphism software will search for permutations $\sigma$ of the vertex set $N$ that preserve the edge list $E$ and vertex coloring. For instance the permutation

$$\sigma = \begin{pmatrix} r_1 & r_2 & c_1 & c_2 & e_{11} & e_{12} & e_{21} & e_{22} & s_1 & s_0 \\ c_1 & r_2 & c_1 & c_2 & e_{11} & e_{12} & e_{21} & e_{22} & s_1 & s_0 \end{pmatrix} \quad (25)$$

is not a symmetry of $\Gamma_m(M)$ since it transposes vertices of different colors; row vertex $r_1$ and column vertex $c_1$. The permutation

$$\sigma = \begin{pmatrix} r_1 & r_2 & c_1 & c_2 & e_{11} & e_{12} & e_{21} & e_{22} & s_1 & s_0 \\ r_2 & r_1 & c_1 & c_2 & e_{11} & e_{12} & e_{21} & e_{22} & s_1 & s_0 \end{pmatrix} \quad (26)$$

is not a symmetry of $\Gamma_m(M)$ since it changes the edge list; $\{c_2, e_{22}\} \in E$ but $\sigma(\{c_2, e_{22}\}) \notin E$. However the permutation

$$\sigma = \begin{pmatrix} r_1 & r_2 & c_1 & c_2 & e_{11} & e_{12} & e_{21} & e_{22} & s_1 & s_0 \\ r_2 & r_1 & c_1 & c_2 & e_{11} & e_{12} & e_{21} & e_{22} & s_1 & s_0 \end{pmatrix} \quad (27)$$

and the identity permutation are elements of $\text{Aut}(\Gamma_m(M))$ since they preserve the edge list $E$ and vertex coloring. In fact these are the only the 10! possible permutations of $N$ that preserve the edge list $E$ and vertex colors. Using the isomorphism (24) we find that the permutation group of the matrix $M = I$ is

$$\text{Perm}(I) = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\}. \quad (28)$$

Alternative graph constructions $\Gamma_m(M)$ can be found in the literature [6], [16]. Using Procedure 2 we can find the permutation group $\text{Perm}(M)$ for matrices $M \in \mathbb{R}^{m \times m}$ in thousand of dimensions $O(m) = 1000$. This means we can find the symmetries of systems with thousands of constraints.
B. Permutations that Commute with Multiple Matrices

In this section we present the procedure for finding the set of all permutation matrices that commute with multiple matrices $M_1, \ldots, M_r$. This result is needed to calculated $\text{Perm}(\Sigma)$ and $\text{Perm}(\text{MPC})$. For notational compactness we will often denote $M_1, \ldots, M_r$ by $\{M_k\}_{k=1}^r$.

The obvious way to calculate the set of all permutation matrices that commute with multiple matrices $M_1, \ldots, M_r$ is to calculate the groups $\text{Perm}(M_1), \ldots, \text{Perm}(M_r)$ and then compute their intersection. However this method is impractical since the groups $\text{Perm}(M_i)$ can have exponential size in $m$. Calculating the intersection of groups using the generators is non-trivial since the intersection of the group generators $\bigcap_{i=1}^r \text{Gen}(\text{Perm}(M_i))$ does not necessarily generate of the intersection of the groups $\bigcap_{i=1}^r \text{Perm}(M_i)$.

To calculate the group $\text{Perm}(\{M_k\}_{k=1}^r) = \bigcap_{i=1}^r \text{Perm}(M_k)$ we construct a matrix $M$ that captures the symmetry structure of the set of matrices $\{M_k\}_{k=1}^r$. The matrix $M$ has a distinct value $M_{ij}$ for each unique $r$-tuple $(M_{ij1}, \ldots, M_{ijr})$. In other words $M_{ij1} = M_{ij2}$ if and only if $M_{i, j, k} = M_{i, j, k'}$ for $k, k' \in \{1, \ldots, r\}$. The following corollary shows that $\text{Perm}(M) = \bigcap_{i=1}^r \text{Perm}(M_k)$.

**Corollary 2:** Let $M$ satisfy $M_{ij1} = M_{ij2}$ if and only if $M_{i, j, k} = M_{i, j, k'}$ for $k, k' \in \{1, \ldots, r\}$. Then $\text{Perm}(M) = \bigcap_{i=1}^r \text{Perm}(M_k)$.

**Proof:** If $P$ commutes $PM_k = M_kP$ with each matrix $M_k$ for $k = 1, \ldots, r$ then is must commute with $M$ and conversely.

Thus we can use Procedure 2 to find generators of the group $\text{Perm}(\{M_k\}_{k=1}^r) = \bigcap_{i=1}^r \text{Perm}(M_k)$. This is demonstrated in the following example.

**Example 3:** Consider the autonomous system (13) from Example 1. In order to find the symmetry group $\text{Aut}(\Sigma)$ we must find the group $\text{Perm}(\Sigma)$ of all permutation matrices $P$ that commute with the matrices $M_1 = H(H^TH)^{-1}H^T$ and $M_2 = H^TA(H^TH)^{-1}H^T$ where

$$M_1 = \frac{1}{2} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}, \quad M_2 = \frac{1}{2} \begin{bmatrix} 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}. \quad (29)$$

The are four unique pairs $(M_{ij2}, M_{ij2'})$ of values for these two matrices

$$(M_{ij2}, M_{ij2'}) \in \{(\frac{1}{2}, 0), (\frac{1}{2}, 0), (0, \frac{1}{2}), (0, \frac{1}{2})\}. \quad (30)$$

Let $s_1, s_2, s_3$, and $s_4$ be arbitrary distinct values. Define the matrix $M \in \mathbb{R}^{4 \times 4}$ such that

$$M_{ij} = \begin{cases} s_1 & \text{if } (M_{ij2}, M_{ij2'}) = (\frac{1}{2}, 0) \\ s_2 & \text{if } (M_{ij2}, M_{ij2'}) = (0, \frac{1}{2}) \\ s_3 & \text{if } (M_{ij2}, M_{ij2'}) = (-\frac{1}{2}, 0) \\ s_4 & \text{if } (M_{ij2}, M_{ij2'}) = (0, -\frac{1}{2}) \end{cases}. \quad (31)$$

Then we get the matrix

$$M = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \\ s_1 & s_2 & s_3 & s_4 \end{bmatrix}. \quad (32)$$

The group $\text{Perm}(M)$ of permutation matrices that commute with the matrix $M$ is identical to the group $\text{Perm}(M_1, M_2)$ of permutation matrices that commute with $M_1$ and $M_2$.

The matrix $M$ is a circulant $4 \times 4$ matrix. Thus its symmetry group is the cyclic-$4$ group $C_4$. Indeed using graph automorphism software to identify the symmetries of $\Gamma(M)$ we find $\text{Aut}(\Gamma(M)) = \langle (\sigma_m) \rangle$ is generated by the $4$-cycle

$$\sigma_m = \begin{bmatrix} r_1 & r_2 & r_3 & r_4 \\ r_4 & r_1 & r_2 & r_3 \end{bmatrix}. \quad (33)$$

Using isomorphism (24) we find that $\text{Perm}(M) = \text{Perm}(M_1, M_2) = \langle (P) \rangle$ is generated by

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (34)$$

Using isomorphism (11) from $\text{Perm}(\Sigma)$ to $\text{Aut}(\Sigma)$ we find that $\text{Aut}(\Sigma) = \langle (\Theta) \rangle$ is the set of all rotations of the state-space by increments of 90 degrees.

**C. Block Structured Permutation Matrices**

The permutation matrices in the permutation groups $\text{Perm}(X, U)$, $\text{Perm}(\Sigma)$, and $\text{Perm}(\text{MPC})$ have a block structure: The state-space and input-state constraints are permuted independently

$$P = \begin{bmatrix} P_x & 0 \\ 0 & P_u \end{bmatrix} \in \mathbb{R}^{m \times m}. \quad (36)$$

where $P_x \in \mathbb{R}^{m_x \times m_x}$ and $P_u \in \mathbb{R}^{m_u \times m_u}$. We can encode this desired block structure for the permutation matrix $P$ into the graph $\Gamma(M)$. We color the first $\beta = m_x$ nodes of the graph differently then the last $m_u$ nodes. The graph automorphism software only returns permutations $\sigma_m$ that do not switch vertices of different colors. Thus by isomorphism (24) the permutation matrix $P$ will have the desired block structure.

**Procedure 2** Create graph $\Gamma_\beta(M)$ whose symmetry group $\text{Aut}(\Gamma_\beta(M))$ is isomorphic to the permutation group $\text{Perm}_\beta(M)$.

1. **Input:** Square matrix $M$ and block dimension $\beta$
2. **Output:** Vertex colored graph $\Gamma_\beta(M) = (\mathcal{V}, \mathcal{E})$
3. **Construct** the vertex set $\mathcal{V}$:
   a. Add a vertex $v_i$ for $i = 1, \ldots, n$ for each row with color $R_1$ for first $\beta$ rows and $R_2$ for remaining rows.
   b. Add a vertex $c_j$ for $j = 1, \ldots, n$ for each row with color $C_1$ for first $\beta$ rows and $C_2$ for remaining rows.
   c. Add a vertex $e_{ij}$ for $i, j = 1, \ldots, n$ for each element.
   d. Add a vertex $s$ for each unique matrix element $M_{ij}$.
4. **Construct** the edge set $\mathcal{E}$:
   a. Connect each row vertex $v_i$ with the element vertex $e_{ij}$.
   b. Connect each column vertex $c_j$ with the element vertex $e_{ij}$.
   c. Connect each element vertex $e_{ij}$ with value vertex $s$.

V. SYMMETRY IDENTIFICATION SOFTWARE

We have implement our symmetry identification procedure in Matlab. The symmetry identification software requires as an input the dynamics matrices $A$ and $B$, and the sets $\mathcal{X}$ and $\mathcal{U}$ to calculate the symmetry group $\text{Aut}(\Sigma)$ of the constrained dynamics system.
The symmetry identification software calculates the matrices

\[
\begin{align*}
M_1 &= \begin{bmatrix}
H_u (H_u^T P_u) \cdots H_u (H_u^T P_u) & 0 \\
0 & H_u (H_u^T P_u) \cdots H_u (H_u^T P_u)
\end{bmatrix} \quad (37) \\
M_2 &= \begin{bmatrix}
H_u A (H_u^T P_u) \cdots H_u A (H_u^T P_u) & 0 \\
0 & H_u A (H_u^T P_u) \cdots H_u A (H_u^T P_u)
\end{bmatrix}. \quad (38)
\end{align*}
\]

In addition if the optional inputs \( P_u, Q_u, \) and \( R_u \) are included the software will also calculate the matrices

\[
M_3 = \begin{bmatrix}
H_u Q H_u^T & 0 \\
0 & H_u R H_u^T
\end{bmatrix} \quad \text{and} \quad M_4 = \begin{bmatrix}
H_u P H_u^T & 0 \\
0 & H_u R H_u^T
\end{bmatrix}. \quad (39)
\]

The symmetry identification software produces a \( m_x + m_y \) by \( m_x + m_y \) matrix \( M \) with distinct values \( M_{ij} \) for each unique pair \( (M_{ij1}, M_{ij2}) \) or each unique 4-tuple \( (M_{ij1}, M_{ij2}, M_{ij3}, M_{ij4}) \) when the optional inputs \( P_u, Q_u, \) and \( R_u \) are included. The symmetry identification software uses Procedure 2 to produce the graph \( \Gamma_{m_{xy}}(M) \).

The symmetries of the graph \( \Gamma_{m_{xy}}(M) \) are identified using Saucy v3.0 [11]. Saucy can find the symmetry group of graphs with millions of vertices in seconds [11]. For application this means we can analyze constrained dynamics systems with thousands of constraints \( m_x \) and \( m_y \).

The symmetries \( \sigma \in \text{Aut}(\Gamma_{m_{xy}}(M_1, M_2)) \) are transformed into permutation matrices \( P_1 \) and \( P_2 \) using (24) to produce the group \( \text{Perm}(\Sigma) = \text{Perm}_{m_{xy}}(M_1, M_2) \) (respectively \( \text{Perm}(\text{MPC}) = \text{Perm}_{m_{xy}}(M_1, M_2, M_3, M_4) \)).

Finally the symmetry identification software applies isomorphism (11) to produce the symmetry group \( \text{Aut}(\Sigma) \) (respectively \( \text{Aut}(\text{MPC}) \)) of the dynamic system (6) (respectively the model predictive control problem (1)).

### A. Example: Quadcopter

We applied our symmetry identification software to the quadcopter system described in [2] and shown in Figure 2. The quadcopter has 13 states; 6 cartesian position and velocity states, 6 angular position and velocity states, and 1 integrator state to counter-act gravity. The quadcopter has 4 inputs corresponding to the voltage applied to the four motors.

The symmetry identification software found 16 symmetries with 3 generators. The generators of the state-space transformations are

\[
\begin{align*}
\Sigma_1 &= \text{blkdiag} \left\{ I_2 \otimes \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}, I_2 \otimes \begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix} \right\}, \quad (40a) \\
\Sigma_2 &= \text{blkdiag} \left\{ I_2 \otimes \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}, I_2 \otimes \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} \right\}, \quad (40b) \\
\Sigma_3 &= \text{blkdiag} \left\{ I_2 \otimes \begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix}, I_2 \otimes \begin{bmatrix} -1 & +1 \\ +1 & -1 \end{bmatrix} \right\}. \quad (40c)
\end{align*}
\]

where \( \otimes \) is the Kronecker product and \( \text{blkdiag} \) represents a block-diagonal matrix. The generators of the input-space transformations are

\[
\begin{align*}
\Omega_1 &= -\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Omega_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Omega_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (41a)
\end{align*}
\]

The generator pair \( (\Theta_1, \Omega_1) \) flips the quadcopter about the \( x \)-axis. The input-space transformation \( \Omega_1 \) swaps motors 1 and 3 and reverse the direction of thrust. The state-space transformation \( \Theta_1 \) reverses the \( y \)- and \( z \)-axes and reverses rotations about the \( x \)- and \( z \)-axes.

The generator pair \( (\Theta_2, \Omega_2) \) rotates the quadcopter 90 degrees about the \( z \)-axis. The input-space transformation \( \Omega_2 \) maps motor 1 to 2, motor 2 to 3, motor 3 to 4, and motor 4 to 1. The state-space transformation \( \Theta_2 \) maps the \( x \)-axis to the \( y \)-axis and the \( y \)-axis to the negative \( x \)-axis. Rotations about the \( x \)- and \( y \)-axis are swapped and rotations about the \( x \)-axis are reversed.

The generator pair \( (\Theta_3, \Omega_3) \) flip the quadcopter about a diagonal axis. The input-space transformation \( \Omega_3 \) swaps motors 1 and 4, and 2 and 3. The state-space transformation \( \Theta_3 \) swaps and reverses the \( x \)- and \( y \)-axes. Rotations about these axes are also swapped and reversed.

We generated an explicit model predictive control for the quadcopter system using mpt-toolbox [15]. Without exploiting symmetry the explicit MPC had 10, 173 regions and required 53.7 megabytes of storage. By exploiting symmetry the number of region is reduced to 772 regions and 4.2 megabytes of storage. Furthermore the computational complexity of implementing the controller has not be increased [10].

### References


