

Feedback Control with Lie Groups

Jon Woolfrey*

June 10, 2025

Abstract: In this article I extend the concept of linear feedback control for scalars and vectors in to the realm of Lie groups. Lie groups are mathematical objects with generalised properties for combining, inverting, and computing "differences". They are used to represent orientation in 3D space in robotics and animation. By understanding their properties we can apply the same logic as linear systems and solve more sophisticated, nonlinear control problems.

1. Linear Feedback Control

In another article I introduced the procedure for solving feedback control of linear systems using a 3-step process. Given the current position $\mathbf{x} \in \mathbb{R}^m$ and the desired position $\mathbf{x}_d \in \mathbb{R}^m$, we:

1. Denote the error from the desired position:

$$\boldsymbol{\epsilon} = \mathbf{x}_d - \mathbf{x}. \quad (1)$$

2. Evaluate the time derivative:

$$\dot{\boldsymbol{\epsilon}} = \dot{\mathbf{x}}_d - \dot{\mathbf{x}} \quad (2)$$

3. Solve the input to force an exponential decay for the error:

$$\dot{\mathbf{x}} = \dot{\mathbf{x}}_d + \mathbf{K}\boldsymbol{\epsilon} \implies \dot{\boldsymbol{\epsilon}} = -\mathbf{K}\boldsymbol{\epsilon} \implies \boldsymbol{\epsilon}(t) = e^{-\mathbf{K}t} \boldsymbol{\epsilon}_0. \quad (3)$$

where $\mathbf{K} \in \mathbb{R}^{n \times n}$ is a positive definite matrix, such that $-\mathbf{K}$ has negative eigenvalues.

How do we perform feedback control for other types of mathematical structures?

For example, it is common to represent the orientation of a rigid body using a rotation matrix:

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \in \mathbb{R}^{3 \times 3}. \quad (4)$$

Each of the columns are unit norm:

$$r_{1i}^2 + r_{2i}^2 + r_{3i}^2 = 1 \quad \text{for } i \in \{1, 2, 3\} \quad (5)$$

and are orthogonal:

$$r_{1i}r_{1j} + r_{2i}r_{2j} + r_{3i}r_{3j} = 0 \quad \text{for } i, j \in \{1, 2, 3\} \text{ and } i \neq j. \quad (6)$$

So we **cannot** add or subtract these matrices $\mathbf{R}_d - \mathbf{R}$ without violating these properties.

*jonathan.woolfrey@gmail.com

2. Lie Groups

Lie groups are mathematical structures that satisfy 4 properties:

1. Closure: combining 2 objects in the group remains in the group,
2. Associativity: the order in which we cluster operations doesn't matter, as long as the sequence remains the same,
3. Identity: The element in the group that results in no change, and
4. Inverse: The element that leads to the identity.

Vectors (over addition) form a Lie group (Table 1).

Table 1: Lie group properties of vectors over addition.

Closure:	$\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n : \mathbf{x}_1 + \mathbf{x}_2 \in \mathbb{R}^n$
Associativity:	$\mathbf{x}_1 + (\mathbf{x}_2 + \mathbf{x}_3) = (\mathbf{x}_1 + \mathbf{x}_2) + \mathbf{x}_3$
Identity:	$\mathbf{0} \in \mathbb{R}^n : \mathbf{x} + \mathbf{0} = \mathbf{x}$
Inverse:	$-\mathbf{x} : \mathbf{x} + (-\mathbf{x}) = \mathbf{0}$

The closure and inverse property were applied to define the position error, Eq. (1). In fact, we can see that when the desired position is equal to the actual position, this leads to the identity:

$$\mathbf{x}_d = \mathbf{x} \implies \mathbf{x}_d - \mathbf{x} = \mathbf{0}. \quad (7)$$

3. Orientation Control with Rotation Matrices

The rotation matrix, Eq. (4), actually belongs to the Special Orthogonal \mathbb{SO} group:

$$\mathbf{R} \in \mathbb{SO}(n) \triangleq \{\mathbf{R} \in \mathbb{R}^{n \times n} : \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1\}. \quad (8)$$

Importantly, the closure property is defined by matrix multiplication, and inverse by its transpose (Table 2).

Table 2: Properties of the Special Orthogonal group.

Closure:	$\mathbf{R}_1, \mathbf{R}_2 \in \mathbb{SO}(n) : \mathbf{R}_1\mathbf{R}_2 \in \mathbb{SO}(n)$
Associativity:	$\mathbf{R}_1(\mathbf{R}_2\mathbf{R}_3) = (\mathbf{R}_1\mathbf{R}_2)\mathbf{R}_3$
Identity:	$\mathbf{I} \in \mathbb{SO}(n) \subset \mathbb{R}^{n \times n} : \mathbf{R}\mathbf{I} = \mathbf{R}$
Inverse:	$\mathbf{R}^T : \mathbf{R}\mathbf{R}^T = \mathbf{I}$

As with Eq. (1), we first apply the closure and inverse properties of $\mathbb{SO}(n)$ to define the rotation error as:

$$\mathbf{E} = \mathbf{R}_d \mathbf{R}^T. \quad (9)$$

The \mathbb{SO} group can actually be written as a matrix exponential, so we can instead write Eq. (9) as:

$$\mathbf{E} = e^{S(\epsilon)} \quad (10)$$

where:

- $\theta = \|\epsilon\| \in [0, 2\pi]$ is the magnitude of the rotation error (rad), and
- $\hat{\epsilon} = \frac{\epsilon}{\|\epsilon\|} \in \mathbb{R}^3$ is the axis to rotate about,

and

$$S(\epsilon) = \begin{bmatrix} 0 & -\epsilon_z & \epsilon_y \\ \epsilon_z & 0 & -\epsilon_x \\ -\epsilon_y & \epsilon_x & 0 \end{bmatrix} \in \mathfrak{so}(3) \quad (11)$$

is the Lie algebra of $\mathbb{SO}(3)$ (a skew-symmetric matrix).

Second, we evaluate the time derivative which, from Eq. (10), becomes:

$$\dot{\mathbf{E}} = \mathbf{S}(\dot{\epsilon})\mathbf{E}, \quad \dot{\epsilon} = \boldsymbol{\omega}_d - \boldsymbol{\omega}. \quad (12)$$

The time derivative of the Lie algebra is actually the difference between the desired velocity $\boldsymbol{\omega}_d \in \mathbb{R}^3$ (rad/s), and the actual velocity $\boldsymbol{\omega} \in \mathbb{R}^3$ (rad/s). Now instead of operating over $\mathbb{SO}(3)$ or $\mathfrak{so}(3)$, we can apply what we already know about \mathbb{R}^n . If we define the input angular velocity as:

$$\boldsymbol{\omega} \triangleq \boldsymbol{\omega}_d + \mathbf{K}\epsilon \quad (13)$$

for a matrix $\mathbf{K} \in \mathbb{R}^{3 \times 3}$, then the error derivative becomes:

$$\dot{\epsilon} = -\mathbf{K}\epsilon \implies \epsilon = e^{-\mathbf{K}t} \epsilon_0. \quad (14)$$

Likewise, the rotation error will decay to the identity:

$$\lim_{t \rightarrow \infty} \mathbf{E} = e^{-\mathbf{S}(\mathbf{K}\epsilon)t} = \mathbf{I}. \quad (15)$$

Figure 1 shows screenshots of a simulation of the ergoCub robot rotating its hands. I used this principle to control the orientation when holding an object.

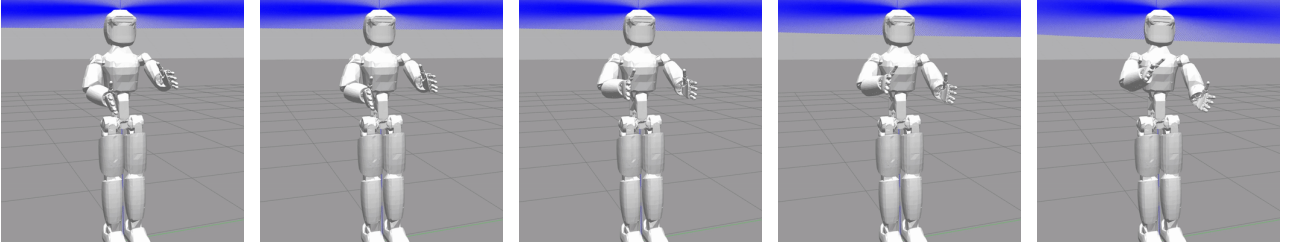


Figure 1: We can control the orientation of robot arms and/or objects using Lie group and Lie algebra properties of the rotation matrix.