Tracking Error in SO(3)

The SO(3) group represents three-dimensional rotations. When implementing feedback control with angular velocities, there are several common methods for defining orientation error:

Axis-Angle Error

Given a current rotation matrix $\mathbf{R}$ and desired rotation matrix $\mathbf{R_d}$, compute the relative rotation:

$$\tilde{\mathbf{R}} = \mathbf{R_d}\mathbf{R}^T$$

The axis-angle error vector can be extracted from $\tilde{\mathbf{R}}$ using the axis-angle representation.

Quaternion Error

For unit quaternions $\mathbf{q}$ (current) and $\mathbf{q_d}$ (desired), compute:

$$\tilde{\mathbf{q}} = \mathbf{q_d} \cdot \mathbf{q}^*$$

The resulting quaternion $\tilde{\mathbf{q}}$ can be converted in different ways to a 3D vector for control purposes.

Logarithmic Map (Matrix Logarithm)

Compute the matrix logarithm of the relative rotation:

$$\tilde{\mathbf{S}} = \log(\mathbf{R_d}\mathbf{R}^T)$$

The resulting skew-symmetric matrix $\tilde{\mathbf{S}}$ can be converted to a vector for control purposes.

Implementation Examples

Using Pinocchio:

```
import pinocchio as pin

# Compute orientation error directly
error_vector = pin.log3(Rd @ R.T)
```

Using Python with SciPy:

```python
import numpy as np
from scipy.linalg import logm

def skew_to_vector(skew_matrix):
    """Extract the vector from a skew-symmetric matrix."""
    return np.array([skew_matrix[2, 1],
                     skew_matrix[0, 2],
                     skew_matrix[1, 0]])

def so3_error(R, Rd):
    """Compute orientation error using matrix logarithm."""
    error_matrix = Rd @ R.T
    error_log = logm(error_matrix)
    error_vector = skew_to_vector(error_log)
    return error_vector
```

The resulting error vector can be used in your control law to command angular velocities, where $\mathbf{R}$ represents the current orientation and $\mathbf{R_d}$ the desired orientation.

Further Reading

For a deeper understanding of SO(3) and control theory, consider these resources:

1. A Mathematical Introduction to Robotic Manipulation by Murray, Li, and Sastry
   - Comprehensive coverage of geometric mechanics and robot control
2. Robotics: Modelling, Planning and Control by Siciliano et al.
   - Excellent treatment of robot kinematics and control
3. Modern Robotics: Mechanics, Planning, and Control by Lynch and Park
   - Modern perspective on geometric mechanics
4. Space Vehicle Dynamics and Control by Bong Wie
   - Detailed coverage of attitude control and SO(3)
5. Stanford's Introduction to Robotics (CS223A)
   - Excellent course materials on robot kinematics and control