Tracking Error in SO(3)

The SO(3) group represents three-dimensional rotations. When defining an error for feedback control with angular velocities, several methods are commonly used:

- **Axis-Angle Error**: If $\mathbf{R}$ is the current rotation matrix and $\mathbf{R_d}$ is the desired rotation matrix, the axis-angle error can be found by:

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

  Convert matrix $\tilde{\mathbf{R}}$ to an axis-angle representation where the resulting vector represents the error.
- **Quaternion Error**: For unit quaternions $\mathbf{q}$ and $\mathbf{q_d}$, the error quaternion is:

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

  Convert $\tilde{\mathbf{q}}$ to a 3-vector for control purposes.
- **Logarithmic Map (Matrix Logarithm)**: The matrix logarithm of the rotation difference can provide a skew-symmetric matrix, which can be converted to a vector using a conversion function.

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

  Where `log` denotes the matrix logarithm. The error vector can be extracted from the skew-symmetric matrix.

Here's an example of computing the logarithmic map error in Python:

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

def skew_to_vector(skew_matrix):
    return np.array([skew_matrix[2, 1], skew_matrix[0, 2], skew_matrix[1, 0]])

def so3_error(R, Rd):
    error_matrix = Rd @ R.T
    error_log = logm(error_matrix)
    error_vector = skew_to_vector(error_log)
    return error_vector

# domega = domega_d + Kp * error_vector + Kd * omega_error
```

This error vector is used in the control law to command angular velocities, where $\mathbf{R}$ is the current orientation, $\mathbf{R_d}$ is the desired orientation.

For further reading on SO(3) and control theory, the following resources can be useful:

1. A Mathematical Introduction to Robotic Manipulation by Richard M. Murray, Zexiang Li, and S. Shankar Sastry.
2. Robotics: Modelling, Planning and Control by Bruno Siciliano et al.
3. Modern Robotics: Mechanics, Planning, and Control by Kevin M. Lynch and Frank C. Park.
4. Space Vehicle Dynamics and Control by Bong Wie.
5. Stanford University's Introduction to Robotics Course (CS223A).