Cheat Sheet for Screw Theory and Rigid Body Transformations

1 Screw Axis

1. **Screw Axis** S:
$$S = \begin{pmatrix} \omega_s \\ v_s \end{pmatrix}$$
Explanation: Represents the screw axis with ω_s as angular component and v_s as linear component.

2. **Skew-Symmetric Matrix** $[\omega_s]$ **from Angular component Vector** ω_s:
$$[\omega_s] = \begin{bmatrix} 0 & -\omega_{sz} & \omega_{sy} \\ \omega_{sz} & 0 & -\omega_{sx} \\ -\omega_{sy} & \omega_{sx} & 0 \end{bmatrix}$$
Explanation: The skew-symmetric matrix $[\omega_s]$ is used to represent the cross product operation with the angular component vector ω_s.

3. **Screw Axis** $[S]$:
$$[S] = \begin{bmatrix} [\omega_s] & v_s \\ 0 & 0 \end{bmatrix}$$
Explanation: The matrix $[S]$ represents the screw axis of a rigid body, combining both its angular component ω_s in a skew-symmetric matrix form and its linear component v_s as a column vector. The bracket notation [S] does not imply that S is a skew-symmetric matrix.

4. **Homogeneous Transformation Matrix** T:
$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$$
Explanation: Describes the pose with R as rotation matrix and p as translation vector.

5. **Exponential Map for Rotation (Rodrigues' Rotation Formula)**:
$$R = e^{[\omega]\theta} = I + [\omega]\sin(\theta) + [\omega]^2(1 - \cos(\theta))$$
Explanation: Computes rotation matrix from angular component ω and rotation angle θ.

6. **Function $G(\theta)$ for Translation Component**:

   - If there is rotation and translation from screw axis S, the general form:
     $$G(\theta) = I\theta + (1 - \cos\theta)[\omega_s] + (\theta - \sin\theta)[\omega_s]^2$$
     Explanation: Computes $G(\theta)$ for rotation and translation from screw axis S and motion parameter θ.
   - if pure translation:
     $$G(\theta) = d$$
     Explanation: Case for pure translation with d as distance and v_s as direction.

7. **Exponential Map for Rigid Body Motions**:

   $$T = e^{[S]\theta} = \begin{bmatrix} R & G(\theta)v_s \\ 0 & 1 \end{bmatrix}$$

   Explanation: Computes transformation for screw axis S and motion.

8. **Matrix Logarithm to Find the Screw Axis**:

   $$[S]q = \begin{bmatrix} [\omega_s]q & v_s q \\ 0 & 0 \end{bmatrix} = \log(T) \in se(3)$$

   Explanation: Computes the screw axis from transformation matrix T.

   where:

   - ω_s is the angular component vector, which defines the axis of rotation.
   - v_s is the linear component vector, which defines the direction and magnitude of translation.
   - q is the scalar that scales the screw axis to achieve the transformation from the initial to the final configuration of the rigid body.

9. **Calculate ω_s :**

   (a) If $R = I$ (the identity matrix), it indicates that there is no rotation, and hence $\omega_s = 0$.

   (b) If $R \neq I$, there is a rotational component. Use the matrix logarithm on $SO(3)$ (the rotation part of T) to find ω_s and the rotation angle θ.

   - The rotation angle θ can be calculated using the formula:
     $$\theta = \arccos\left(\frac{\text{trace}(R) - 1}{2}\right)$$
     where trace(R) is the sum of the diagonal elements of R.

- Calculation of ω_s The angular part ω_s, representing the angular component vector, can be calculated differently based on whether the rotation angle θ is an integer multiple of π:
  - Normal Case For the normal case where $\theta \neq n\pi$, ω_s is calculated using the formula:

$$\omega_s = \frac{R - R^T}{2\sin(\theta)} = \frac{1}{2\sin(\theta)} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}$$

  - Special Case In the special case where $\theta = n\pi$ and n is an odd integer, $\sin(\theta)$ equals zero and the above formula for ω_s does not apply. Instead, the rotation matrix R simplifies to $R = I + 2[\omega_s]^2$, and we have:

$$[\omega_s]^2 = \frac{1}{2}(R - I)$$

    The squared form of the skew-symmetric matrix $[\omega_s]^2$ will be:

$$[\omega_s]^2 = \begin{bmatrix} -\omega_{sy}^2 - \omega_{sz}^2 & \omega_{sx}\omega_{sy} & \omega_{sx}\omega_{sz} \\ \omega_{sy}\omega_{sx} & -\omega_{sx}^2 - \omega_{sz}^2 & \omega_{sy}\omega_{sz} \\ \omega_{sz}\omega_{sx} & \omega_{sz}\omega_{sy} & -\omega_{sx}^2 - \omega_{sy}^2 \end{bmatrix}$$

    To find the elements ω_{sx}, ω_{sy}, and ω_{sz} of the skew-symmetric matrix $[\omega_s]$ from its squared form, we use the following relationships based on the diagonal and off-diagonal elements:

    For the diagonal elements of $[\omega_s]^2$:

    i. $-\omega_{sy}^2 - \omega_{sz}^2 = \frac{1}{2}(r_{11} - 1)$
    ii. $-\omega_{sx}^2 - \omega_{sz}^2 = \frac{1}{2}(r_{22} - 1)$
    iii. $-\omega_{sx}^2 - \omega_{sy}^2 = \frac{1}{2}(r_{33} - 1)$

    And for the off-diagonal elements:

    i. $\omega_{sx}\omega_{sy} = \frac{1}{2}r_{12}$ (and similar equations for other off-diagonal elements).

10. **Calculate** v_s :

- The inverse of the function $G(\theta)$, denoted as $G^{-1}(\theta)$, is given by:
  - If there is rotation and translation from screw axis S, the general form:

$$G^{-1}(\theta) = \frac{1}{\theta}I - \frac{1}{2}[\omega_s] + \left(\frac{1}{\theta} - \frac{1}{2}\cot\left(\frac{\theta}{2}\right)\right)[\omega_s]^2$$

- If there is pure translation:

$$G^{-1}(\theta) = \frac{1}{d}$$

- v_s formula:

$$v_s = G^{-1}(\theta)p$$

Explanation: Derives linear component v_s by "removing" rotational effect from p.

where:

- p is the translation vector in T.

11. **Calculate The pitch of a screw (h)**:

$$h = \frac{\mathbf{v_s} \cdot \omega_\mathbf{s}}{\|\omega_\mathbf{s}\|^2}$$

2 POE

The transformation matrix representing the pose of the end-effector is given by:

$$T(q) = e^{[S_1]q_1} e^{[S_2]q_2} \cdots e^{[S_n]q_n} M \tag{1}$$

where:

- $T(q)$ is the transformation matrix representing the pose of the end-effector.
- $e^{[S_i]q_i}$ is the matrix exponential of the screw axis S_i for the i^{th} joint, multiplied by the joint variable q_i.
- M is the home configuration matrix of the end-effector.
- $S_1, S_2, \ldots, S_n$ are the screw axes for each joint.
- $q_1, q_2, \ldots, q_n$ are the joint variables, which can be angles for revolute joints or displacements for prismatic joints.

1. Translational Component of the Screw Axis - (Resolute joint)

   The translational component of the screw axis, denoted as S_v, is a crucial element in describing the motion of a joint in robotic kinematics, particularly for revolute joints. It is defined as the cross product of the rotational axis vector S_ω and the position vector a of a point on the joint axis.

   For a revolute joint rotating about the z-axis and centered at the origin, S_v is given by:

$$S_v = a \times S_\omega = \begin{bmatrix} a_y S_{\omega_z} - a_z S_{\omega_y} \\ a_z S_{\omega_x} - a_x S_{\omega_z} \\ a_x S_{\omega_y} - a_y S_{\omega_x} \end{bmatrix}$$

For a joint at the origin, where $a = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, and S_ω is aligned with the z-axis, S_v simplifies to:

$$S_v = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

In this case, the translational component is zero because there is no linear component induced by the rotation of the joint.

2. Screw Axis Components (Prismatic Joint)

   In the analysis of robotic mechanisms, particularly for prismatic joints, the screw axis plays a vital role in characterizing joint movement. Unlike revolute joints, prismatic joints exhibit translation along their axis without rotation. This characteristic defines their screw axis components as follows:

   (a) Rotational Component S_ω For a prismatic joint, the rotational component is non-existent since there is no rotation about any axis. This leads to the rotational component being a zero vector:

   $$S_\omega = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

   (b) Translational Component S_v The translational component S_v represents the direction of the prismatic joint's translation. If the joint translates along the z-axis, for example, the translational component would be:

   $$S_v = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

   This vector indicates a unit translation along the z-axis, consistent with the motion of the prismatic joint.

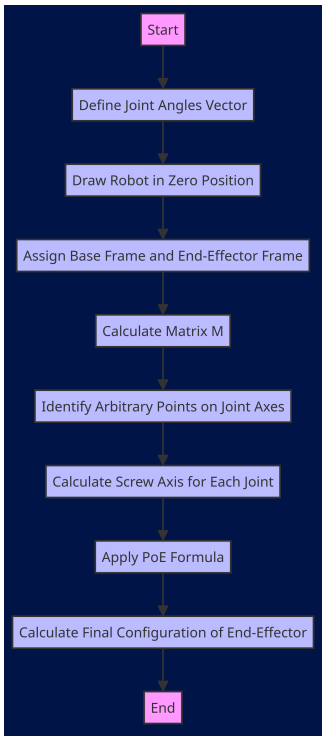

Figure 1: Flowchart illustrating the Product of Exponentials method in robotic kinematics.

3 Twists in Robotics

In robotics, a twist is defined as a continuous change in pose over time. A twist combines both the angular and linear velocities of a moving body into a compact 6-vector. This representation is crucial for understanding and controlling the motion of robots and mechanical systems.

- **Representation of Twist**:

$$\mathcal{V} = \begin{pmatrix} \omega \\ v \end{pmatrix}_{6\times 1} \in \mathbb{R}^6$$

  Explanation: A twist is represented as a 6-vector, combining angular velocity ω and linear velocity v, essential for describing the motion of a body in space.

- **Body Frame Configuration**:

$$T_{sb}(t) = T(t) = \begin{pmatrix} R(t) & p(t) \\ 0 & 1 \end{pmatrix}$$

  Explanation: The configuration of the body frame relative to the space frame is described by the homogeneous transformation matrix T, which includes the rotation matrix R and the position vector p.

- **Twist in Body Frame**:

$$[\mathcal{V}_b] = T^{-1}\dot{T} = \begin{pmatrix} R^T & -R^T p \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{R} & \dot{p} \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} R^T\dot{R} & R^T\dot{p} \\ 0 & 0 \end{pmatrix}$$

$$[\mathcal{V}_b] = \begin{pmatrix} [\omega_b] & v_b \\ 0 & 0 \end{pmatrix} \in se(3)$$

  Explanation: The matrix $[\mathcal{V}_b]$ represents the twist in the body frame, comprising the angular velocity ω_b and the linear velocity v_b.

- **Twist in Space Frame**:

$$[\mathcal{V}_s] = \dot{T}T^{-1} = \begin{pmatrix} \dot{R}R^T & \dot{p} - \dot{R}R^T p \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} [\omega_s] & v_s \\ 0 & 0 \end{pmatrix}$$

  Explanation: This formula calculates the angular velocity in the space frame, with $[\omega_s]$ representing the skew-symmetric matrix of angular velocity ω_s and v_s being the linear velocity in the space frame.

- **Twist as Screw Motion**:

$$\mathcal{V} = \mathcal{S}\dot{q}$$

  Explanation: A twist can also be viewed as screw motion, represented by a screw axis $\mathcal{S}$ and a rate of motion $\dot{q}$, integrating both rotational and translational aspects of motion.

- **Angular velocity in the Space Frame (ω_s)**

  The angular velocity of a rotating body in the space frame is given by the equation $[\omega_s] = \dot{R}R^T$.

  - Here, $\dot{R}$ (or $\dot{R_{sb}}$) represents the time derivative of the rotation matrix R (or R_{sb}).
  - R^T (or R_{sb}^T) is the transpose of R (or R_{sb}).
  - The product $\dot{R}R^T$ (or $\dot{R_{sb}}R_{sb}^T$) isolates the rotational velocity, yielding the angular velocity ω_s in the space frame.
  - $[\omega_s]$ is the skew-symmetric matrix form of the angular velocity vector ω_s, facilitating cross product operations in matrix form.

- **Angular velocity in the Body Frame (ω_b)**

  The angular velocity of a rotating body in the body frame is described by the equation $[\omega_b] = R^T\dot{R}$.

  - This equation views the rotational motion from the perspective of the moving body.
  - $R^T\dot{R}$ (or $R_{sb}^T\dot{R_{sb}}$) calculates the angular velocity ω_b in the body frame.
  - $[\omega_b]$ is the skew-symmetric matrix form of the angular velocity vector ω_b in the body frame.

- **Linear velocity Transformation**

  When considering linear velocities, the origin of the frames becomes important, particularly if they are not coincident.

  - In a scenario where the origins of the space and body frames are different, the linear velocity v_s in the space frame and v_b in the body frame are related by the equation:

    $$v_b = R_{sb}^T(v_s - \omega_s \times p) = R_{sb}^T\dot{p}$$

    where
  - p is the position vector of the body frame's origin relative to the space frame's origin.
  - $\dot{p}$ represents the time derivative of the position vector p, which is the rate of change of the position of the body frame's origin with respect to the space frame. It effectively represents the translational velocity of the body frame's origin in the space frame.
  - This equation accounts for the translational offset between the frames, with $\omega_s \times p$ representing the additional linear velocity induced by the frame's rotation around a point other than its origin.

These concepts are fundamental for analyzing and controlling the motion of rotating bodies in robotics and physics, allowing for a precise understanding and manipulation of their movements.

4 Space Jacobian Calculation

The Space Jacobian, denoted as $J_s(q)$, is a matrix that maps joint velocities to the spatial velocity of the end-effector in robotics. It is particularly important for understanding and controlling the movement of robotic manipulators.

4.1 Screw Axes

The first step is to define the screw axes for each joint in the space (fixed) frame. The screw axis for joint i is represented as $\mathcal{S}_i$, and it is a 6-dimensional vector:

$$\mathcal{S}_i = \begin{bmatrix} \omega_i \\ v_i \end{bmatrix}$$

where ω_i represents the angular velocity component and v_i represents the linear velocity component of the screw axis.

4.2 Forward Kinematics

The forward kinematics of the robot is expressed using the product of exponential formula, which leads to the transformation matrix T representing the pose of the end-effector:

$$T(q_1, \ldots, q_n) = e^{[\mathcal{S}_1]q_1} \ldots e^{[\mathcal{S}_n]q_n} M$$

Here, M is the home configuration matrix of the end-effector, and q_i represents the joint variables.

4.3 Adjoint Transformation

The adjoint transformation is used to adjust the screw axes considering the motion of all preceding joints. For the i-th joint, the adjoint transformation of the cumulative product of exponentials of the previous joints is calculated as:

$$Ad_{e^{[\mathcal{S}_1]q_1} \ldots e^{[\mathcal{S}_{i-1}]q_{i-1}}}$$

4.4 Constructing the Jacobian

Each column J_{si} of the Jacobian is then determined by applying the adjoint transformation to the corresponding screw axis. For the first joint, $J_{s1} = \mathcal{S}_1$. For subsequent joints:

$$J_{si} = Ad_{e^{[\mathcal{S}_1]q_1} \ldots e^{[\mathcal{S}_{i-1}]q_{i-1}}} \mathcal{S}_i$$

4.5 Space Jacobian Matrix

Finally, the Space Jacobian matrix is assembled by stacking these columns together:

$$J_s(q) = \begin{bmatrix} J_{s1} & J_{s2} & \dots & J_{sn} \end{bmatrix}$$

4.6 Relating Joint Velocities to Spatial Velocity

The Space Jacobian relates the joint velocities $\dot{q}$ to the spatial velocity of the end-effector $\mathcal{V}_s$:

$$\mathcal{V}_s = J_s(q)\dot{q}$$

This equation is fundamental in robotic control, as it allows for the calculation of necessary joint velocities to achieve a desired end-effector motion.

4.7 Singularity of the Jacobian

Singularity in the context of Jacobian matrices refers to configurations where the Jacobian loses rank and the manipulator loses one or more degrees of freedom. This can occur when the determinant of the Jacobian matrix becomes zero. In such situations, the end-effector may be unable to move in certain directions regardless of joint velocities. Singularity is mathematically expressed as:

$$\det(J_s(q)) = 0$$

Understanding singularities is crucial for avoiding them during motion planning or for taking special actions when they are encountered.

4.8 Rank of the Jacobian

The rank of the Jacobian matrix indicates the number of independent rows or columns in the matrix, which corresponds to the number of independent directions in which the end-effector can move. The rank can be determined using linear algebra techniques. A full-rank Jacobian implies that the end-effector can achieve motion in all directions. It is defined as:

$$\text{Rank}(J_s(q))$$

For a robot with n joints, the maximum rank of the Jacobian is n, and a rank less than n indicates a loss of mobility.

4.9 Determinant of the Jacobian

The determinant of the Jacobian matrix is a scalar value that provides important information about the kinematic behavior of the robotic manipulator. It is

particularly useful for identifying singular configurations. The determinant is calculated as:

$$\det(J_s(q))$$

A zero determinant indicates a singular configuration, while a non-zero determinant implies that the manipulator is away from singularities.

4.10 Analyzing and Avoiding Singularities

In robotic control and motion planning, it is important to analyze the Jacobian matrix to identify and avoid singular configurations. Techniques such as dexterity measures and redundancy resolution can be employed to ensure smooth and efficient motion of the robotic manipulator, especially near singular configurations.

4.11 Calculating Joint Velocities from End-Effector Twist

Given the spatial velocity (or twist) of the end-effector, $\mathcal{V}_s$, and the Space Jacobian matrix $J_s(q)$, the joint velocities $\dot{q}$ can be calculated. This is particularly useful in inverse kinematics where the desired motion of the end-effector is known, and the corresponding joint motions need to be determined.

$$\mathcal{V}_s = J_s(q)\dot{q}$$

To find $\dot{q}$, the equation can be rearranged as:

$$\dot{q} = J_s(q)^{-1}\mathcal{V}_s$$

However, this direct inversion is only feasible if $J_s(q)$ is square and non-singular (i.e., has a non-zero determinant and full rank). In cases where the Jacobian is not square or is singular, alternative methods such as pseudo-inverse or damped least squares can be used.

4.12 Pseudo-Inverse for Non-Square Jacobians

When the Jacobian is not square, the Moore-Penrose pseudo-inverse $J_s(q)^+$ is used to find an approximate solution:

$$\dot{q} = J_s(q)^+\mathcal{V}_s$$

The pseudo-inverse provides the least squares solution to the problem, minimizing the error in the twist approximation.

4.13 Handling Singularities

In configurations where the Jacobian is singular, direct inversion or even pseudo-inverse may lead to unstable or unbounded solutions. To mitigate this, methods like the damped least squares (or Levenberg-Marquardt algorithm) can be employed:

$$\dot{q} = J_s(q)^T \left(J_s(q) J_s(q)^T + \lambda^2 I \right)^{-1} \mathcal{V}_s$$

where λ is a small damping factor that helps to regularize the solution near singular configurations.

4.14 Mathematical Considerations

It's important to note that these methods provide mathematical solutions which may not always be physically feasible due to joint limits, obstacles, or other practical constraints. Therefore, the calculated joint velocities $\dot{q}$ should always be evaluated in the context of the robot's capabilities and environment.

5 Transformation

5.1 Wrench Representation

A wrench is a tool used in robotics to represent the combined effect of forces and torques acting on a body. It is usually expressed as a 6-vector, which contains both the force vector $\mathbf{f}$ and the torque (or moment) vector $\boldsymbol{\tau}$:

$$\text{Wrench} = \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix}$$

This representation is essential for understanding and controlling the motion and interaction of robotic components with their environment.

5.2 Adjoint Transformation Matrix

The adjoint transformation matrix is used when changing the frame of reference for screw axes:

$$[\text{Ad}]_{T_{sb}} = \begin{bmatrix} R & \mathbf{0} \\ [\mathbf{p}]R & R \end{bmatrix}$$

where R is the rotation matrix and $[\mathbf{p}]R$ is the skew-symmetric matrix formed from the position vector $\mathbf{p}$ when transformed by the rotation R. This matrix is crucial for transforming the coordinates of vectors between different frames of reference in space.

5.3 Inverse Adjoint Transformation Matrix

Similarly, the inverse adjoint transformation matrix is used for transforming screw axis from one coordinate frame back to another:

$$[\mathrm{Ad}]_{T_{bs}} = \begin{bmatrix} R^T & \mathbf{0} \\ -R^T[\mathbf{p}] & R^T \end{bmatrix}$$

Here, R^T denotes the transpose of the rotation matrix, which is equivalent to the inverse rotation when R is orthogonal.

5.4 Transformation of Screw Axes, Twist and Wrenches

For transforming screw axes from frame a to frame b, we use:

$$\mathbf{s}_b = [\mathrm{Ad}]_{T_{bs}} \mathbf{s}_s$$

This equation applies the adjoint transformation to the screw axis to update its frame of reference. The same principle is used for changing the frame of reference for Twists. Jacobian and wrenches:

$$\mathrm{Twist}_b = [\mathrm{Ad}]_{T_{bs}} \mathrm{Twist}_s$$

$$\mathrm{Wrench}_b = [\mathrm{Ad}]_{T_{bs}} \mathrm{Wrench}_s$$

$$Jacobian_b(q) = [\mathrm{Ad}_{T_{bs}}]\, Jacobian_s(q)$$

These transformations ensure that the physical quantities representing motion and forces remain consistent across different coordinate frames, which is fundamental for robotic manipulation and control.

6 Inverse Kinematics Algorithm

The inverse kinematics algorithm computes the joint angles needed to achieve a desired end-effector position and orientation. This process uses an iterative numerical method, such as the Newton-Raphson technique, for optimization. The following outlines the algorithm steps:

1. **Initialization:** Begin with an initial guess for the joint variables q_0.
2. **Forward Kinematics:** Compute the current end-effector configuration $T_{sb}(q)$ using the forward kinematics equation:

$$T_{sb}(q) = \prod_{i=1}^{n} e^{[S_i]q_i} M$$

where $e^{[S_i]q_i}$ represents the matrix exponential of the screw axis S_i scaled by the joint variable q_i, and M is the end-effector configuration in the home position.

3. **Error Computation:**

   (a) Calculate the error transformation matrix T_{bd} as the product of the inverse of the current end-effector configuration $T_{sb}(q)$ and the desired configuration T_{sd}:

   $$T_{bd} = T_{sb}(q)^{-1} T_{sd}$$

   (b) Determine the body twist V_{bd} by computing the matrix logarithm of T_{bd}, which represents the required twist to align $T_{sb}(q)$ with T_{sd}:

   $$V_{bd} = \log(T_{bd})$$

4. **Constructing the space Jacobian:** Each column J_{si} of the Jacobian is computed by applying the adjoint transformation to the corresponding screw axis. For the first joint, $J_{s1} = S_1$. For subsequent joints, J_{si} is found using:

   $$J_{si} = \mathrm{Ad}_{e^{[S_1]q_1 \ldots e^{[S_{i-1}]q_{i-1}}}} S_i$$

   The Space Jacobian matrix $J_s(q)$ is then assembled by stacking the columns J_{si} together:

   $$J_s(q) = \begin{bmatrix} J_{s1} & J_{s2} & \ldots & J_{sn} \end{bmatrix}$$

5. **Convert to body Jacobian** Compute the body Jacobian J_b, which relates the twist V_{bd} to changes in joint variables Δq. The body Jacobian is derived by transforming the space Jacobian to the body frame using the adjoint representation of $T_{bs}(q)$:

   $$J_b = [\mathrm{Ad}_{T_{bs}}] J_s$$

   where $[\mathrm{Ad}_{T_{bs}}]$ is the adjoint transformation of $T_{bs}(q)$, and J_s is the space Jacobian.

6. **Joint Variable Update:**

   (a) Calculate the change in joint variables Δq that minimizes the error using the pseudoinverse of the body Jacobian $J_b^\dagger$:

   $$\Delta q = J_b^\dagger V_{bd}$$

   (b) Update the joint variables:

   $$q_{\text{new}} = q + \Delta q$$

7. **Convergence Check:** Evaluate whether the norm of V_{bd} is below a predetermined tolerance to determine if the solution has sufficiently converged to the desired configuration. If not, update q with q_{new} and repeat the iteration.

8. **Termination:** The algorithm concludes when the joint variables reach a solution within the tolerance or after a maximum number of iterations without convergence.

This methodical procedure refines the joint variables until the end-effector configuration aligns with the desired pose to a satisfactory degree of precision. The pseudoinverse of the Jacobian is employed to manage situations where the Jacobian is non-square or singular.

7 Author's Signature

Muhammed Elyamani

LinkedIn: Muhammed Elyamani's LinkedIn
Email: melya038@uottawa.ca