# An SDP Optimization Formulation for the Inverse Kinematics Problem


**Liangting Wu** tomwu@bu.edu

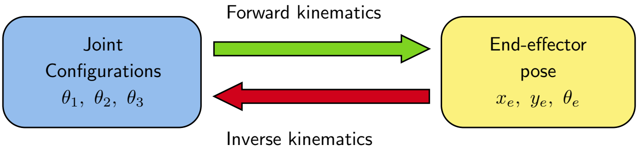
**Roberto Tron** tron@bu.edu

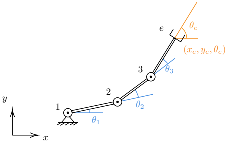

12/14/2023

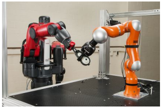
BOSTON UNIVERSITY

# Introduction

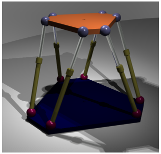
# Inverse Kinematics (IK)



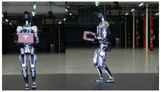



**(a)** Multi-arm cooperation



**(b)** Parallel robot



**(c)** Humanoids

# Inverse Kinematics: previous work

- Analytical solutions
  - Finite number of solutions to arms up to 6 DoFs exists. Solver: IKFast[1].
- Numerical solutions
  - Jacobian inverse technique[2].
  - Heuristic methods. Solvers: CCD[3], FABRIK[4]
  - Nonlinear programming[5][6]

---

[1] R. Diankov (2010), Automated construction of robotic manipulation programs.

[2] S. R. Buss, *IEEE Journal of Robotics and Automation* **17**, 16 (2004).

[3] B. Kenwright, *Journal of Graphics Tools* **16**, 177–217 (2012).

[4] A. Aristidou, J. Lasenby, *Graphical Models* **73**, 243–260 (2011).

[5] T. Le Naour *et al.*, *Computers & Graphics* **84**, 13–23 (2019).

[6] P. Beeson, B. Ames, *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 928–935 (2015).

# Inverse Kinematics: previous work

- Relaxations
  - Mixed-integer programming (MIP)[7]
  - Riemannian optimization[8]
  - Semidefinite programming (SDP)[9,10]

[7] H. Dai *et al.*, *The International Journal of Robotics Research* **38**, 1420–1441 (2019).

[8] F. Marić *et al.*, *IEEE Transactions on Robotics* **38**, 1703–1722 (2021).

[9] T. Yenamandra *et al.*, presented at the 2019 International Conference on 3D Vision (3DV), pp. 318–327.

[10] M. Giamou *et al.*, *IEEE Robotics and Automation Letters* **7**, 1952–1959 (2022).

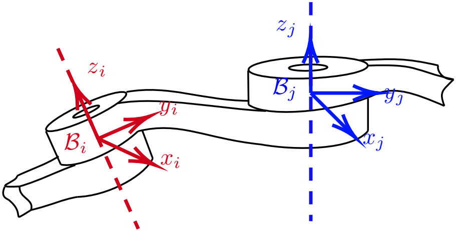
# Contributions

- Single formulation that can incorporate more general kinematic constraints and arrangements of links
  - E.g., parallel robots, multi-robots, humanoids.
- Novel SDP relaxation
  - Novel low-rank projection based on fixed-trace matrices
  - Infeasibility certificates

# Methods

# Parameterization

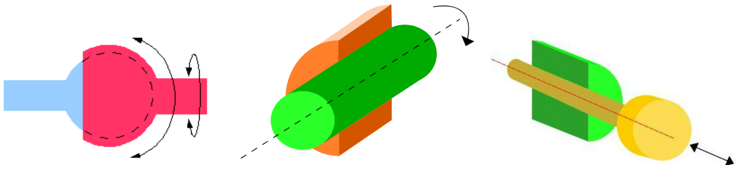
$G = (\mathcal{V}, \mathcal{E})$ := a graph to represent the robot kinematic chain
$\mathcal{V}$ := indices of the links
$\mathcal{E}$ := connections (joints) among the links.




$\mathcal{B}_i := (\mathbf{T}_i, \mathbf{R}_i) \quad \mathcal{B}_j := (\mathbf{T}_j, \mathbf{R}_j)$

Novelty: we parameterize with absolute poses, i.e., $\mathbf{R}_i = {}^{\mathcal{W}}\mathbf{R}_{\mathcal{B}_i}$ and $\mathbf{T}_i = {}^{\mathcal{W}}\mathbf{T}_{\mathcal{B}_i}$.

# Kinematic Constraints on joints



**(a)** Spherical joint **(b)** Revolute joint **(c)** Prismatic joint

💡 We can write the above constraints as matrix linear/semidefinite equalities/inequalities on rotations $\{\mathbf{R}_i\}$.

# Relaxation of the Feasible Set

By dropping the fact that $\mathbf{R}_i \in \mathbf{SO}(3)$, we can develop constraints linear to the (vectorized) free rotations $\mathbf{u} = \mathrm{stack}(\{\mathrm{vec}(\mathbf{R}_i)\}_{i\in\mathcal{V}})$.

$$\left.\begin{array}{r}\text{revolute joint common axis}\\ \text{revolute joint angle limit}\\ \dots\end{array}\right\} \Rightarrow \mathbf{A}_{\mathrm{k}}\mathbf{u} = (\leq)\mathbf{b}_{\mathrm{k}} \tag{1}$$

For $(i, j) \in \mathcal{E}$, the following relation holds:

$$\mathbf{T}_j - \mathbf{T}_i = \mathbf{R}_i {}^i\mathbf{T}_j. \tag{2}$$

We can represent $\{\mathbf{T}_i\}$ with $\{\mathbf{R}_i\}$.

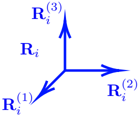
## Relaxation of $\mathrm{SO}(3)$

We want $\mathbf{R}_i \in \mathbf{SO}(3)$, i.e.,

$$\mathbf{R}_i^\mathsf{T}\mathbf{R}_i = \mathbf{I}_3 \text{ and } \det(\mathbf{R}_i) = +1. \tag{3}$$

We propose a **novel** way to relax $\mathbf{SO}(3)$ using convex constraints.

$$\begin{cases} \|\mathbf{R}_i^{(1)}\| = 1 \\ \|\mathbf{R}_i^{(2)}\| = 1 \\ \mathbf{R}_i^{(1)} \cdot \mathbf{R}_i^{(2)} = 0 \\ \mathbf{R}_i^{(1)} \times \mathbf{R}_i^{(2)} = \mathbf{R}_i^{(3)} \end{cases} \Rightarrow \quad \mathbf{Y}_i = \begin{bmatrix} \mathbf{R}_i^{(1)} \\ \mathbf{R}_i^{(2)} \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_i^{(1)} \\ \mathbf{R}_i^{(2)} \\ 1 \end{bmatrix}^\mathsf{T} = \begin{bmatrix} \mathbf{R}_i^{(1)}(\mathbf{R}_i^{(1)})^\mathsf{T} & \mathbf{R}_i^{(1)}(\mathbf{R}_i^{(2)})^\mathsf{T} & \mathbf{R}_i^{(1)} \\ * & \mathbf{R}_i^{(2)}(\mathbf{R}_i^{(2)})^\mathsf{T} & \mathbf{R}_i^{(2)} \\ * & * & 1 \end{bmatrix} \in \mathbb{S}_+^7. \tag{4}$$




"$*$'s" stand for symmetric counterparts

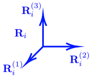
# Relaxation of $\mathrm{SO}(3)$

$\mathbf{R}$:
rotation matrices

$\in$

$\mathcal{U}$:
$$\begin{cases} \text{Kinematic constraints} \\ \mathbf{R}_i \in \mathbf{SO}(3) \end{cases}$$

Original



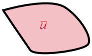

Relaxation
$\Rightarrow$

$\mathbf{Y}$:
decision variables

$\in$

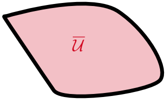
$\bar{\mathcal{U}}$
$$\begin{cases} \text{Kinematic constraints} \\ \mathbf{SO}(3) \end{cases}$$

Lifted



# Relaxation of $\mathrm{SO}(3)$

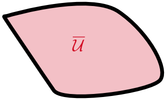
**Proposition 1**
*The set $\bar{\mathcal{U}}$ is compact.*



# Relaxation of $\mathrm{SO}(3)$

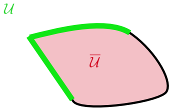
**Proposition 1**

*The set $\bar{\mathcal{U}}$ is compact.*



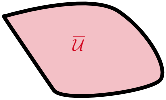

**Proposition 2**

*The relaxed set $\bar{\mathcal{U}}$ contains every element of $\mathcal{U}$, i.e., $\mathcal{U} \subset \bar{\mathcal{U}}$. Moreover, $\mathcal{U}$ is a subset of the boundary of $\bar{\mathcal{U}}$, i.e., $\mathcal{U} \subset \partial\bar{\mathcal{U}}$.*

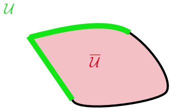


# Relaxation of $\mathrm{SO}(3)$

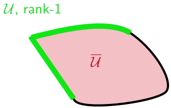
**Proposition 1**
*The set $\bar{\mathcal{U}}$ is compact.*




**Proposition 2**
*The relaxed set $\bar{\mathcal{U}}$ contains every element of $\mathcal{U}$, i.e., $\mathcal{U} \subset \bar{\mathcal{U}}$. Moreover, $\mathcal{U}$ is a subset of the boundary of $\bar{\mathcal{U}}$, i.e., $\mathcal{U} \subset \partial\bar{\mathcal{U}}$.*



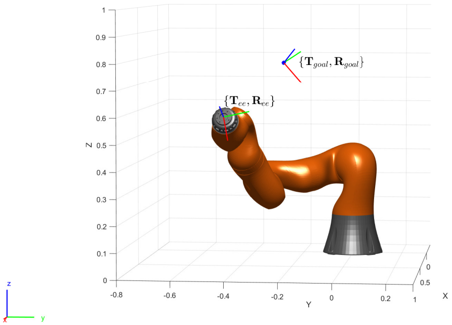

**Proposition 3**
*The set $\mathcal{U}$ is the intersection of $\bar{\mathcal{U}}$ rank-1 matrices.*



# Inverse Kinematics as an Optimization Problem

The inverse kinematics problem aims to find a solution such that the end-effector, $ee$, reaches a desired location $\mathbf{T}_{goal}$ with an orientation $\mathbf{R}_{goal}$. This can be encoded as the cost

$$f(\mathbf{u}) = \|\mathrm{vec}(\mathbf{R}_{ee}) - \mathrm{vec}(\mathbf{R}_{goal})\|_2^2 + \|\mathbf{T}_{ee} - \mathbf{T}_{goal}\|_2^2, \tag{5}$$



# Inverse Kinematics as an Optimization Problem

$\mathbf{R}$:
rotation
matrices

$$\min_{\mathbf{u}} \quad f(\mathbf{u}) \tag{6a}$$

$$\text{subject to} \quad \mathbf{u} \in \mathcal{U} \tag{6b}$$

# Inverse Kinematics as an Optimization Problem

$\mathbf{R}$:
rotation matrices

$$\begin{aligned} \min_{\mathbf{u}} \quad & f(\mathbf{u}) && \text{(6a)} \\ \text{subject to} \quad & \mathbf{u} \in \mathcal{U} && \text{(6b)} \end{aligned}$$

$\Updownarrow$ $\Updownarrow$

# Inverse Kinematics as an Optimization Problem

**R**: rotation matrices

$\Updownarrow$

**Y**: decision variables

$$
\begin{aligned}
\min_{\mathbf{u}} \quad & f(\mathbf{u}) & \text{(6a)} \\
\text{subject to} \quad & \mathbf{u} \in \mathcal{U} & \text{(6b)}
\end{aligned}
$$

$\Updownarrow$

$$
\begin{aligned}
\min_{\mathbf{Y}} \quad & f(\mathbf{Y}) & \text{(7a)} \\
\text{subject to} \quad & \mathbf{Y} \in \bar{\mathcal{U}} & \text{(7b)} \\
& \operatorname{rank}(\mathbf{Y}_i) = 1, \forall i \in \mathcal{V} & \text{(7c)}
\end{aligned}
$$

# Inverse Kinematics as an Optimization Problem

**R**: rotation matrices

$$\begin{aligned} \min_{\mathbf{u}} \quad & f(\mathbf{u}) && \text{(6a)} \\ \text{subject to} \quad & \mathbf{u} \in \mathcal{U} && \text{(6b)} \end{aligned}$$

$\Downarrow$ $\Downarrow$

**Y**: decision variables

$$\begin{aligned} \min_{\mathbf{Y}} \quad & f(\mathbf{Y}) && \text{(7a)} \\ \text{subject to} \quad & \mathbf{Y} \in \bar{\mathcal{U}} && \text{(7b)} \\ & \cancel{\operatorname{rank}(\mathbf{Y}_i) = 1, \forall i \in \mathcal{V}} && \text{(7c)} \end{aligned}$$

# Rank-Unconstrained Problem

**Problem 2b (Relaxed inverse kinematics)**

$$\min_{\mathbf{Y} \in \mathbb{R}^{7 \times 7 n_r}} \quad f(\mathbf{Y}) \tag{8a}$$

$$\text{subject to} \quad \mathbf{Y} \in \bar{\mathcal{U}} \tag{8b}$$

Strategy: solve Problem 2b then find rank-1 solution in $\bar{\mathcal{U}}$.

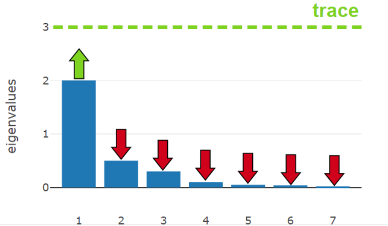
**Remark 1**

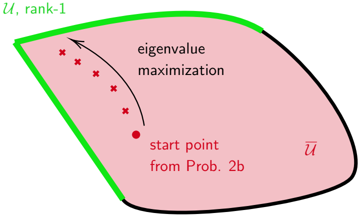
💡Because The relaxed set $\bar{\mathcal{U}}$ contains every element of $\mathcal{U}$, i.e., $\mathcal{U} \subset \bar{\mathcal{U}}$, problem 2b can certify infeasibility.

# Rank Minimization

Main idea:

- The trace of a matrix is also the sum of all of its eigenvalues,
- which are all non-negative when the matrix is positive semidefinite.
- Maximizing the largest eigenvalue of a matrix with constant trace is the same as minimizing all other eigenvalues.



# Rank Minimization: Gradient Approach

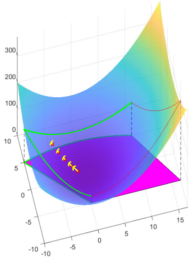



**Figure 3:** Maximizing a convex function over a convex set using gradient ascent

# Simulation Results

# Simulation Results

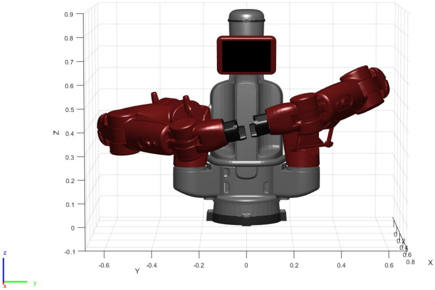
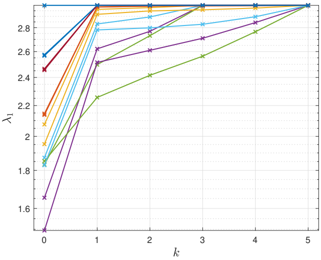


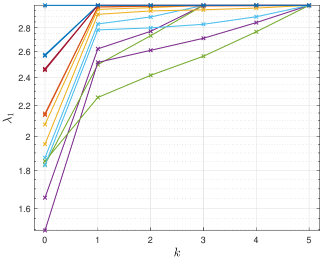

**Figure 4:** An example posture solved for Baxter, where the two arms are modeled as one closed kinematic chain.

# Simulation Results

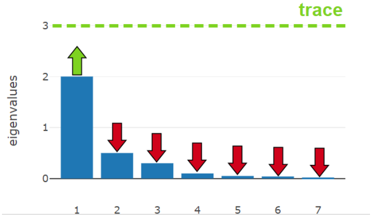



**(a)** The largest eigenvalues $\lambda_1$ of each $\mathbf{Y}_{k,i}$ over iteration $k$, where each line corresponds to one matrix.

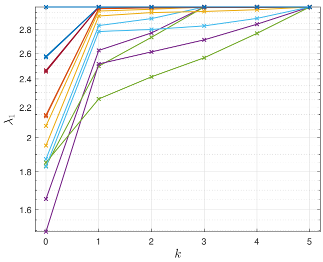
# Simulation Results



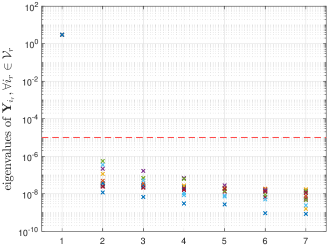

**(a)** The largest eigenvalues $\lambda_1$ of each $\mathbf{Y}_{k,i}$ over iteration $k$, where each line corresponds to one matrix.



# Simulation Results

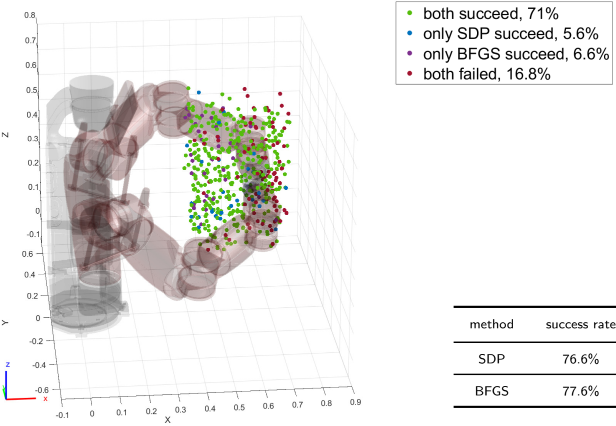



**(a)** The largest eigenvalues $\lambda_1$ of each $\mathbf{Y}_{k,i}$ over iteration $k$, where each line corresponds to one matrix.




**(b)** The eigenvalues of each $\mathbf{Y}_i$ in the solution, where all eigenvalues except the largest one are below the tolerance $\epsilon_1$ (red dashed line)

# Simulation Results



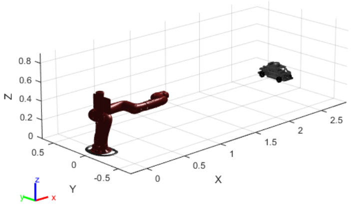

| method | success rate | avg. time |
|---|---|---|
| SDP | 76.6% | 1.2629 s |
| BFGS | 77.6% | 0.1966 s |

# Conclusion

- Single formulation that can incorporate more general kinematic constraints and arrangements of links
  - E.g., parallel robots, multi-robots, humanoids.
- Novel SDP relaxation
  - Novel low-rank projection based on fixed-trace matrices
  - Infeasibility certificates

# On-going/Future Work

- Incorporate different constraints in robot control and motion planning.
    - e.g., vision-based constraints, forces, etc.
- Investigate ways to further reduce the problem size.
- Develop constraints for prismatic joints.




## Thank you!

# Questions?

**Liangting Wu** tomwu@bu.edu

**Roberto Tron** tron@bu.edu

Website: https://sites.bu.edu/tron

# Rank Minimization: Gradient Approach

**Problem 3 (Sequential problem)**

$$\max_{\mathbf{U}_k \in \mathbb{R}^{7\times 7n_r}} \quad \sum_{i\in\mathcal{V}} \mathrm{vec}(\mathbf{U}_{k,i})^{\mathsf{T}} \frac{\partial \lambda_1(\mathbf{Y}_{k-1,i})}{\partial \mathbf{Y}_{k-1,i}} \tag{9a}$$

$$\text{subject to} \quad \nabla h(\mathbf{U}_k) = \mathbf{0} \tag{9b}$$

$$g(\mathbf{Y}_{k-1} + \mathbf{U}_k) \in \bar{\mathcal{U}} \tag{9c}$$

- The variable $\mathbf{U}_k$ is an update of $\mathbf{Y}_k$, i.e., $\mathbf{Y}_{k,i} = \mathbf{Y}_{k-1,i} + \mathbf{U}_{k,i}$.
- (9a) ensures that each $\mathbf{U}_{k,i}$ moves in the direction of the largest possible improvement in terms of increasing the sum of largest eigenvalues of the matrices $\mathbf{Y}_{k,i}$.

## Algorithm

**Algorithm 1**

**Input** $\mathbf{T}_{goal}, \mathbf{R}_{goal}, \mu, \epsilon_1, \epsilon_2, k_{max}$

**Output** $\mathbf{x}^*$

Solve Problem 2b to get an initial solution $\mathbf{Y}_0$ and set $k = 1$.
**while** $\exists \lambda_{1,i} \leq 3 - \epsilon_1 \ \& \ \|\mathbf{U}_k\|_F \geq \epsilon_2 \ \& \ k \leq k_{max}$ **do**
    For each $\mathbf{Y}_{k,i}$, compute the largest eigenvalue $\lambda_{1,i}$ and the corresponding normalized eigenvector $\mathbf{V}_{k-1,i}^{(1)}$.
    Solve Problem 3 to get $\mathbf{U}_k$.
    Update $\mathbf{Y}_{k,i} = \mathbf{Y}_{k-1,i} + \mathbf{U}_{k,i}$ for all $i \in \mathcal{V}$ and set $k = k+1$.
**end while**
Recover the rotations $\{\mathbf{R}_i\}$ by reshaping $g(\mathbf{Y}_{k-1})$.
Recover the translations $\{\mathbf{T}_i\}$ using the relation $\mathbf{T}_j - \mathbf{T}_i = \mathbf{R}_i {}^i\mathbf{T}_j$.
**return** $\mathbf{x}^*$

# Revolute joint

For each pair of links $(i, j) \in \mathcal{E}_r$ that are connected with a revolute joint, the orientations $\mathbf{R}_i$ and $\mathbf{R}_j$ are limited by the equation

$$\mathbf{R}_j = \mathbf{R}_i \mathbf{R}_e \mathbf{R}_\theta \tag{10}$$

$\mathbf{R}_\theta : \mathbb{R} \mapsto \mathbf{SO}(3)$ is a function of the joint angle $\theta$ defined as a rotation about the $z$-axis.

$\mathbf{R}_e$ is a parameter defined as the rotation from $\mathcal{B}_j$ to $\mathcal{B}_i$ when $\theta = 0$.

$\mathbf{R}_\theta : \mathbb{R} \mapsto \mathbf{SO}(3)$ is a function of the joint angle $\theta$ defined such that $\mathbf{R}_\theta = \mathbf{I}$ when $\theta = 0$;

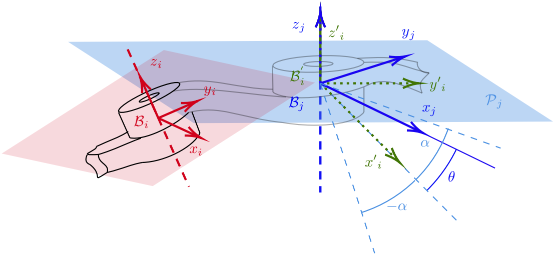
$\mathbf{R}_e$ is a parameter defined as the rotation from $\mathcal{B}_j$ to $\mathcal{B}_i$ when $\theta = 0$.

**Figure 6:** A visualization of a revolute joint and the associated reference frames.

# Revolute Joint Axis Constraints

The frame $\mathcal{B}'_i$ and $\mathcal{B}_j$ share the same $z-$axis, that is:

$$\mathbf{R}_i\mathbf{R}_e\mathbf{e}_3 - \mathbf{R}_j\mathbf{e}_3 = \mathbf{0} \tag{11}$$




---

$\mathbf{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\mathsf{T}$

# Revolute Joint Angle Limits

The joint angle limits require that, $\forall (i,j) \in \mathcal{E}_r$,

$$\mathbf{w}_i - \mathbf{w}_j = \mathbf{R}_i \mathbf{R}_e \mathbf{e}_1 - \mathbf{R}_j \mathbf{e}_1 \in \mathcal{S}\big(\sqrt{2 - 2\cos(\alpha_{ij})}\big). \qquad (12)$$

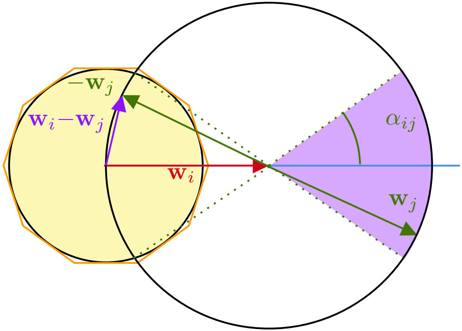



**Figure 7:** The joint limit between two links $(i,j) \in \mathcal{E}_r$ can be written as an angle limit between two unit vectors $\mathbf{w}_i = \mathbf{R}_i \mathbf{R}_e \mathbf{e}_1$ and $\mathbf{w}_j = \mathbf{R}_j \mathbf{e}_1$ (purple sector), which can be further bounded by a ball (painted yellow) on $\mathbf{w}_i - \mathbf{w}_j$.