ASTRO-2022-A004

Nonholonomic Attitude Reorientation of Free-floating Astronauts by Joint Actuation
（浮遊状態の宇宙飛行士の関節駆動による非ホロノミック姿勢変更）

2022/07/25

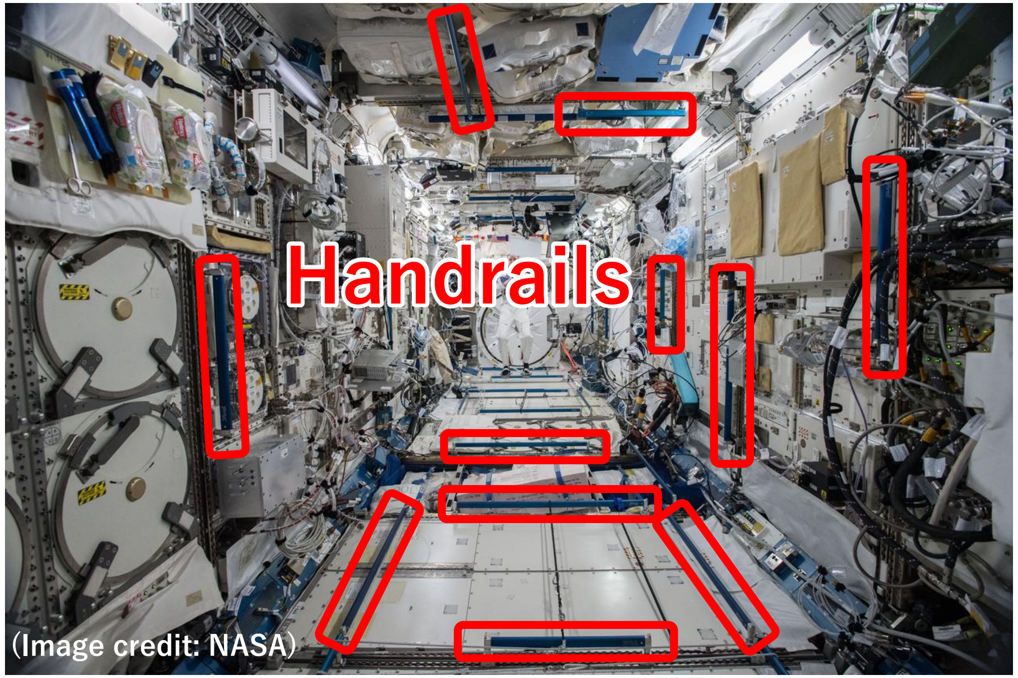
32nd Workshop on JAXA Astrodynamics and Flight Mechanics

○ Yuki Kubo (JAXA)
Junichiro Kawaguchi (ANU)

1. Introduction
- Background

- ✓ More manned space exploration & space tourism
  - ➢ **Longer** & **wider** space habitation
- ✓ Broader space in zero-G area
  - ➢ **Some area might not have handrails**

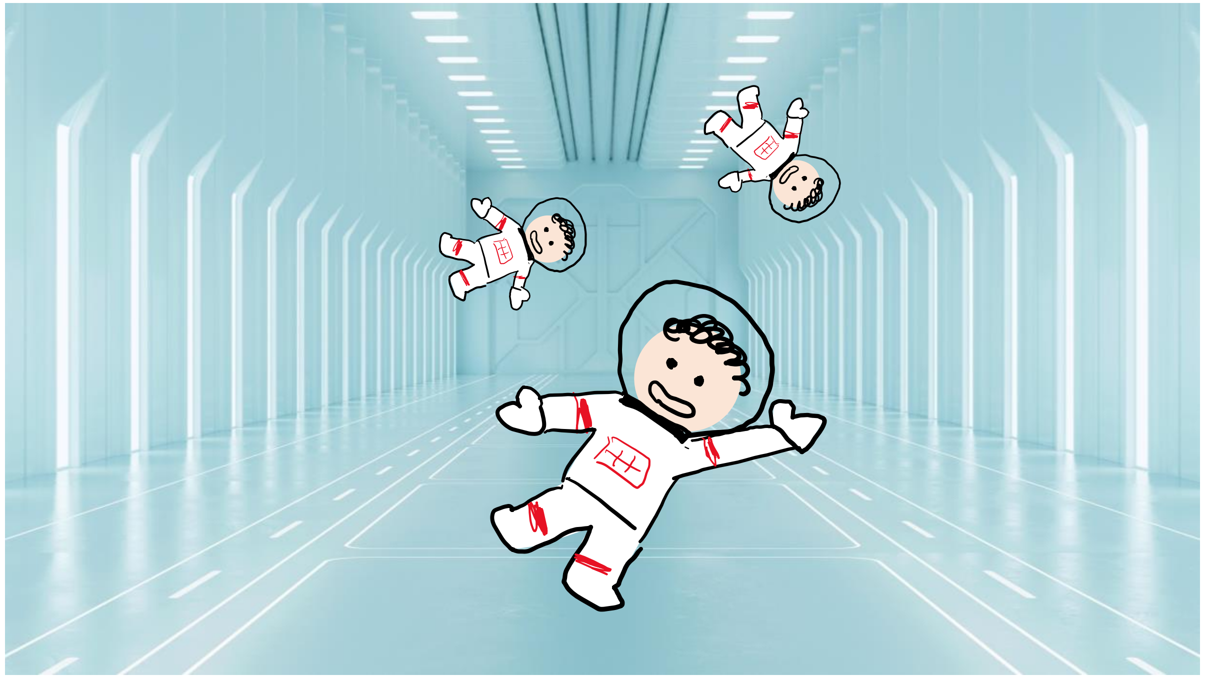
Question:
How can we change the attitude in free-floating state?



(Image credit: NASA)

Current design of ISS

?

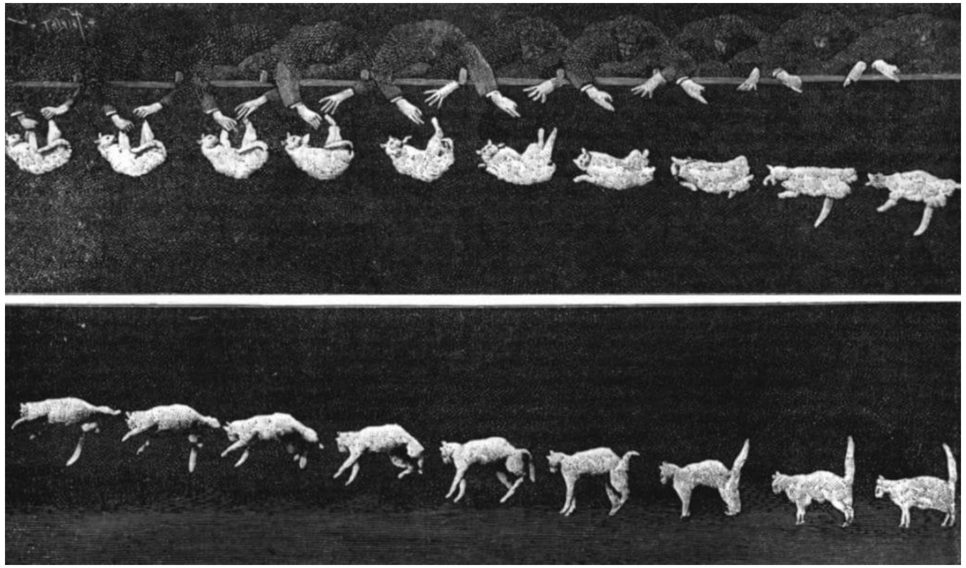


Broader space in zero-G area

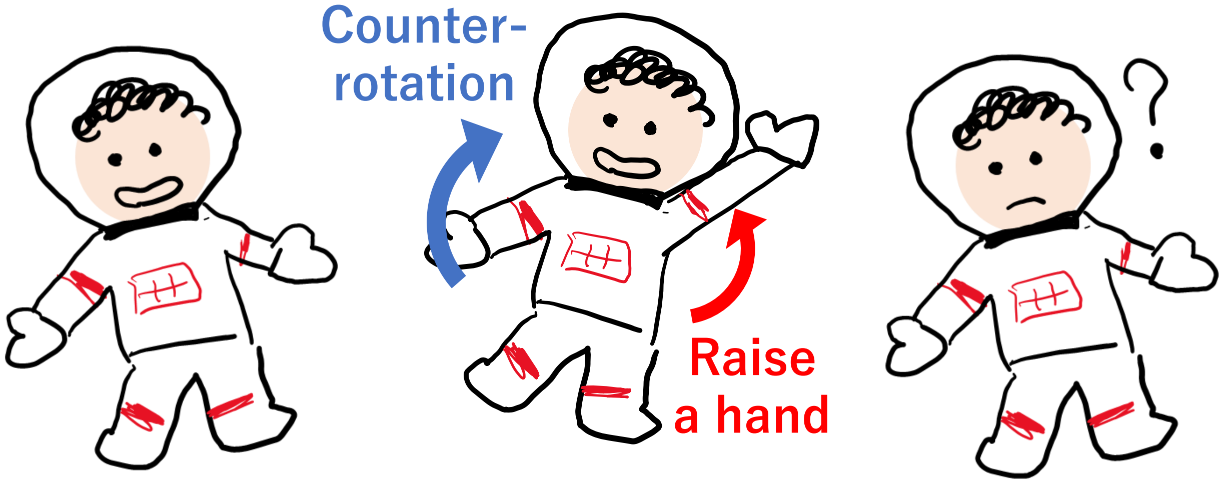
1. Introduction
- Is it possible?

Q. Can we change the attitude only by joint actuation?

Well known phenomenon:
Falling cat motion (cat righting reflex)



Q. But... how?



Free-floating state

Return to original attitude

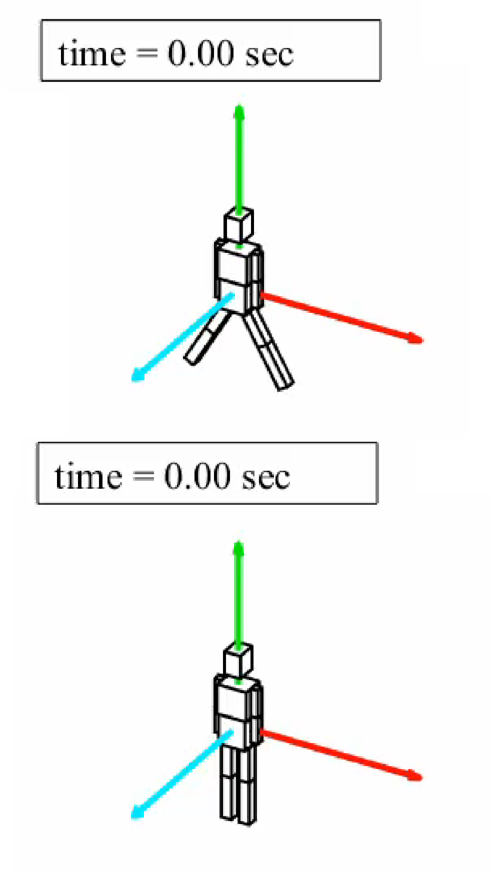
Nonholonomy → Final attitude depends on joint actuation procedure

Need to manage
- attitude (absolute attitude)
- joint config (relative attitude)

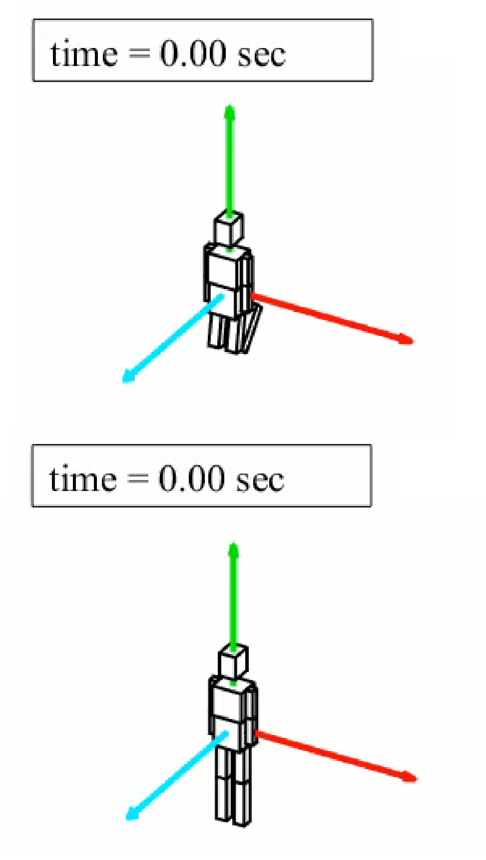
1. Introduction
- Fundamental maneuvers

- ✓ Fundamental maneuvers have been investigated during Mercury & Gemini program [Kulwicki & Schlei, 1962]
- ✓ Easy to carry out, but efficiency is not high

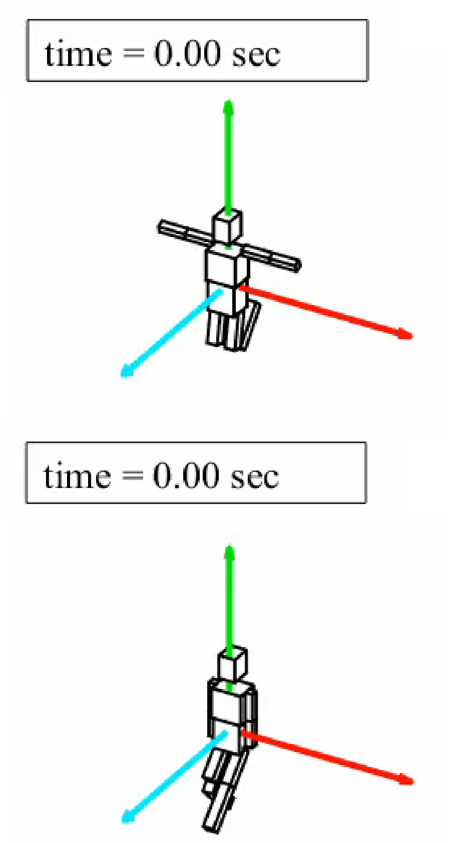
Yaw



Roll



Pitch

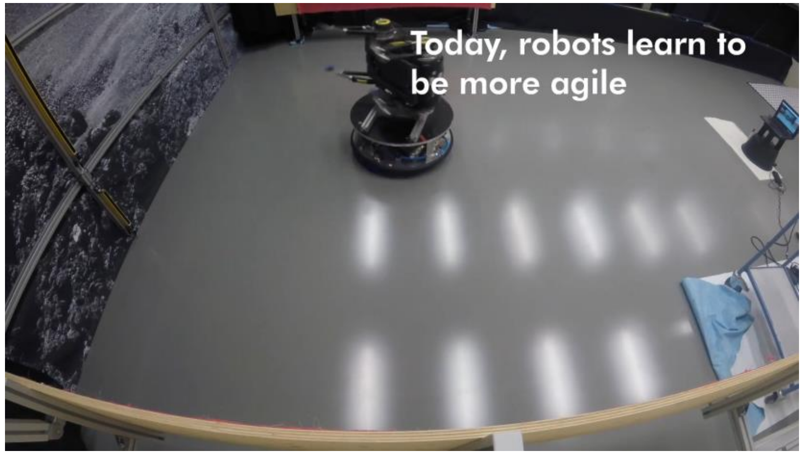


1. Introduction
- Related studies

- Dynamics of free-flying multibody has been intensively studied since 1980s
- Various nonholonomic motion planning methods
  - ➢ Deep reinforcement learning, Motion primitive-based planning, Sinusoidal input…

Micro-G legged robot (ETHZ)

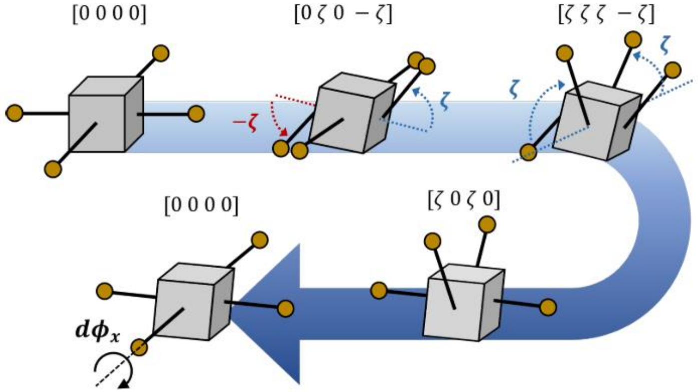
- Jump & reorientation by pedaling
- Deep reinforcement learning



(N. Rudin et al., 2021)

Hibari (Tokyo Tech.)

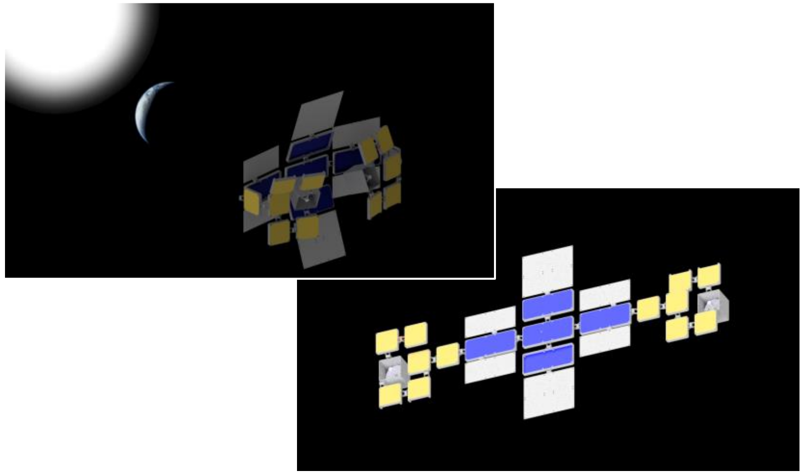
- Agile (holonomic) reorientation
- Motion primitive-based nonholonomic reorientation



(F. Watanabe et al., 2016)

Transformer (JAXA, TF WG)

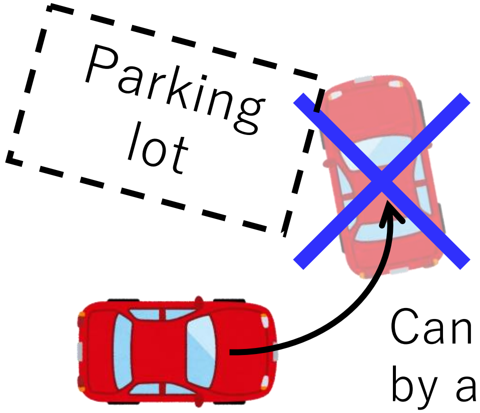
- Nonholonomic reorientation & body reconfiguration
- Optimization with constraints



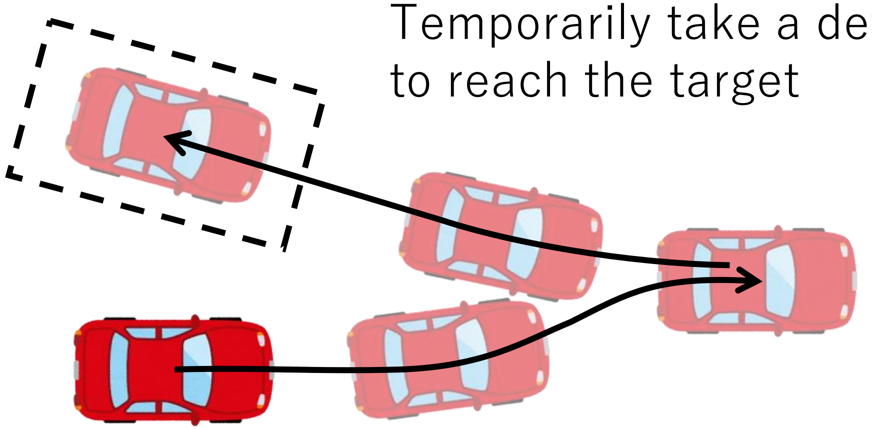
(Kubo & Kawaguchi, 2022)

1. Introduction
- Difficulties of nonholonomic motion planning

e.g.) Car parking

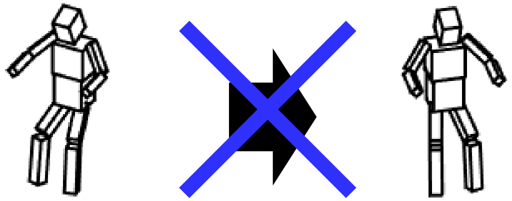


Cannot reach the target
by a time-invariant feedback



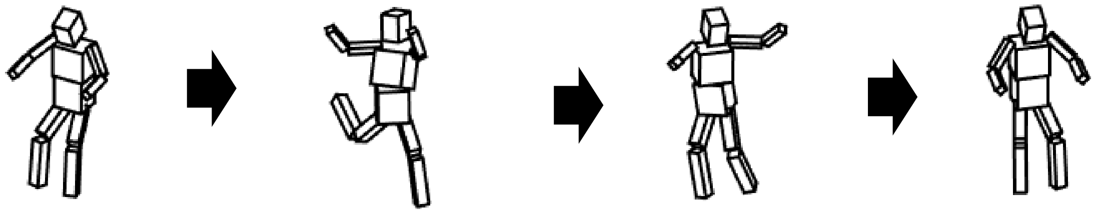
Temporarily take a detour route
to reach the target

e.g.) Nonholonomic reorientation



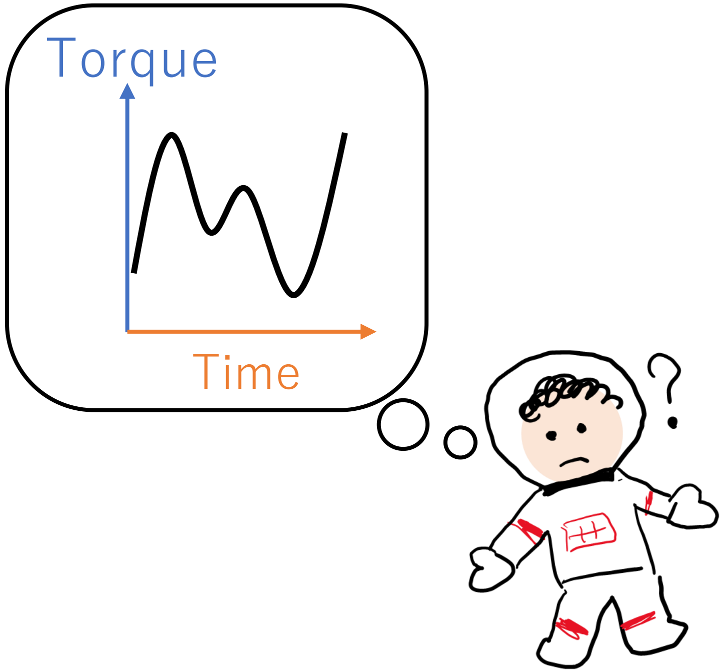
Cannot reach the target
with a time-invariant feedback

Target config.
& attitude

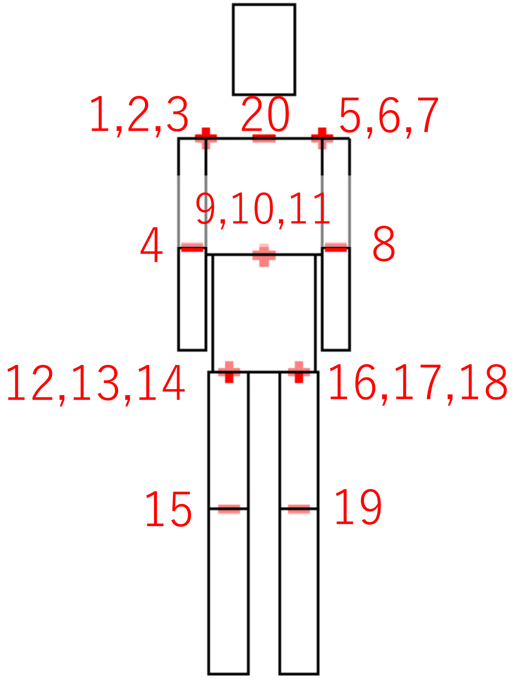


Need to take intermediate complicated pose
→ Infinite number of solutions, difficult to explore

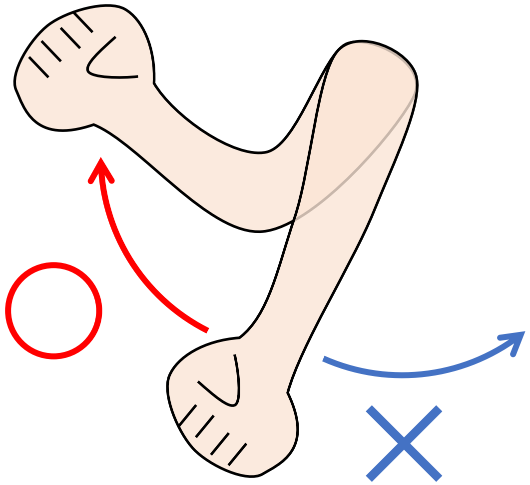
1. Introduction
- Specific difficulties for astronauts



Torque tracking difficulty



Large input DoF



Limited joint angle range

Most of the conventional methods {
generates **too complex** joint motion
are **difficult to handle** large input DoF
does not satisfy some constraints

1. Introduction
- Purpose

Difficulties of nonholonomic system

- ✓ Cannot use time-invariant feedback
- ✓ Infinite number of solutions

Difficulties for astronauts

- ✓ Torque tracking
- ✓ Large input DoF
- ✓ Limited joint angle range

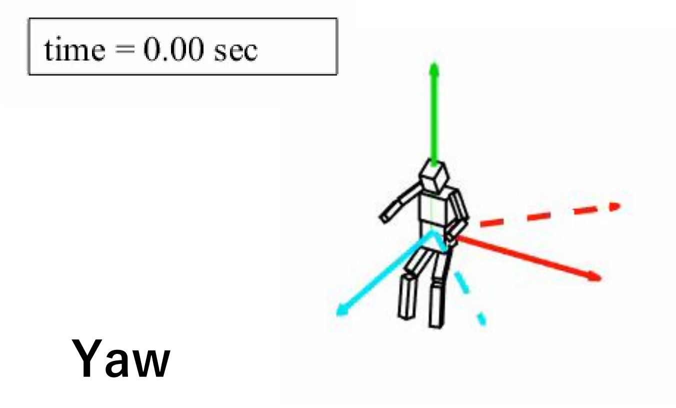
<u>Purpose</u>

To propose **nonholonomic attitude reorientation** method available for **free-floating astronauts**

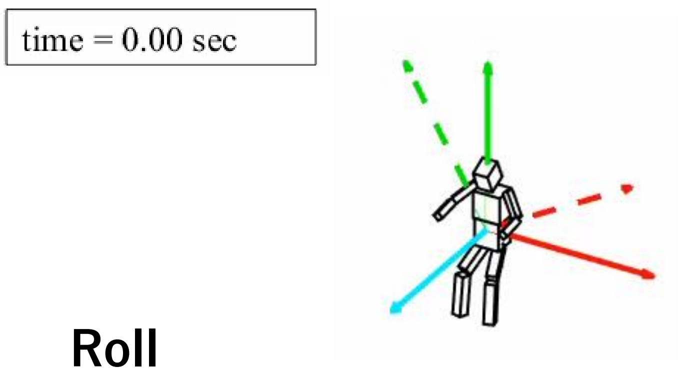
→ **<u>Rectilinear transformation planning (Posing-based maneuver)</u>**

✓ Can generate <u>constrained optimized motion</u>

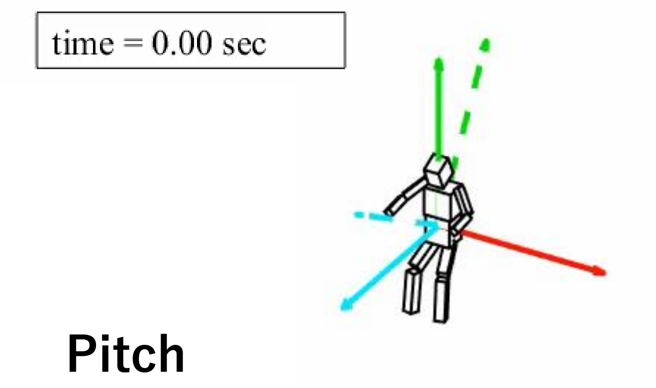
✓ Can reorient efficiently with <u>simple joint actuation</u> maneuvers



Yaw



Roll



Pitch

2. Method
- Idea of the proposed method

Rectilinear transformation planning

Design a **rectilinear** path in **joint angle space**

→ **Posing-based** maneuver

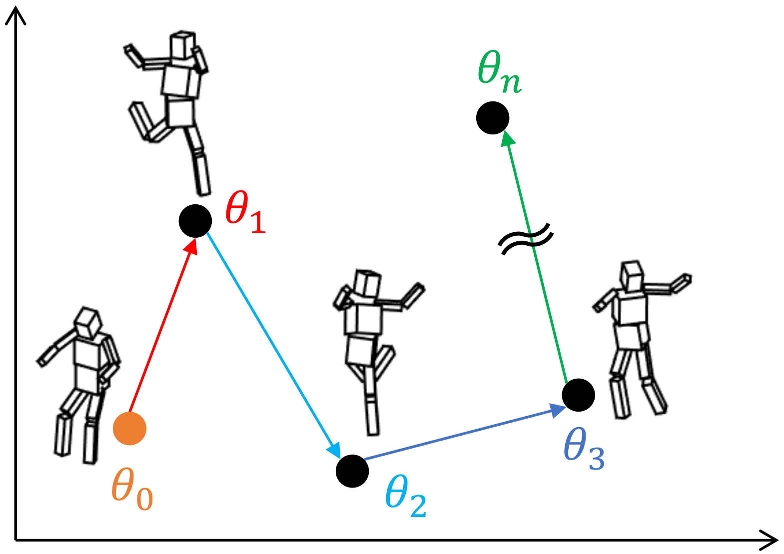
1. Easy actuation & complex attitude motion
2. No combinatorial explosion
3. Easy to handle joint angle constraints

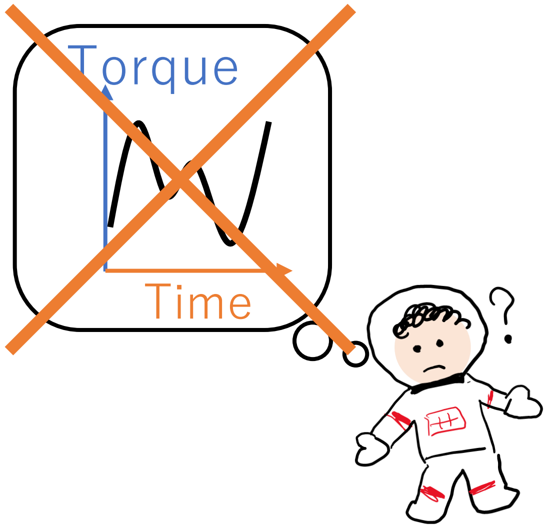
↔

Difficulties for astronauts

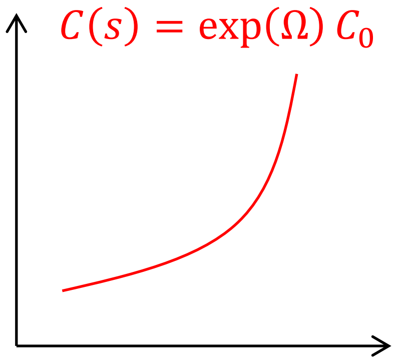
1. Torque tracking
2. Large input DoF
3. Limited joint angle range

Joint angle space (m-dim)

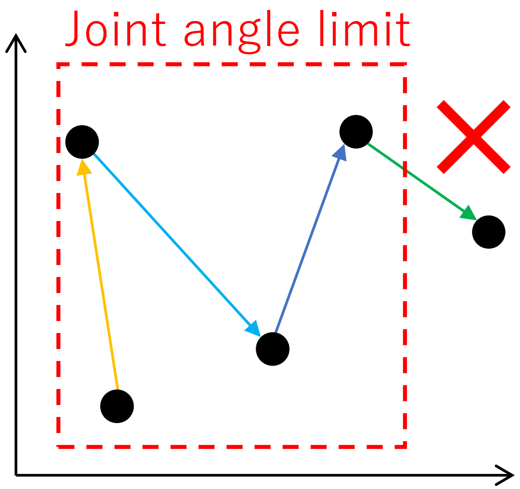




NOT torque-based



Analytical solution



Easy limit check

2. Method
- Approximate analytical solution

Assumption

- ✓ Zero angular momentum
- ✓ No external force (AM conservation)
- ✓ Joint speed input

Fundamental eq.: DCM Kinematics eq.

$$\dot{C} = -\omega^{\times} C = \left(g(\theta)\dot{\theta}\right)^{\times} C$$

C	Rotational matrix (DCM)
ω	Angular velocity vector $\in \mathbb{R}^3$
θ	Joint angle vector $\in \mathbb{R}^m$ (m: Joint num, input DoF)
$g(\theta)$	Rotational inertia $\in \mathbb{R}^{3\times m}$

Differential eq. $\dot{x} = Ax$

If A is constant : $x = \exp\left(\int A\mathrm{d}t\right) x_0$

If $A = A(t)$: Magnus Expansion

General solution as matrix exponential of infinite series

$$C(t) = \exp\left(\sum_{k}^{\infty} \Omega_k(t)\right) C_0$$

$$\Omega_1(t) = \left(\int_0^t g(t)\dot{\theta}(t)\,\mathrm{d}t\right)^{\times}$$

※ $[X, Y] = XY - YX$ (Lie bracket)

$$\Omega_2(t) = \frac{1}{2}\int_0^t \mathrm{d}t_1 \int_0^{t_1} \mathrm{d}t_2 \underbrace{\left[\left(g(t_1)\dot{\theta}(t_1)\right)^{\times}, \left(g(t_2)\dot{\theta}(t_2)\right)^{\times}\right]}_{\text{Effect of rotational axis drift}}$$

$$\vdots$$

2. Method
- Approximate analytical solution

General solution

$$C(t) = \exp\left(\sum_{k}^{\infty} \Omega_k(t)\right) C_0$$

$$\Omega_1(t) = \left(\int_0^t g(t)\dot{\theta}(t)\,\mathrm{d}t\right)^{\times}$$

$$\Omega_2(t) = \frac{1}{2}\int_0^t \mathrm{d}t_1 \int_0^{t_1} \mathrm{d}t_2 \left[\left(g(t_1)\dot{\theta}(t_1)\right)^{\times}, \left(g(t_2)\dot{\theta}(t_2)\right)^{\times}\right]$$

$$\vdots$$

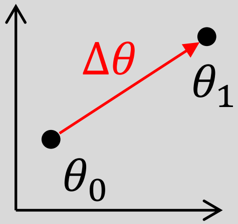
Approximate $g(t)$ as polynomial with Newton or Hermite iterpolation

$$g(t) = \sigma_0(t)A_0\gamma_0$$

Rectilinear Transformation

(Linear path in joint angle space)

$$\theta = \theta_0 + s\Delta\theta$$
$$(0 \le s \le 1)$$



Approximate analytical solution for rectilinear joint actuation

Ω_k becomes k-th order tensor

$$C(s) = \exp\left(\sum_{k}^{\infty} \Omega_k(s)\right) C_0$$

$$\Omega_1(s) = \sigma_1(s)A_1\gamma_1$$
$$\Omega_2(s) = \sigma_2(s)A_2\gamma_2$$
$$\vdots$$

$\sigma_k(s)$	Polynomial tensor of s
A_k	Coefficient tensor for interpolation
γ_k	g values at interpolated points

2. Method
- Nonholonomic sensitivity tensor

Take a Jacobian of C_N with respect to θ_k

$$C_N = \Phi(\theta_N,\theta_{N-1})\cdots\Phi(\theta_2,\theta_1)\Phi(\theta_1,\theta_0)C_0$$

Derivative of matrix exponential

$$\frac{\mathrm{d}e^{\Omega(\theta)}}{\mathrm{d}\theta} = \int_0^1 e^{\alpha\Omega(\theta)}\frac{\mathrm{d}\Omega(\theta)}{\mathrm{d}\theta}e^{(1-\alpha)\Omega(\theta)}\mathrm{d}\alpha$$

$$\frac{\partial C_N}{\partial\theta_k} = \Phi_N\cdots\Phi_{k+2}\left(\frac{\partial\Phi_{k+1}}{\partial\theta_k}\Phi_k+\Phi_{k+1}\frac{\partial\Phi_k}{\partial\theta_k}\right)\Phi_{k-1}\cdots\Phi_1C_0$$

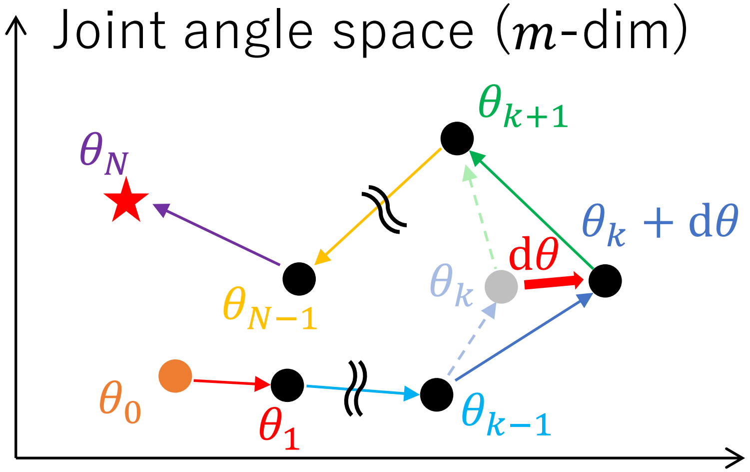
Transform into $\frac{\partial e_N}{\partial\theta_k}$

$\frac{\partial e_N}{\partial\theta_k}$ expresses sensitivity of final attitude with respect to difference of θ_k

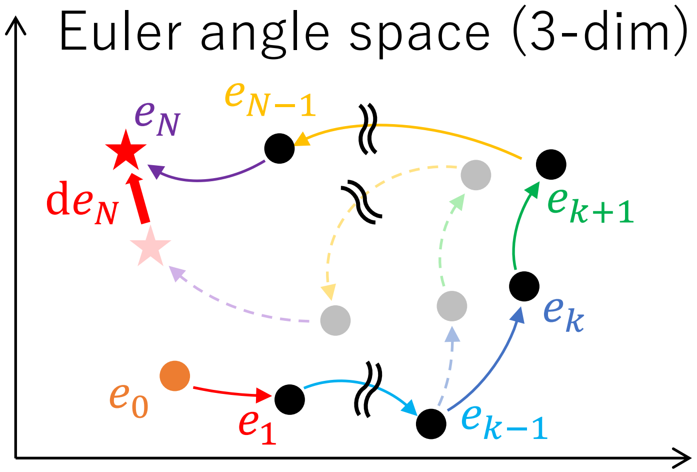
→ Provides **path modification**

$$[\theta_1 \quad \cdots \quad \theta_{n-1}]^{\text{updated}} = [\theta_1 \quad \cdots \quad \theta_{n-1}] + \left(\frac{\partial e_n}{\partial\theta}\right)^{\dagger}(e_{\mathrm{t}} - e_n)$$

$()^{\dagger}$: Proper pseudo inverse



Shift θ_k by $\mathrm{d}\theta$



Final attitude is shifted by $\mathrm{d}e$

3. Simulation
- Optimization procedure

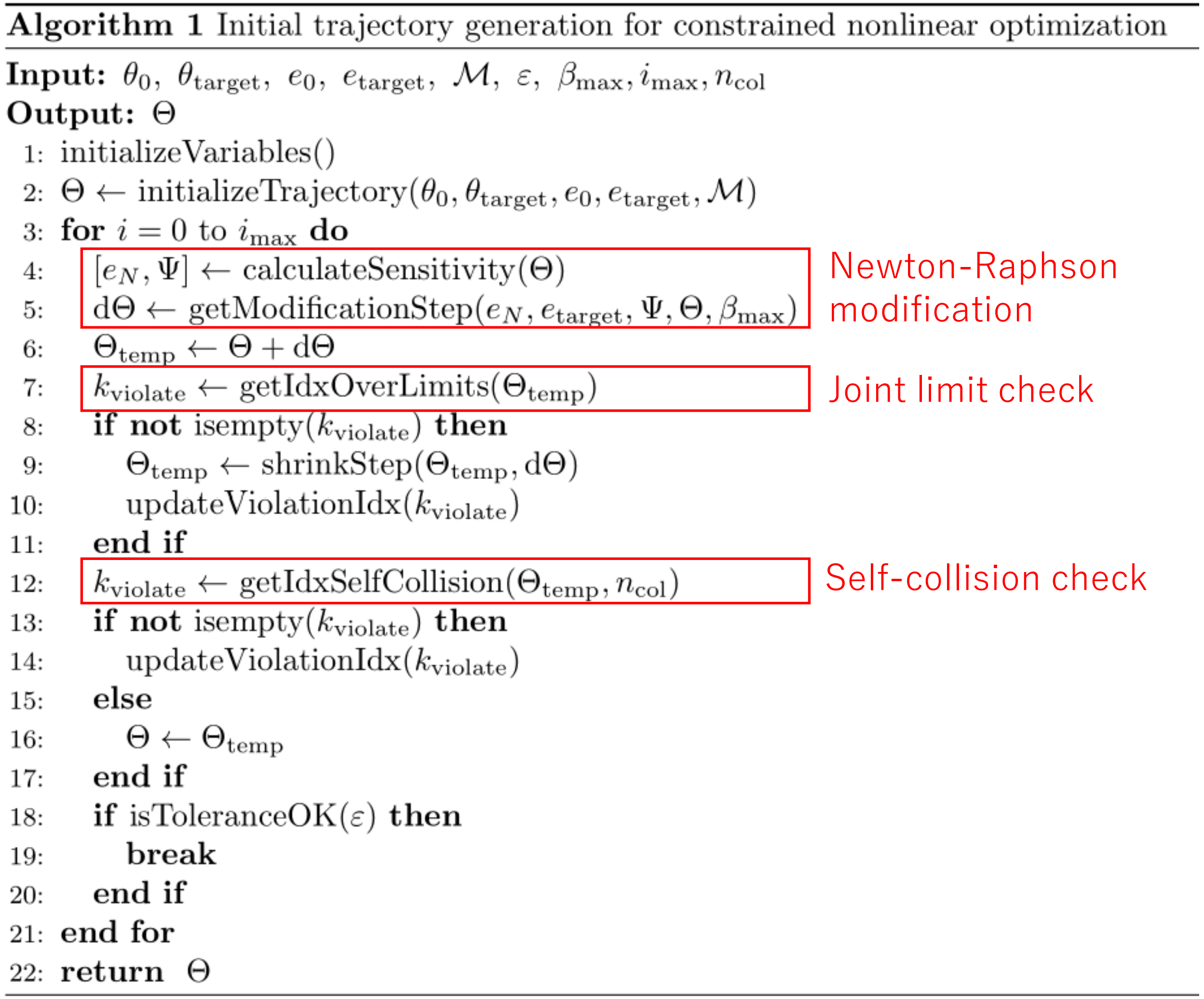
Design the following rectilinear transformation planning which satisfies:

- **joint angle limits**
- **self-collision avoidance**,

while **minimizing total joint travel**

Strategy

1. Approach target attitude as close as possible while satisfying all constraints (Algorithm 1)
2. Using 1. as initial guess, obtain solution which satisfies final target attitude
3. Using 2. as initial guess, minimize total joint travel with constrained optimization solver (Algorithm: interior point method)



Algorithm 1 Initial trajectory generation for constrained nonlinear optimization

```
Input: $\theta_0,\ \theta_{\text{target}},\ e_0,\ e_{\text{target}},\ \mathcal{M},\ \varepsilon,\ \beta_{\max}, i_{\max}, n_{\text{col}}$
Output: $\Theta$
initializeVariables()
$\Theta \leftarrow$ initializeTrajectory($\theta_0, \theta_{\text{target}}, e_0, e_{\text{target}}, \mathcal{M}$)
for $i = 0$ to $i_{\max}$ do
    $[e_N, \Psi] \leftarrow$ calculateSensitivity($\Theta$)
    $\mathrm{d}\Theta \leftarrow$ getModificationStep($e_N, e_{\text{target}}, \Psi, \Theta, \beta_{\max}$)
    $\Theta_{\text{temp}} \leftarrow \Theta + \mathrm{d}\Theta$
    $k_{\text{violate}} \leftarrow$ getIdxOverLimits($\Theta_{\text{temp}}$)
    if not isempty($k_{\text{violate}}$) then
        $\Theta_{\text{temp}} \leftarrow$ shrinkStep($\Theta_{\text{temp}}, \mathrm{d}\Theta$)
        updateViolationIdx($k_{\text{violate}}$)
    end if
    $k_{\text{violate}} \leftarrow$ getIdxSelfCollision($\Theta_{\text{temp}}, n_{\text{col}}$)
    if not isempty($k_{\text{violate}}$) then
        updateViolationIdx($k_{\text{violate}}$)
    else
        $\Theta \leftarrow \Theta_{\text{temp}}$
    end if
    if isToleranceOK($\varepsilon$) then
        break
    end if
end for
return $\Theta$
```

Newton-Raphson modification

Joint limit check

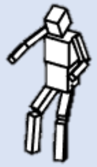
Self-collision check

3. Simulation
- Result

Result

- ✓ Satisfying joint limits and self-collision constraints
- ✓ Final attitude reaches the target with a certain accuracy
- ✓ Total joint travel is minimized from 3128.2 deg to 1376.3 deg

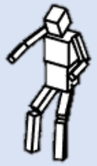
Initial config.



Initial attitude

$[0 \quad 0 \quad 0]$ deg

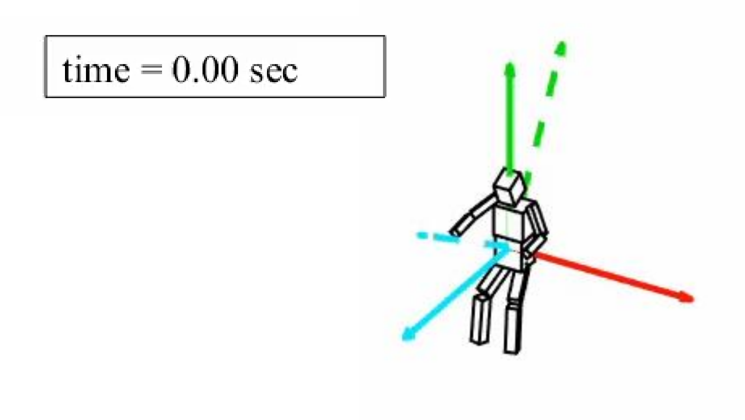
Target config.



Target attitude

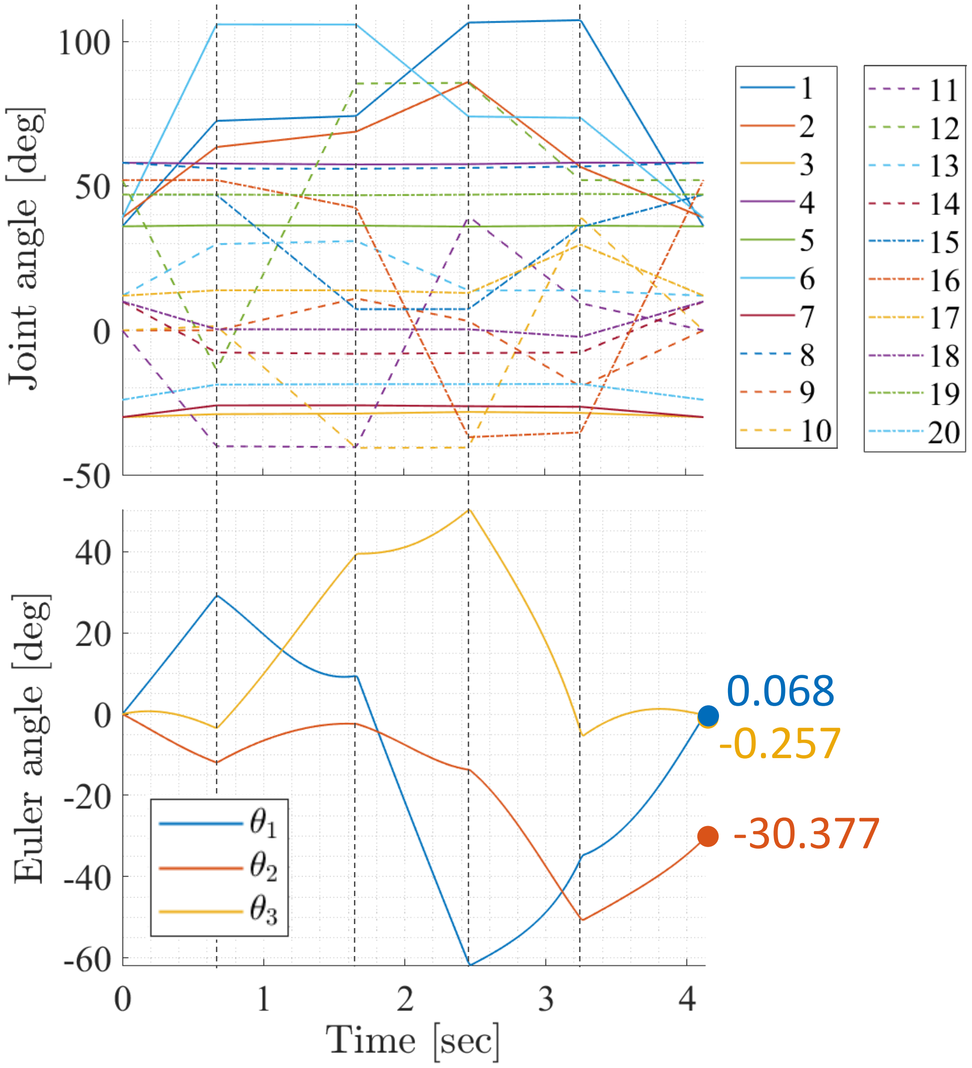
$[0 \quad -30 \quad 0]$ deg in 2-1-3 Euler angle

Number of strokes: 5



X Y Z ← Current attitude

X Y Z ⇠ Target attitude



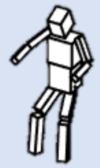
3. Simulation
- Result

Result

- ✓ Satisfying joint limits and self-collision constraints
- ✓ Final attitude reaches the target with a certain accuracy
- ✓ Total joint travel is minimized from 3128.2 deg to 1376.3 deg

Initial config.

Target config.



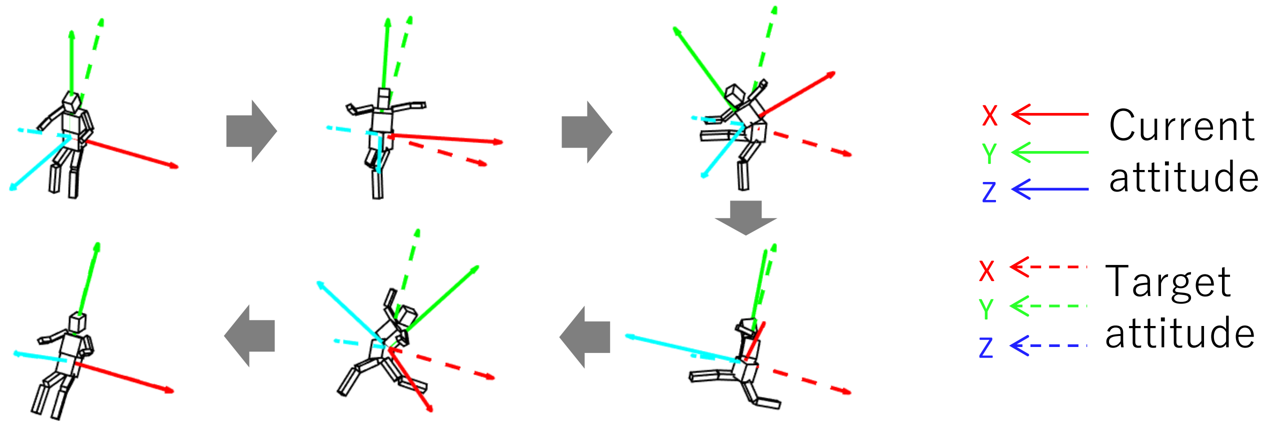
Number of strokes: 5

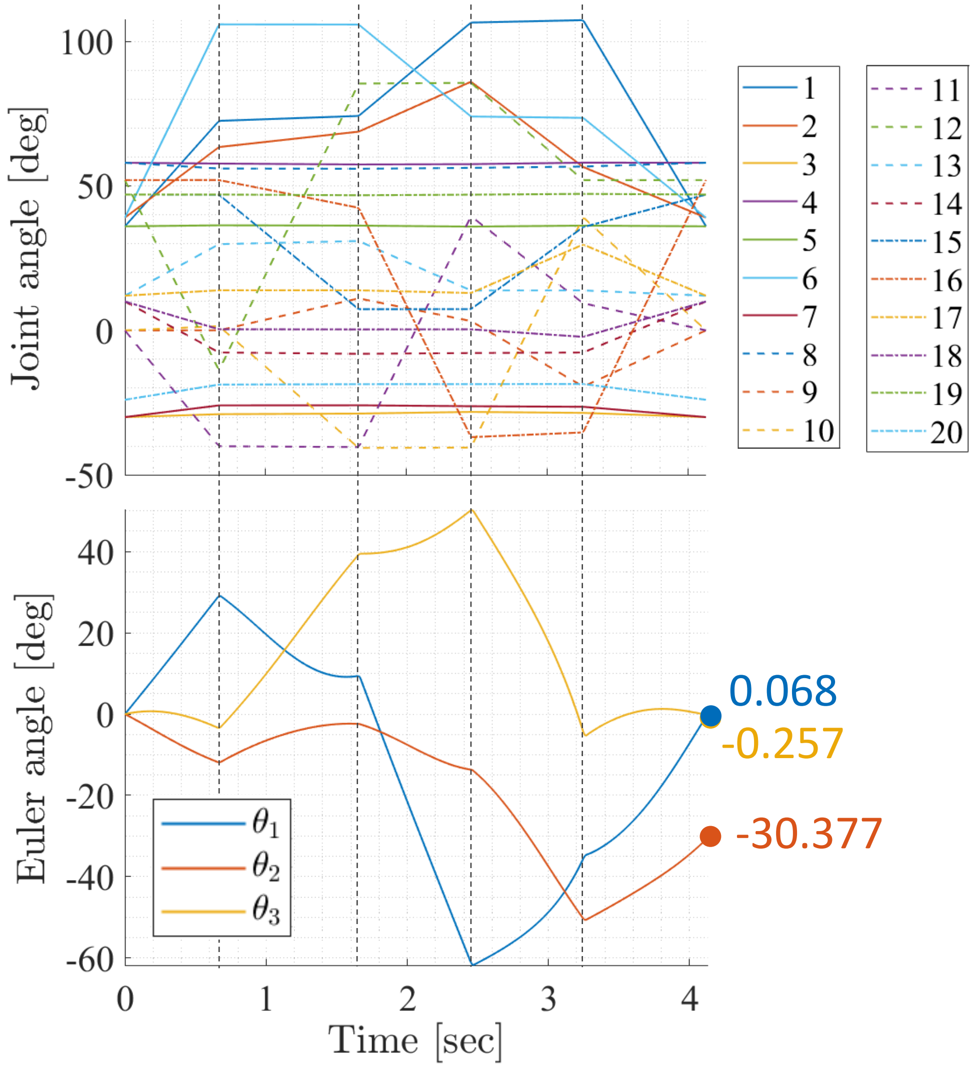
Initial attitude

$[0 \quad 0 \quad 0]$ deg

Target attitude

$[0 \quad -30 \quad 0]$ deg in 2-1-3 Euler angle

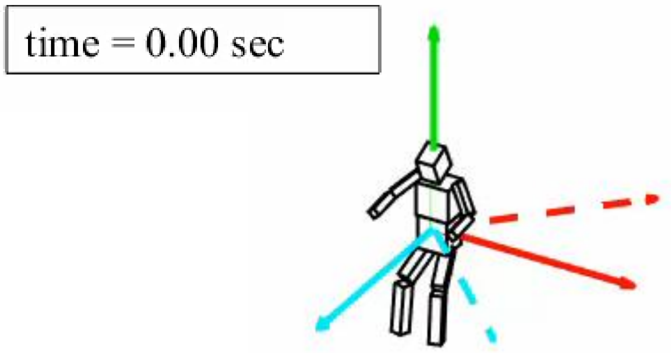




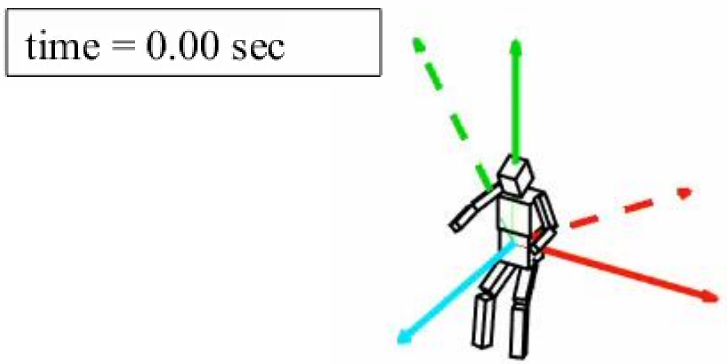
3. Simulation
- Result

Result

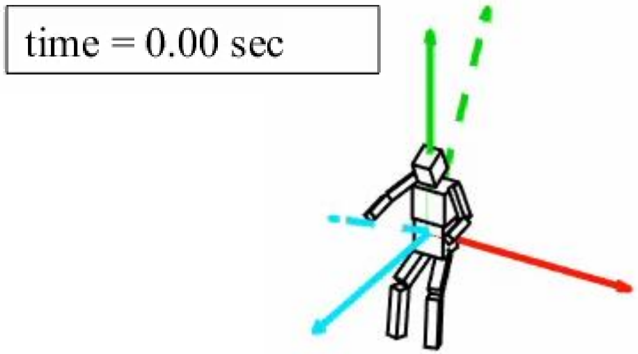
- ✓ Satisfying joint limits and self-collision constraints
- ✓ Final attitude reaches the target with a certain accuracy
- ✓ Total joint travel is minimized from 3128.2 deg to 1376.3 deg



Yaw



Roll



Pitch

→ Rapid nonholonomic reorientation maneuvers are obtained!

4. Summary

- ✓ **Nonholonomic attitude reorientation** maneuver for **free-floating astronauts** by joint actuation

- ✓ Proposed rectilinear transformation planning
  1. **Easy actuation & complex attitude motion**
  2. **No combinatorial explosion**
  3. **Easy to handle joint angle constraints**

- ✓ **Optimized** solutions **satisfying constraints** are obtained