SO FAR:

- Direct Kinematics $\quad \boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q})$
- Inverse Kinematics $\quad \boldsymbol{q} = \boldsymbol{f}^{-1}(\boldsymbol{x})$
- Jacobian $\quad \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \mathbf{J}\,\dot{\mathbf{q}}$

Geometric, Analytic

$$\mathbf{J}_G = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}(\gamma) \end{bmatrix} \mathbf{J}_A$$

KINEMATIC SINGULARITIES

The Jacobian is a 6 × n matrix mapping the $\mathbb{R}^n$ joint velocity space to the $\mathbb{R}^6$ operational velocity space:

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad \Rightarrow \quad d\mathbf{x} = \mathbf{J}(\mathbf{q})d\mathbf{q}$$

- So, this can be interpreted as a relationship between infinitesimal displacements in $\mathbb{R}^n$ and $\mathbb{R}^6$
- In general, rank($\mathbf{J}(\mathbf{q})$) = min (6, n)
- On the other hand, since the elements of **J** are functions of the joints, some robot configurations exist such that the Jacobian loses rank
- These configurations are denoted as **kinematic singularities**

In these configurations, there are “directions” (vectors $\dot{\mathbf{x}}$) in $\mathbb{R}^6$ without any correspondent “direction” ($\dot{\mathbf{q}}$) in $\mathbb{R}^n$: these directions cannot be actuated and the robot loses motion capabilities

Singular configurations

The singular configurations are important for several reasons:

1. They represent configurations in which the motion capabilities of the robot are reduced: it is not possible to impose arbitrary motions of the end-effector

2. In the proximity of a singularity, small velocities in the operational space may generate large (infinite) velocities in the joint space

3. They correspond to configurations that have not a well defined solution to the inverse kinematic problem: either no solution or infinite solutions

Singular configurations

There are two types of singularities:

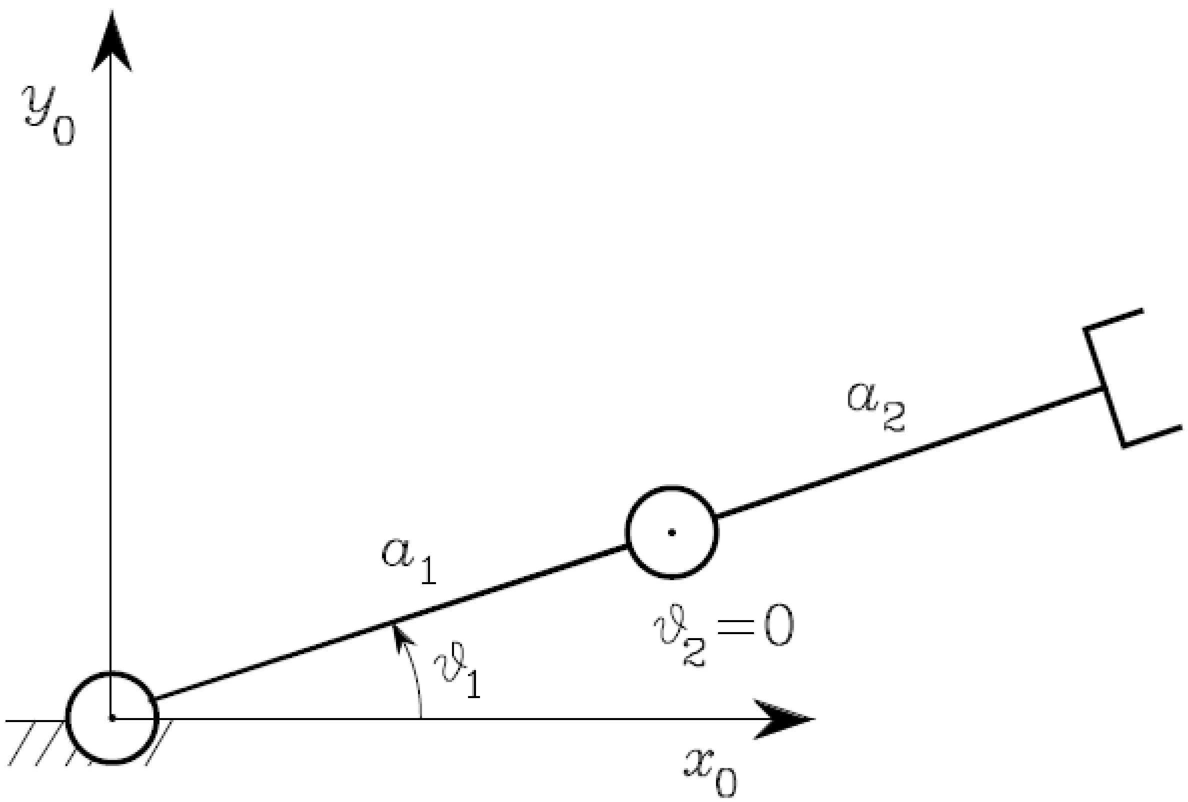
1. **Boundary singularities**, that correspond to points on the border of the workspace, i.e. when the robot is either fully extended or retracted. These singularities may be easily avoided by not driving the manipulator to the border of its workspace.

2. **Internal singularities**, that occur inside the reachable workspace and are generally caused by the alignment of two or more axes of motion, or else by the attainment of particular end-effector configurations. These singularities constitute a serious problem, as they can be encountered anywhere in the reachable workspace for a planned path in the operational space.

KINEMATIC SINGULARITIES-Example

- Two–link planar arm

$$\boldsymbol{J} = \begin{bmatrix} -a_1 s_1 - a_2 s_{12} & -a_2 s_{12} \\ a_1 c_1 + a_2 c_{12} & a_2 c_{12} \end{bmatrix}$$



$$\det(\boldsymbol{J}) = a_1 a_2 s_2$$

$$\Downarrow$$

$$\vartheta_2 = 0 \qquad \vartheta_2 = \pi$$

$[\,-(a_1 + a_2)s_1 \quad (a_1 + a_2)c_1\,]^T$ parallel to $[\,-a_2 s_1 \quad a_2 c_1\,]^T$

(components of end-effector velocity non-independent)

SINGULARITY DECOUPLING

In case of complex structures, the analysis of the kinematic singularities via the Jacobian determinant det(**J**) may prove quite difficult

In case of manipulators with spherical wrist, by similarity with the inverse kinematics, it is possible to decompose the study of the singular configurations into two sub-problems:

- computation of **arm** singularities
- computation of **wrist** singularities

If **J** $\in \mathbb{R}^{6\times n}$ then

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{J}_{12} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix}$$

where, since the last three joints are of the revolute type:

$$\mathbf{J}_{12} = [\mathbf{z}_3 \times (\mathbf{p}_6 - \mathbf{p}_3), \ \ \mathbf{z}_4 \times (\mathbf{p}_6 - \mathbf{p}_4), \ \ \mathbf{z}_5 \times (\mathbf{p}_6 - \mathbf{p}_5)]$$

$$\mathbf{J}_{22} = [\mathbf{z}_3, \ \ \mathbf{z}_4, \ \ \mathbf{z}_5]$$

SINGULARITY DECOUPLING

Singularities depend on the mechanical structure, not on the frames chosen to describe kinematics

Therefore, it is convenient to choose the origin of the end-effector frame at the intersection of the wrist axes, where also the last frames are placed

Then

$$\mathbf{J}_{12} = [\mathbf{0}, \;\; \mathbf{0}, \;\; \mathbf{0}] \quad \Longrightarrow \quad \mathbf{J} = \begin{bmatrix} \mathbf{J}_{11} & \mathbf{0} \\ \mathbf{J}_{21} & \mathbf{J}_{22} \end{bmatrix}$$

In this manner, **J** is a block lower-triangular matrix, and

$$det(\mathbf{J}) = det(\mathbf{J}_{11})det(\mathbf{J}_{22})$$

The singularities are then decoupled, since

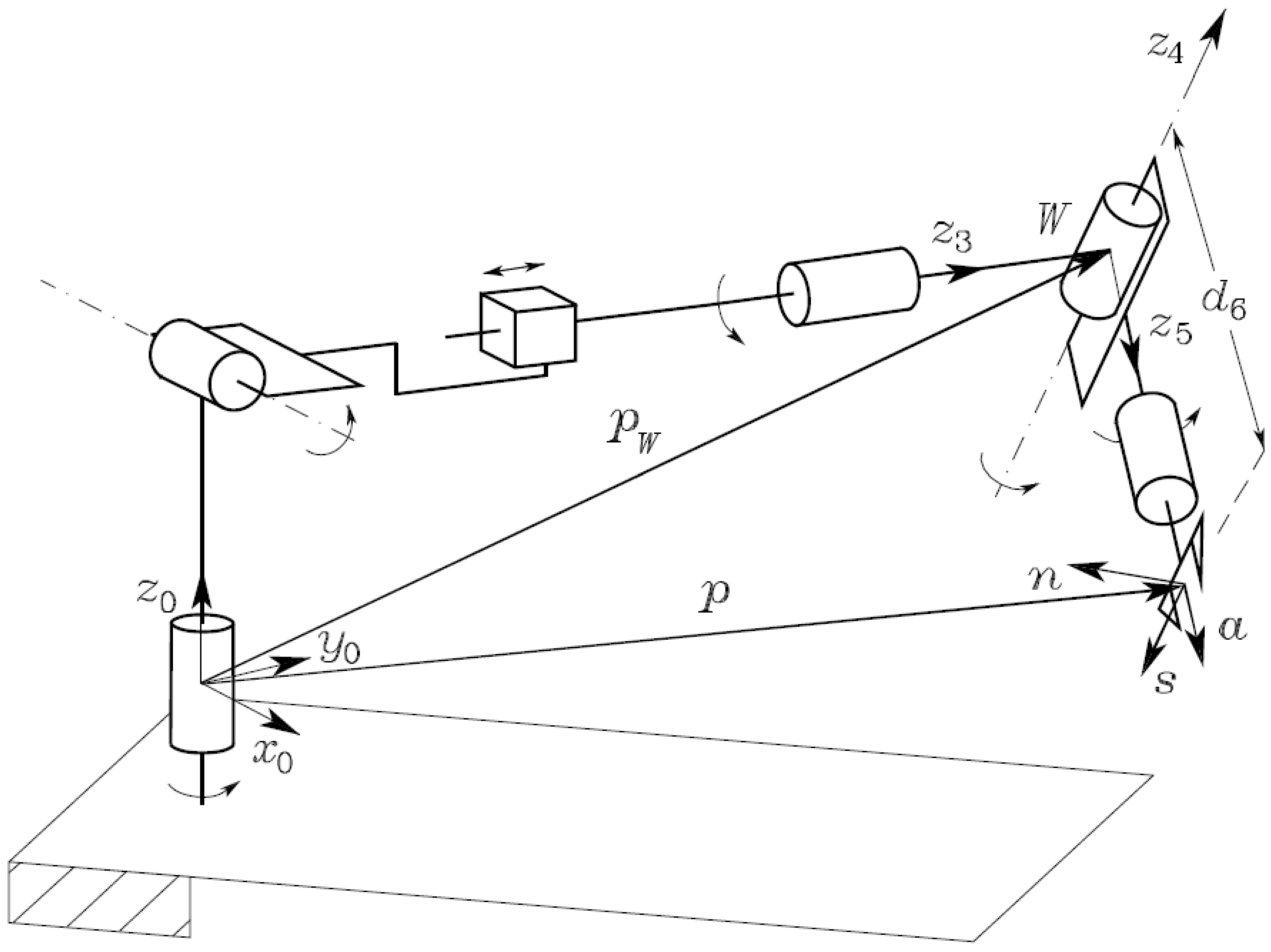
$det(\mathbf{J}_{11}) = 0$ gives the arm singularities

while

$det(\mathbf{J}_{22}) = 0$ gives the wrist singularities

SINGULARITY DECOUPLING-EXAMPLE

- computation of **arm** singularities
- computation of **wrist** singularities



SINGULARITY DECOUPLING

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_{11} & \boldsymbol{J}_{12} \\ \boldsymbol{J}_{21} & \boldsymbol{J}_{22} \end{bmatrix}$$

$$\boldsymbol{J}_{12} = \begin{bmatrix} \boldsymbol{z}_3 \times (\boldsymbol{p} - \boldsymbol{p}_3) & \boldsymbol{z}_4 \times (\boldsymbol{p} - \boldsymbol{p}_4) & \boldsymbol{z}_5 \times (\boldsymbol{p} - \boldsymbol{p}_5) \end{bmatrix}$$

$$\boldsymbol{J}_{22} = \begin{bmatrix} \boldsymbol{z}_3 & \boldsymbol{z}_4 & \boldsymbol{z}_5 \end{bmatrix}$$

- $\boldsymbol{p} = \boldsymbol{p}_W \quad \Longrightarrow \quad \boldsymbol{p}_W - \boldsymbol{p}_i$ parallel to $\boldsymbol{z}_i$, $i = 3, 4, 5$

$$\boldsymbol{J}_{12} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

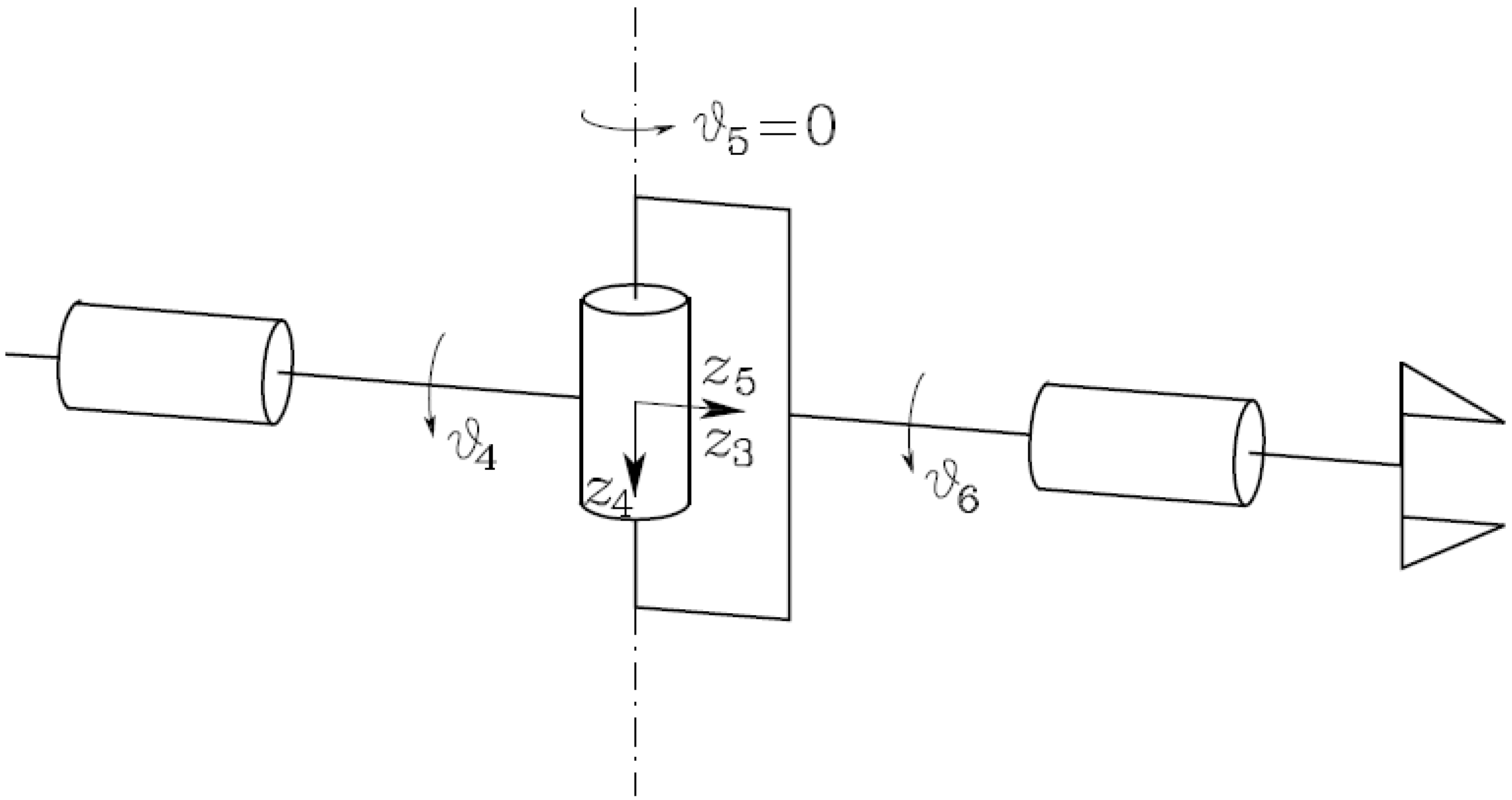
$$\det(\boldsymbol{J}) = \det(\boldsymbol{J}_{11})\det(\boldsymbol{J}_{22})$$

$$\det(\boldsymbol{J}_{11}) = 0 \qquad\qquad \det(\boldsymbol{J}_{22}) = 0$$

Wrist singularities

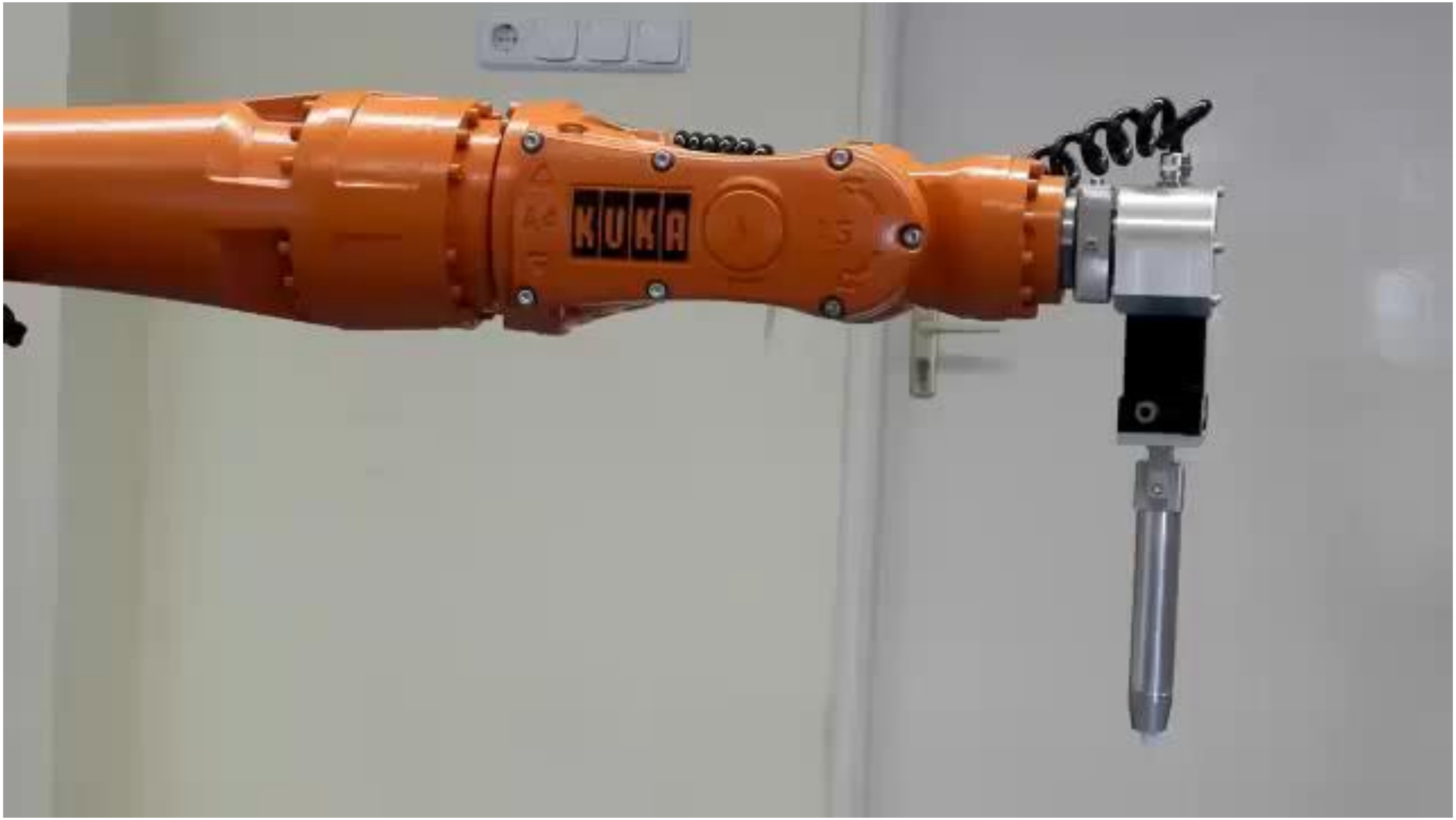
- $\boldsymbol{z}_3$ parallel to $\boldsymbol{z}_5$

$$\vartheta_5 = 0 \qquad \vartheta_5 = \pi$$



Rotations of equal magnitude about opposite directions on ϑ_4 and ϑ_6 do not produce any rotation at the end-effector

Wrist singularities



Arm singularities

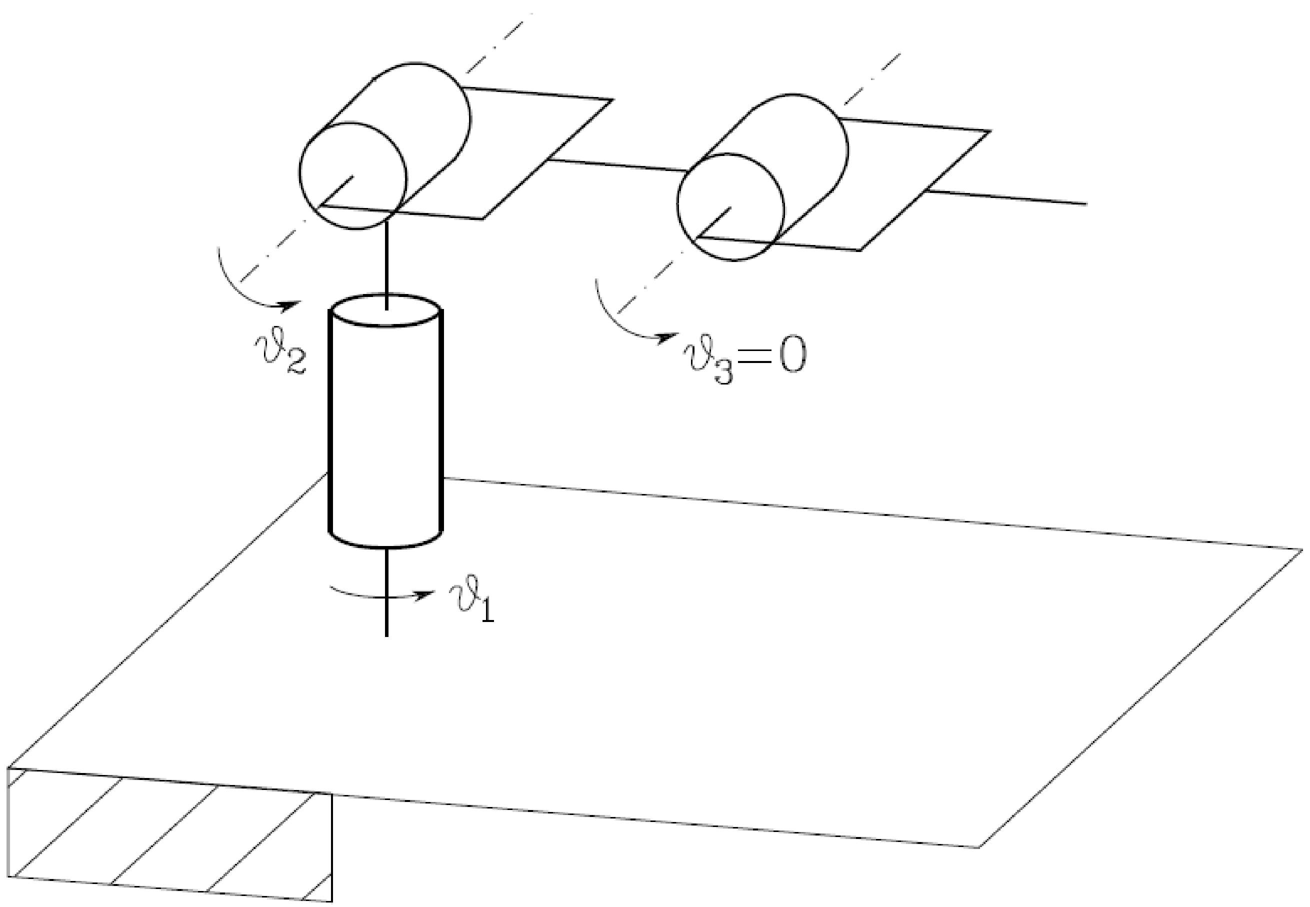
- Anthropomorphic arm

$$\det(\boldsymbol{J}_P) = -a_2 a_3 s_3 (a_2 c_2 + a_3 c_{23})$$

$$s_3 = 0 \qquad\qquad a_2 c_2 + a_3 c_{23} = 0$$

  - *Elbow* singularity

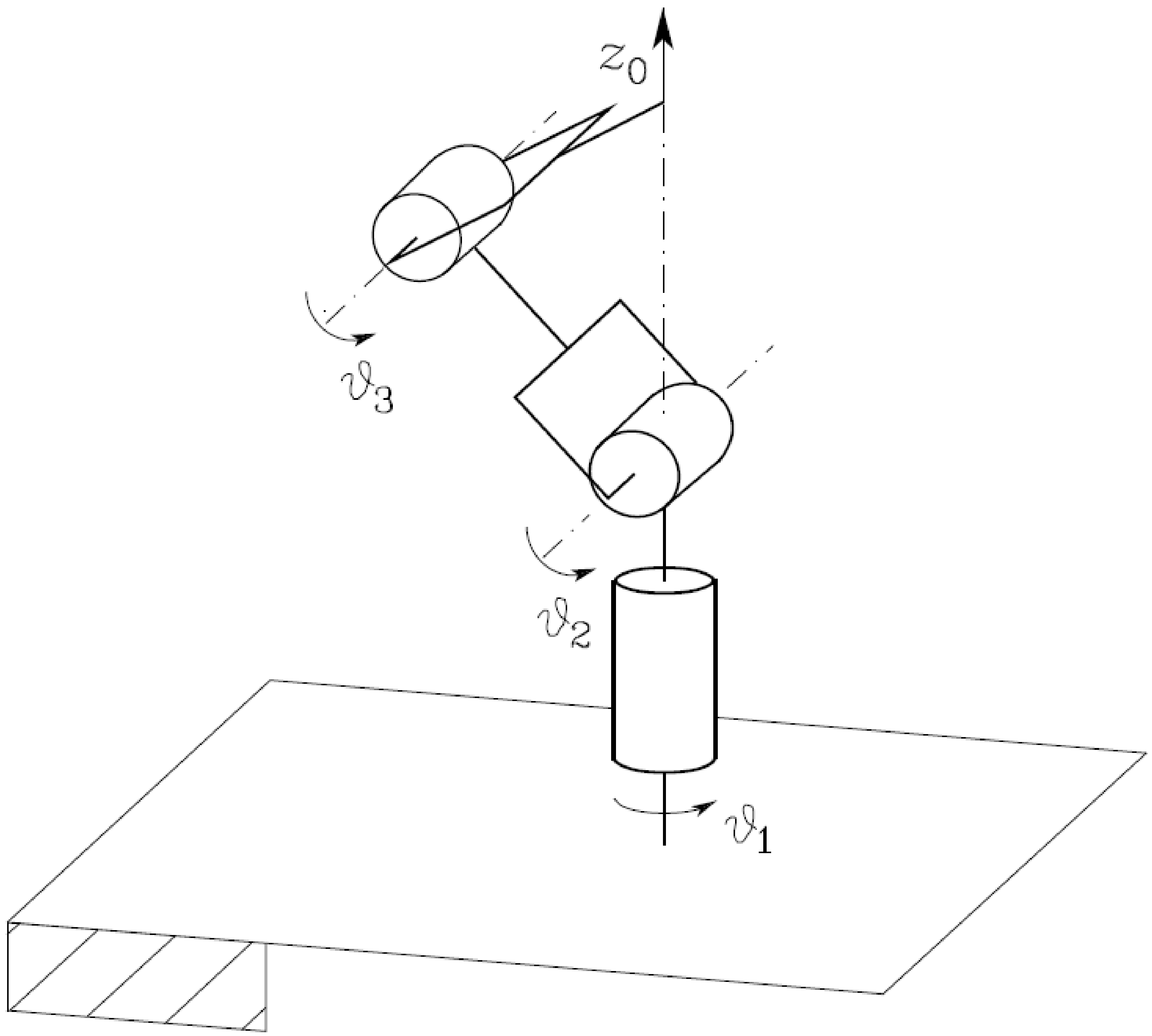
$$\vartheta_3 = 0 \qquad\qquad \vartheta_3 = \pi$$



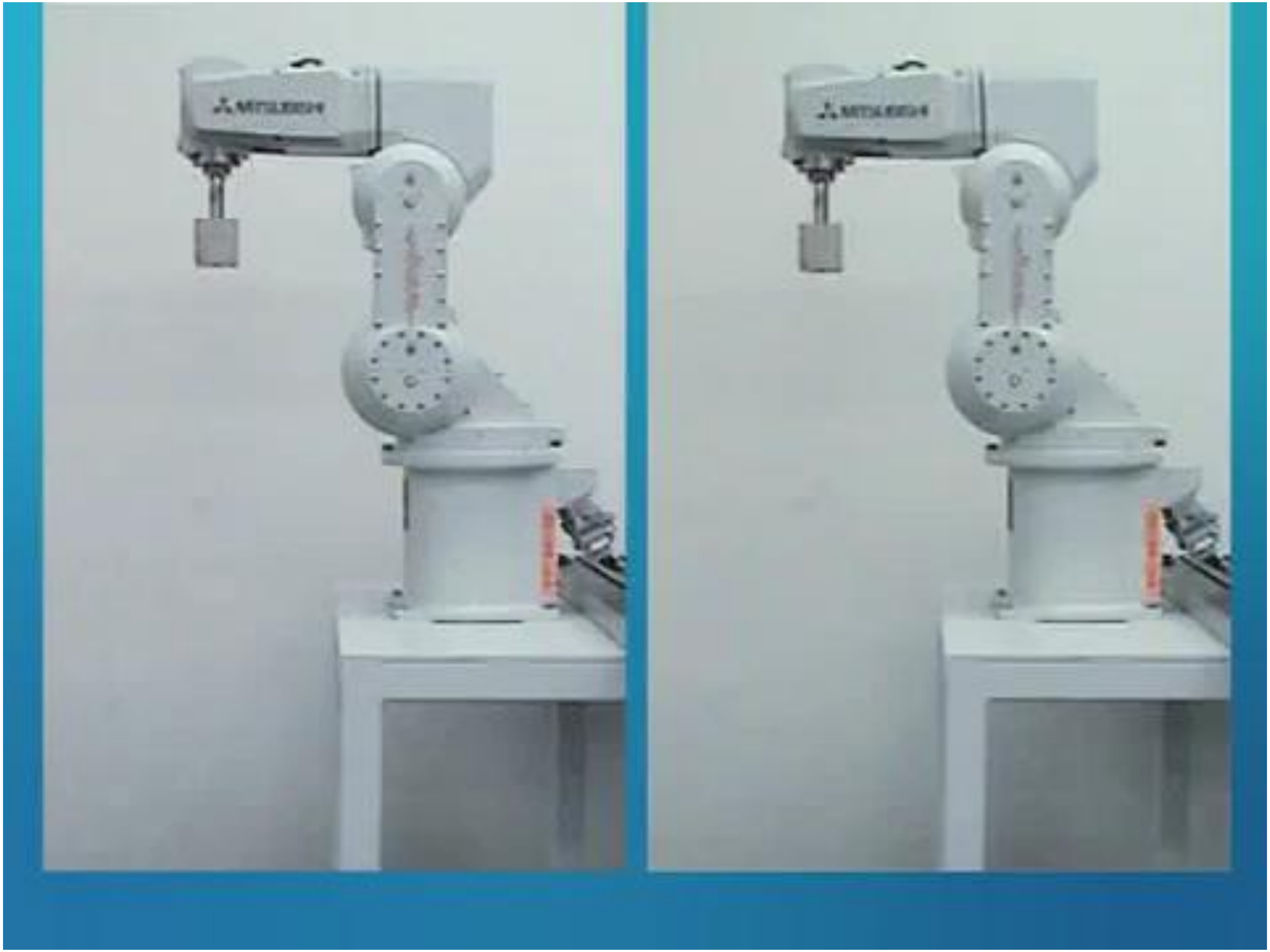
Arm singularities

- *Shoulder* singularity

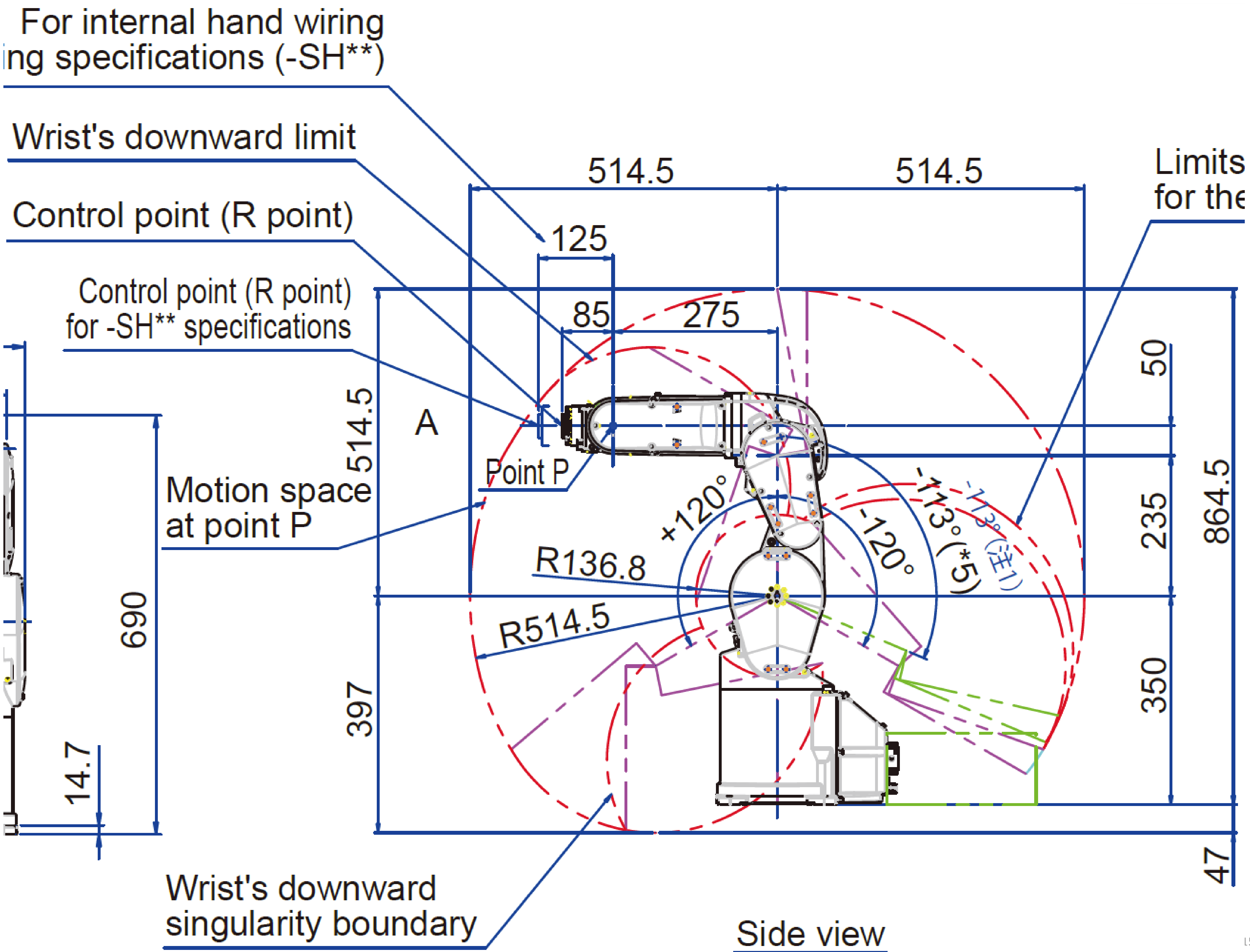
$$p_x = p_y = 0$$



KINEMATIC SINGULARITIES-Movie



KINEMATIC SINGULARITIES



KINEMATIC SINGULARITIES-Movie

KINEMATIC SINGULARITIES-Movie

AVOIDING KINEMATIC SINGULARITIES

Manipulator Performance Constraints in Cartesian Admittance Control for Human-Robot Cooperation

Fotios Dimeas
Charalampos Papakonstantinou
Vassilis C. Moulianitis
Nikos Aspragathos

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

Dept. of Mechanical Engineering & Aeronautics
University of Patras
Greece

ROBOTiCS GROUP
http://robotics.mech.upatras.gr

ICRA 2016
IEEE International Conference on Robotics & Automation
Stockholm, Sweden

INVERSE DIFFERENTIAL KINEMATICS

- Nonlinear kinematics equation
- Differential kinematics equation linear in the velocities
- Given *v*(*t*) + initial conditions ⇒ $(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\boldsymbol{v}$$

$$\boldsymbol{q}(t) = \int_0^t \dot{\boldsymbol{q}}(\varrho)d\varrho + \boldsymbol{q}(0)$$

⋆ (Euler) numerical integration

$$\boldsymbol{q}(t_{k+1}) = \boldsymbol{q}(t_k) + \dot{\boldsymbol{q}}(t_k)\Delta t$$

It is necessary that the Jacobian be square and of full rank

INVERSE DIFFERENTIAL KINEMATICS ALGORITHMS

- Kinematic inversion

$$\boldsymbol{q}(t_{k+1}) = \boldsymbol{q}(t_k) + \boldsymbol{J}^{-1}(\boldsymbol{q}(t_k))\boldsymbol{v}(t_k)\Delta t$$

  - ⋆ *drift of solution*

- Closed-Loop Inverse Kinematics (CLIK) algorithm
  - ⋆ *operational space error*

$$\boldsymbol{e} = \boldsymbol{x}_d - \boldsymbol{x}$$

$$\begin{aligned}\dot{\boldsymbol{e}} &= \dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}} \\ &= \dot{\boldsymbol{x}}_d - \boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}\end{aligned}$$

  - ⋆ find $\quad \dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}(\boldsymbol{e})\text{:} \qquad \boldsymbol{e} \rightarrow \boldsymbol{0}$

Jacobian (pseudo-) inverse

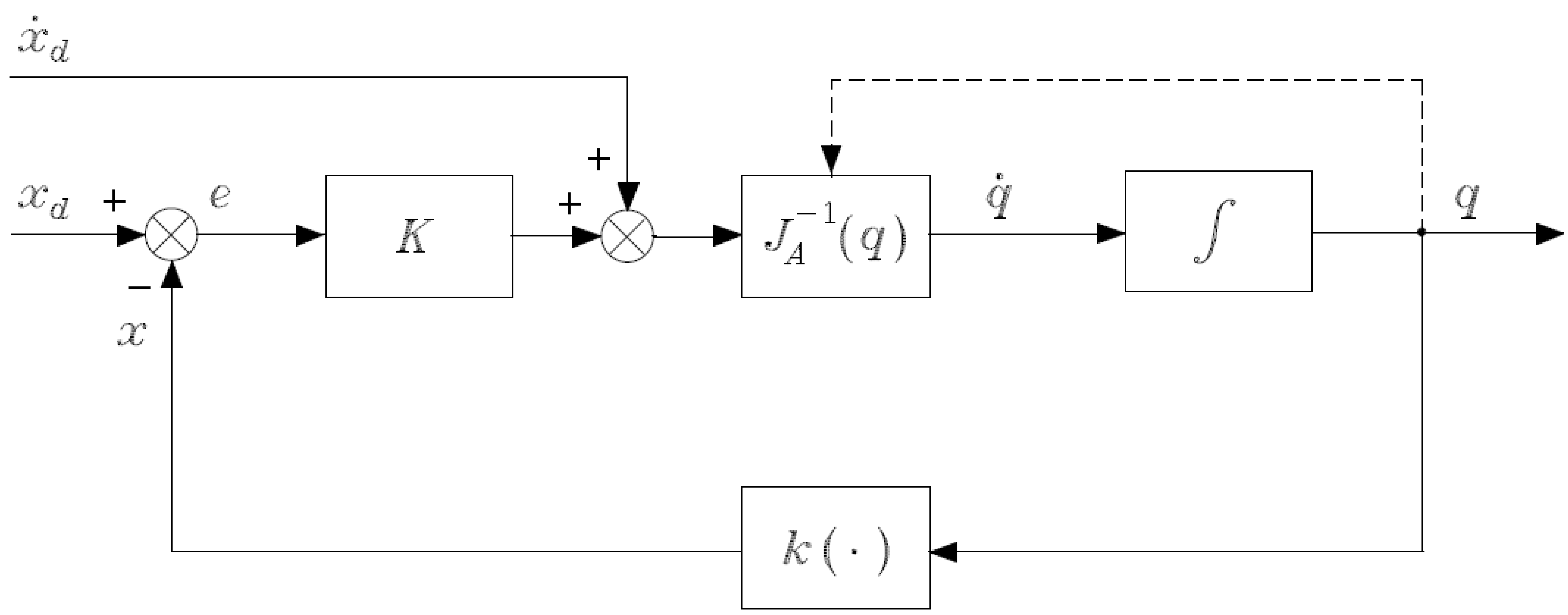
- Linearization of error dynamics

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})(\dot{\boldsymbol{x}}_d + \boldsymbol{K}\boldsymbol{e})$$

$$\Downarrow$$

$$\dot{\boldsymbol{e}} + \boldsymbol{K}\boldsymbol{e} = \boldsymbol{0}$$

The eigenvalues of ***K*** determine stability and speed of convergence



Jacobian transpose

- Solve $\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\boldsymbol{v}_e$ <u>without</u> linearizing error dynamics

- Lyapunov method

$$V(\boldsymbol{e}) = \frac{1}{2}\boldsymbol{e}^T\boldsymbol{K}\boldsymbol{e}$$

where

$$V(\boldsymbol{e}) > 0 \quad \forall \boldsymbol{e} \neq \boldsymbol{0} \qquad\qquad V(\boldsymbol{0}) = 0$$

$$\begin{aligned}\dot{V}(\boldsymbol{e}) &= \boldsymbol{e}^T\boldsymbol{K}\dot{\boldsymbol{x}}_d - \boldsymbol{e}^T\boldsymbol{K}\dot{\boldsymbol{x}} \\ &= \boldsymbol{e}^T\boldsymbol{K}\dot{\boldsymbol{x}}_d - \boldsymbol{e}^T\boldsymbol{K}\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}\end{aligned}$$

⋆ the choice

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}\boldsymbol{e}$$

leads to

$$\dot{V}(\boldsymbol{e}) = \boldsymbol{e}^T\boldsymbol{K}\dot{\boldsymbol{x}}_d - \boldsymbol{e}^T\boldsymbol{K}\boldsymbol{J}_A(\boldsymbol{q})\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}\boldsymbol{e}$$

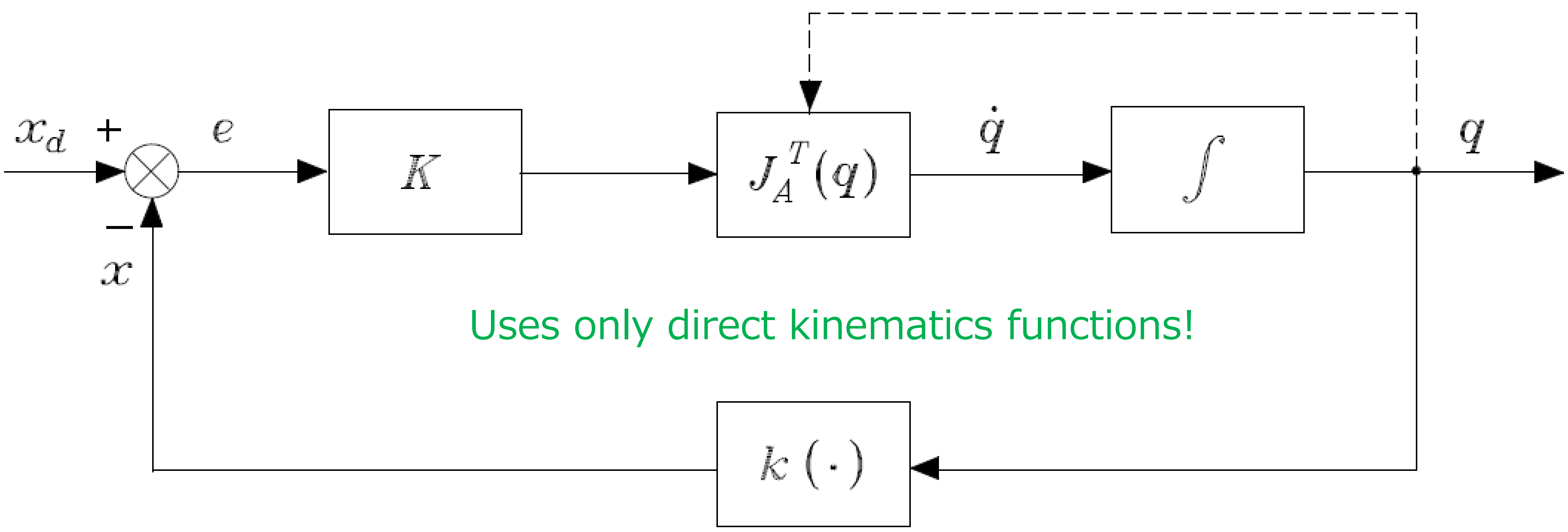
Jacobian transpose

$$\dot{V}(\boldsymbol{e}) = \boldsymbol{e}^T \boldsymbol{K} \dot{\boldsymbol{x}}_d - \boldsymbol{e}^T \boldsymbol{K} \boldsymbol{J}_A(\boldsymbol{q}) \boldsymbol{J}_A^T(\boldsymbol{q}) \boldsymbol{K} \boldsymbol{e}$$

- ⋆ if $\dot{\boldsymbol{x}}_d = \boldsymbol{0} \implies \dot{V} < 0 \text{ with } V > 0$
  (asymptotic stability)

- ⋆ if $\mathcal{N}(\boldsymbol{J}_A^T) \neq \emptyset \implies \dot{V} = 0 \text{ if } \boldsymbol{K}\boldsymbol{e} \in \mathcal{N}(\boldsymbol{J}_A^T)$

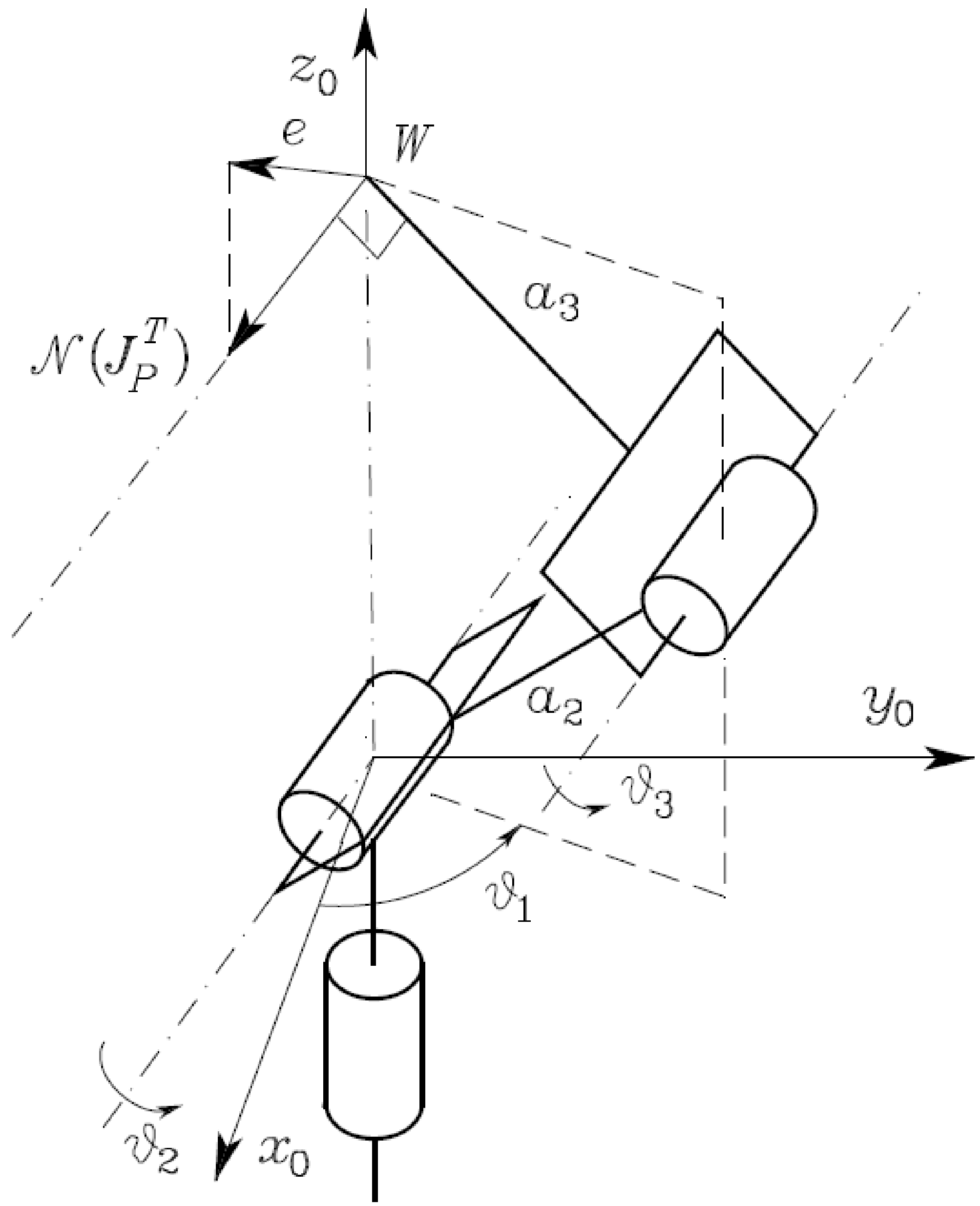
  $\dot{\boldsymbol{q}} = \boldsymbol{0}$ with $\boldsymbol{e} \neq \boldsymbol{0}$ (stuck?)



Uses only direct kinematics functions!

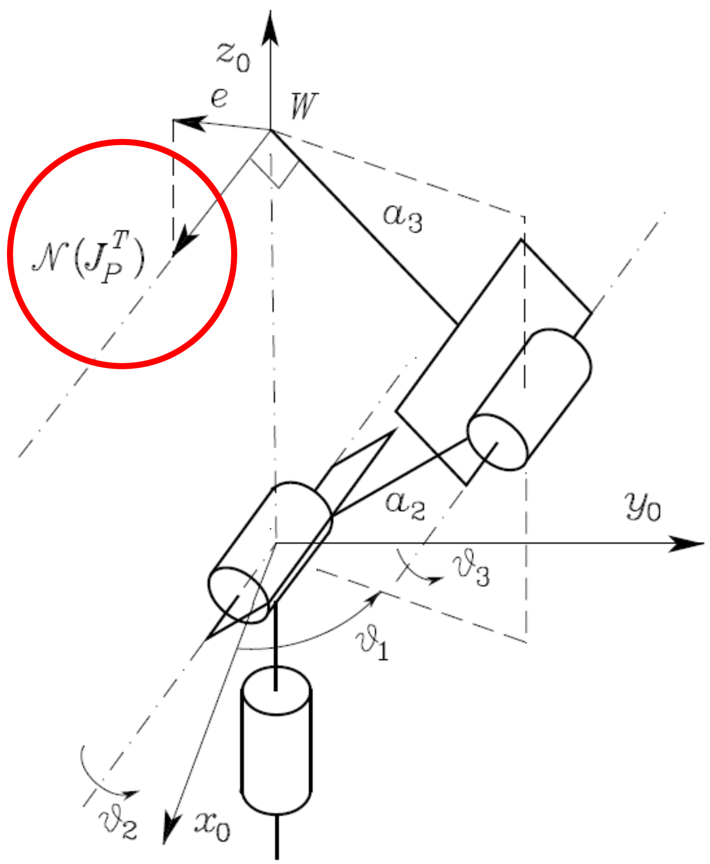
If $\dot{\boldsymbol{x}}_d \neq \boldsymbol{0}$ then **e**(t) bounded and **e** (∞) $\rightarrow$ 0

Example



$$
\boldsymbol{J}_P^T = \begin{bmatrix} 0 & 0 & 0 \\ -c_1(a_2 s_2 + a_3 s_{23}) & -s_1(a_2 s_2 + a_3 s_{23}) & 0 \\ -a_3 c_1 s_{23} & -a_3 s_1 s_{23} & a_3 c_{23} \end{bmatrix}
$$

Example



Compute the null space of $\boldsymbol{J}_P^T$

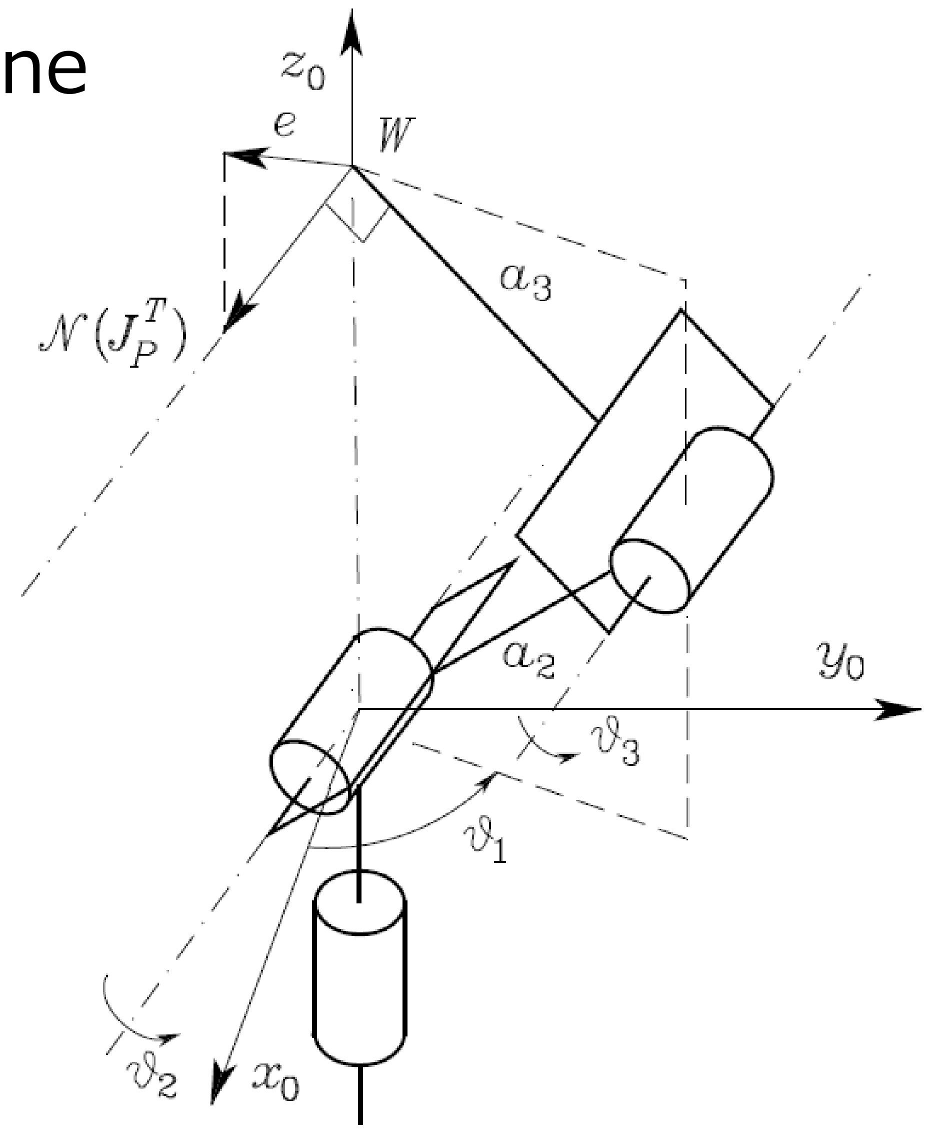
If v_x, v_y and v_z denote the components of vector **v** along the axes of the base frame, we obtain

$$\frac{\nu_y}{\nu_x} = -\frac{1}{\tan\vartheta_1} \qquad \nu_z = 0$$

which means that the direction of $\mathcal{N}(\boldsymbol{J}_P^T)$ coincides with the direction orthogonal to the plane of the structure

Example

- The Jacobian transpose algorithm gets stuck if, with ***K*** diagonal and having all equal elements, the desired position is along the line normal to the plane of the structure at the intersection with the wrist point
- On the other hand, the end-effector cannot physically move from the singular configuration along such a line
- Instead, if the prescribed path has a non-null component in the plane of the structure at the singularity, algorithm convergence is ensured because then $\boldsymbol{K}\boldsymbol{e} \notin \mathcal{N}(\boldsymbol{J}_P^T)$



THAT WAS JUST THE BEGINNING…

- Dynamics
- Control
- Path/Trajectory Planning
- Redundant robots
- Parallel manipulators

Also:

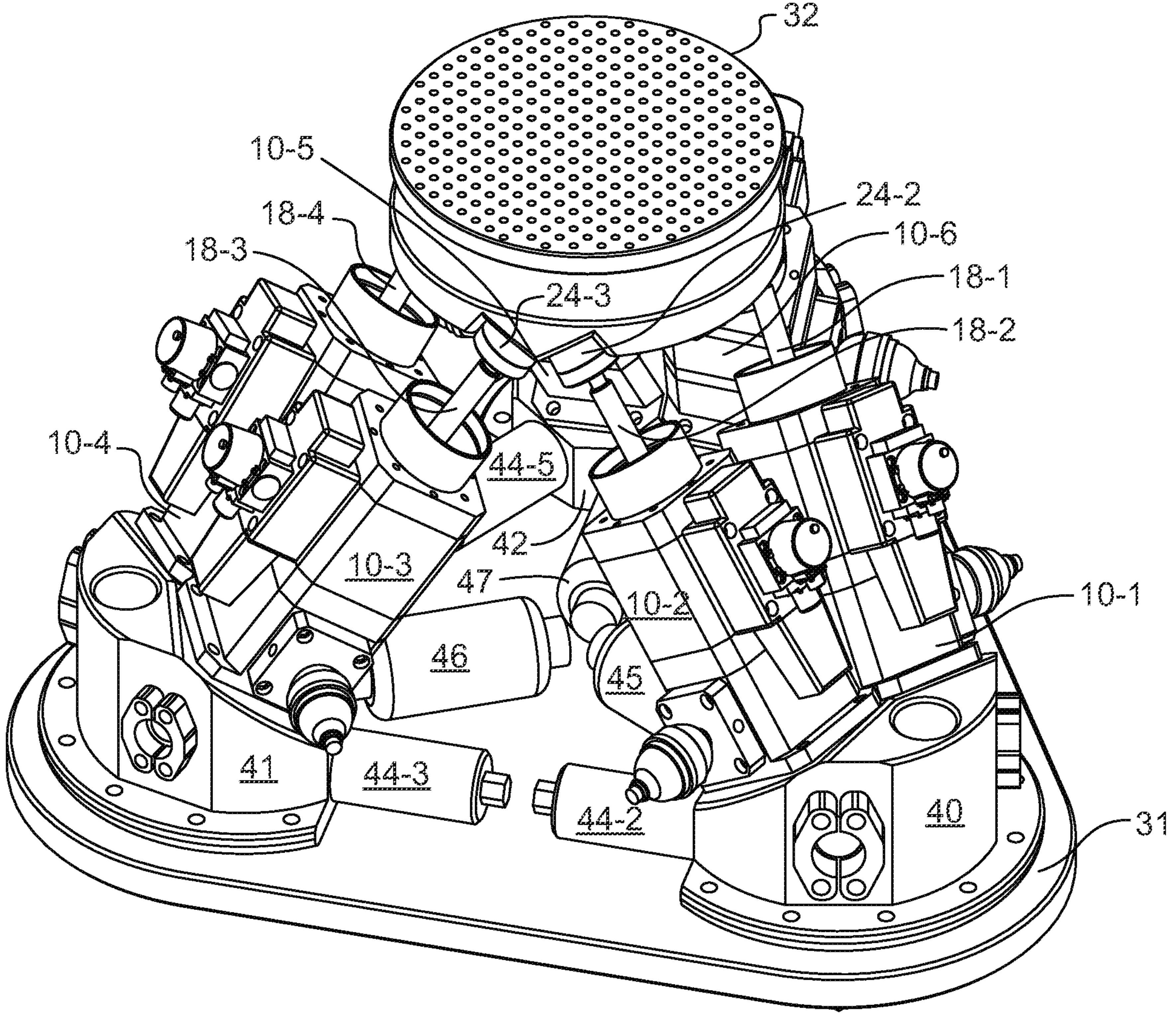
- ➢ Mobile robotics
- ➢ Aerial robotics
- ➢ Underwater robotics
- ➢ Combinations (e.g. a manipulator riding on a mobile)
- ➢ Walking, jumping, exoskeletons/power suits, ++

Samples-Parallel manipulators

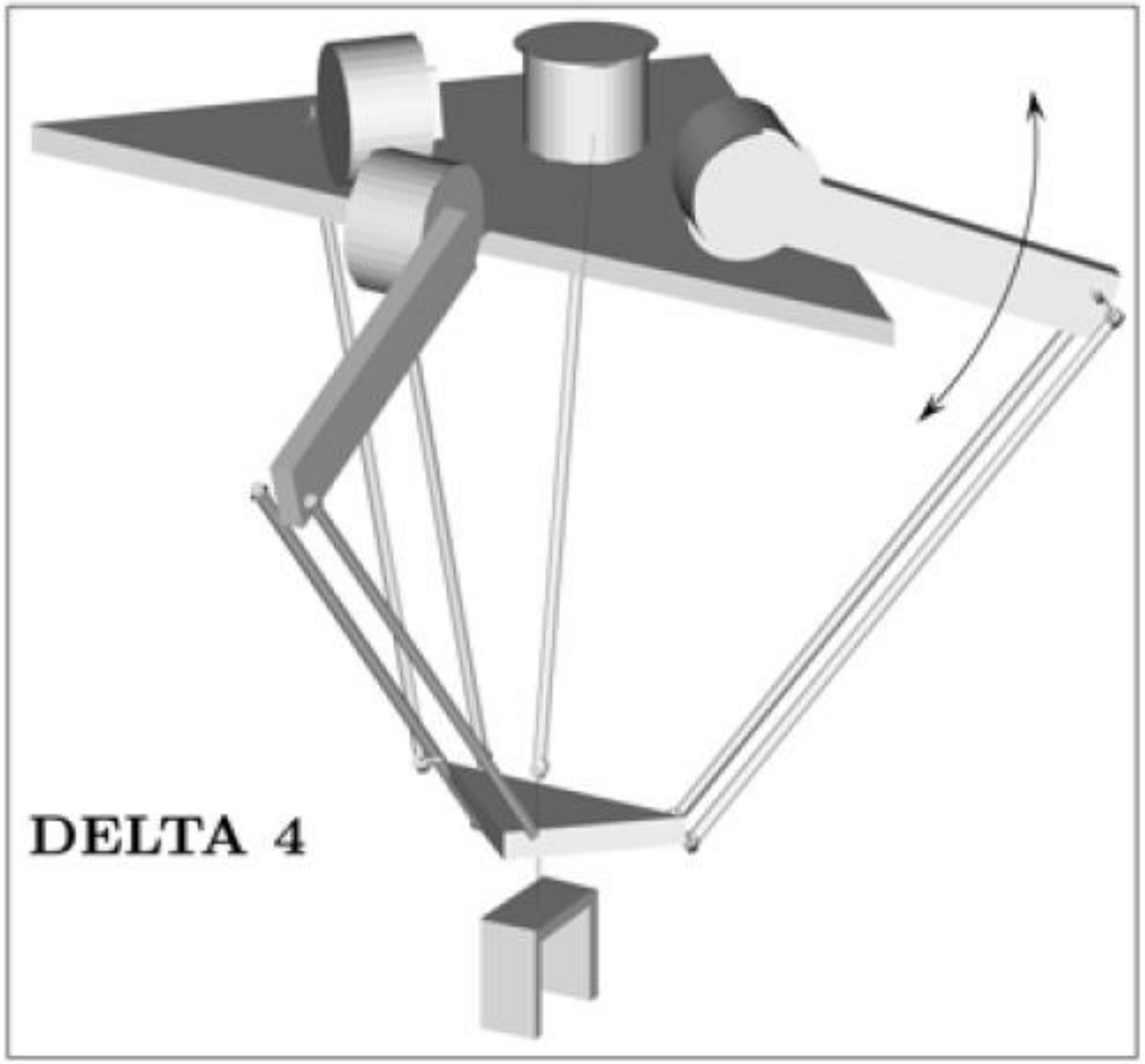
Gough-Stewart Platform Concept

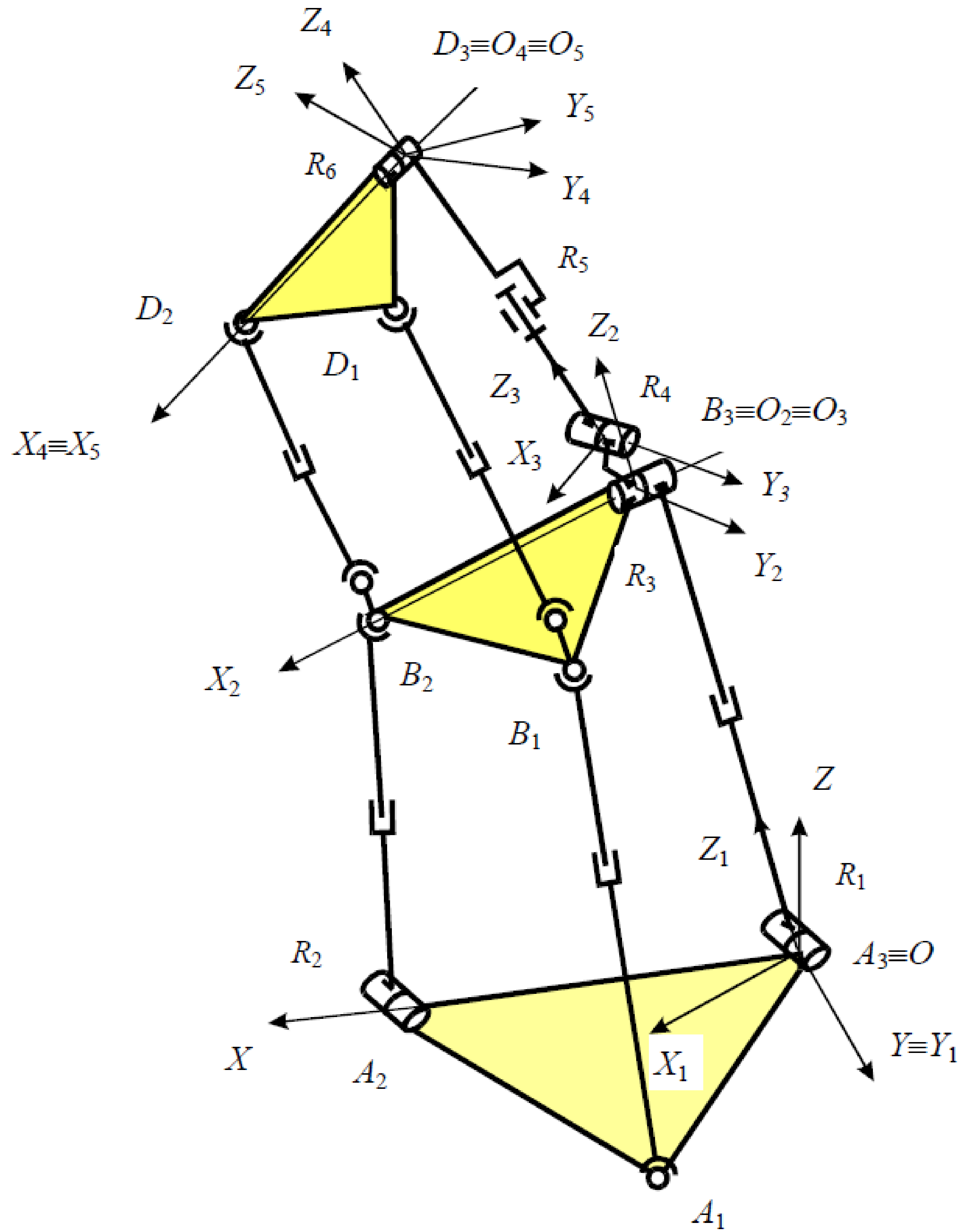
Samples-Parallel manipulators

Gough-Stewart Platform Implemented



Samples-Parallel manipulators





Samples-Parallel manipulators

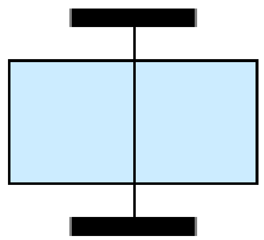
Samples-Parallel manipulators

Samples-Mobile robots

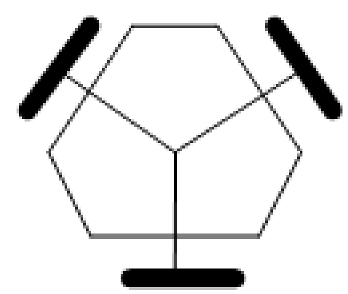
Samples-Mobile robots

Samples-Mobile robots

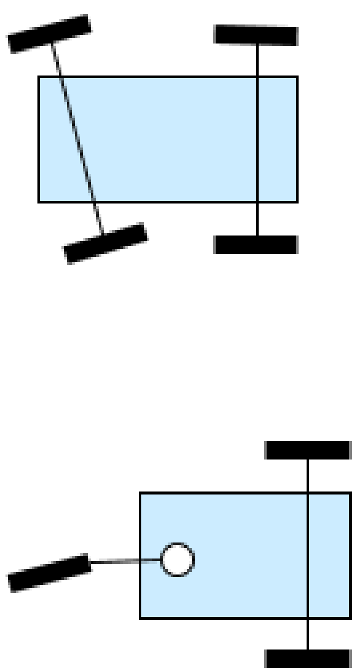
A wheeled mobile robot (WMR) can be driven by wheels in various formations:



Differential

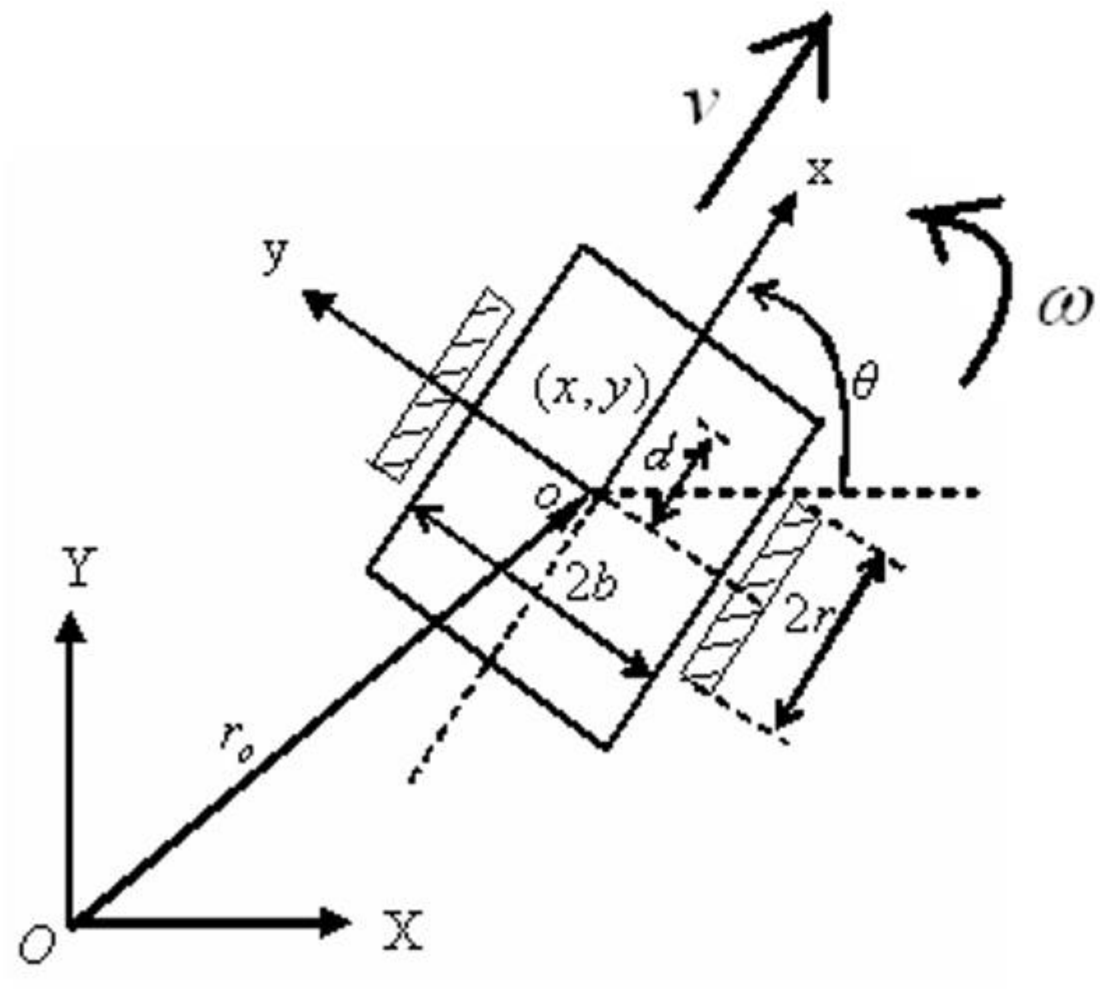


Omni Directional



Steering

Samples-Mobile robots



Differential Wheel Robot

- Kinematic Model

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}$$

- *Nonholonomic Constraint* (rolling contact without slipping)

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0$$

❑ Nonhonolonic (Nonintegrable) and underactuated (2-input~3-output)

❑ **Cannot be stabilized by time-invariant or smooth feedback control**

Samples-Mobile robots

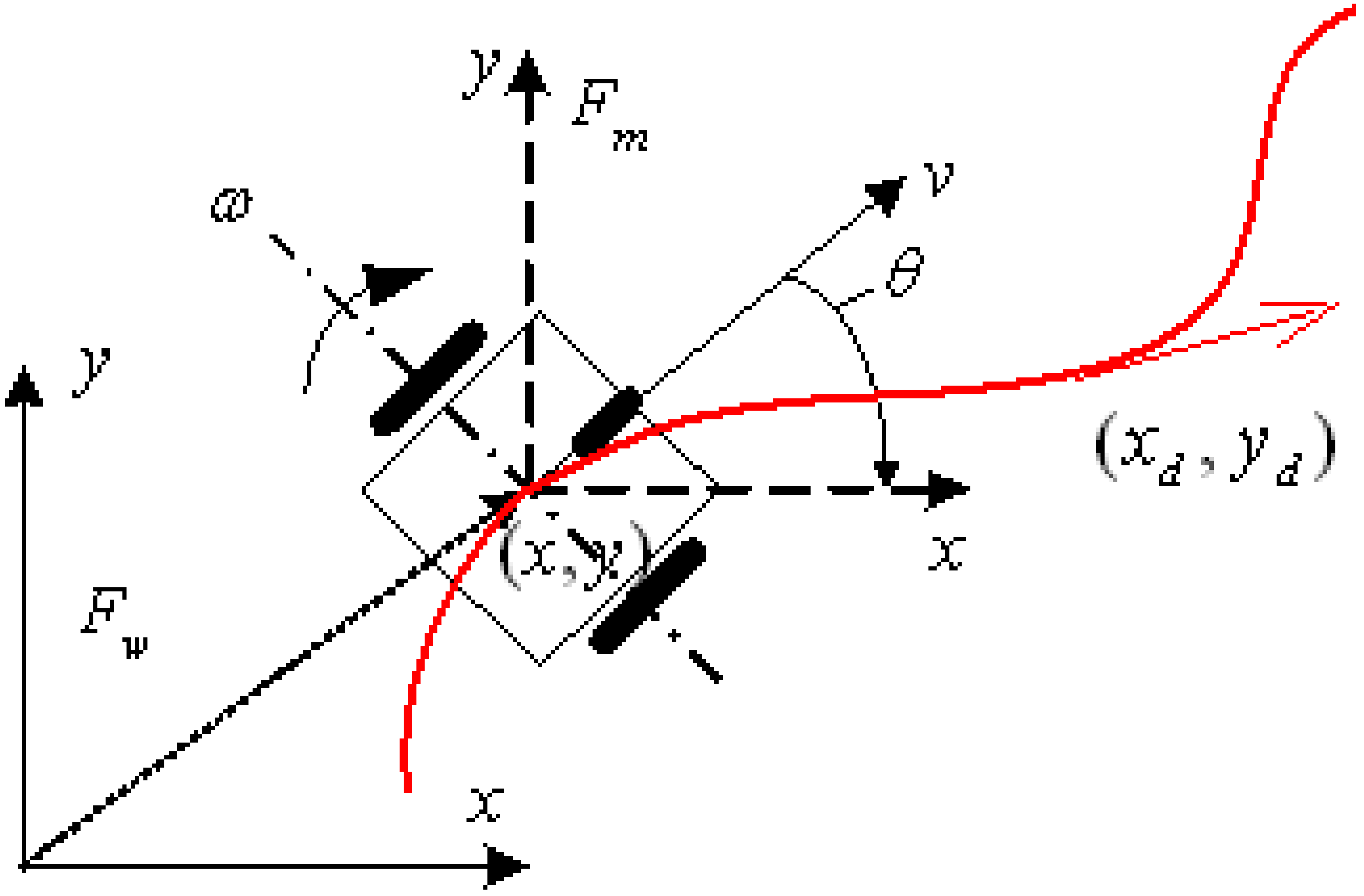
Cannot be stabilized by time-invariant or smooth feedback control

Mobile robots-Trajectory tracking
(Cartesian coordinates based)

Given $x_d, y_d, \dot{x}_d$ and $\dot{y}_d$

find v and ω

to make $x \to x_d, y \to y_d$



Mobile robots-Trajectory tracking
(Cartesian coordinates based)

It can be proved (due to Lyapunov and Barbalat), the following control can meet the objective:

$$v = v_d \cos(\theta_d - \theta) + k_1[\cos\theta(x_d - x) + \sin\theta(y_d - y)]$$

$$\omega = \omega_d + k_2 \,\mathrm{sgn}(v_d)[\sin\theta(x_d - x) - \cos\theta(y_d - y)] + k_3(\theta_d - \theta)$$

$$v_d = \pm\sqrt{\dot{x}_d^2 + \dot{y}_d^2}$$

Desired linear velocity (along the trajectory)

$$\omega_d = \frac{\ddot{y}_d \dot{x}_d - \ddot{x}_d \dot{y}_d}{\dot{x}_d^2 + \dot{y}_d^2}$$

Desired angular velocity

$$\theta_d = ATAN2(\dot{y}_d, \dot{x}_d) + k\pi$$

Desired direction

$$k_1 = k_3 = 2\xi\sqrt{\omega_d^2 + b v_d^2}, \quad k_2 = b|v_d|$$

From the planned trajectory

!The controller fails when $v_d = 0$