SO FAR:

- Direct Kinematics  \( x = f(q) \)
- Inverse Kinematics  \( q = f^{-1}(x) \)
- Jacobian \[
\begin{bmatrix}
\dot{v} \\
\omega
\end{bmatrix} = J \dot{q}
\]

Geometric, Analytic

\[
J_G = \begin{bmatrix}
1 & 0 \\
0 & T(\gamma)
\end{bmatrix} J_A
\]
KINEMATIC SINGULARITIES

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

$$\dot{x} = J(q)\dot{q} \implies dx = J(q)\,dq$$

- So, this can be interpreted as a relationship between infinitesimal displacements in $\mathbb{R}^n$ and $\mathbb{R}^6$
- In general, $\text{rank}(J(q)) = \min (6, n)$
- 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 $\dot{x}$) in $\mathbb{R}^6$ without any correspondent “direction” ($\dot{q}$) 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

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

\[ \det(J) = a_1 a_2 s_2 \]

\[ \begin{align*}
\dot{\theta}_2 &= 0 \\
\ddot{\theta}_2 &= \pi
\end{align*} \]

\[ \begin{bmatrix} -(a_1 + a_2) s_1 & (a_1 + a_2) c_1 \end{bmatrix}^T \text{ parallel to } \begin{bmatrix} -a_2 s_1 & a_2 c_1 \end{bmatrix}^T \]

(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 \textbf{arm} singularities
- computation of \textbf{wrist} singularities

If \( J \in \mathbb{R}^{6 \times n} \) then

\[
J = \begin{bmatrix}
J_{11} & J_{12} \\
J_{21} & J_{22}
\end{bmatrix}
\]

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

\[
J_{12} = [z_3 \times (p_6 - p_3), \ z_4 \times (p_6 - p_4), \ z_5 \times (p_6 - p_5)]
\]

\[
J_{22} = [z_3, \ z_4, \ z_5]
\]
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
\[ J_{12} = [0, 0, 0] \implies J = \begin{bmatrix} J_{11} & 0 \\ J_{21} & J_{22} \end{bmatrix} \]

In this manner, \( J \) is a block lower-triangular matrix, and
\[ \det(J) = \det(J_{11}) \det(J_{22}) \]

The singularities are then decoupled, since
\[ \det(J_{11}) = 0 \] gives the arm singularities
while
\[ \det(J_{22}) = 0 \] gives the wrist singularities.
• computation of **arm** singularities
• computation of **wrist** singularities
SINGULARITY DECOUPLING

\[ J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \]

\[ J_{12} = \begin{bmatrix} z_3 \times (p - p_3) & z_4 \times (p - p_4) & z_5 \times (p - p_5) \end{bmatrix} \]

\[ J_{22} = \begin{bmatrix} z_3 & z_4 & z_5 \end{bmatrix} \]

- \[ p = p_W \implies p_W - p_i \text{ parallel to } z_i, \ i = 3, 4, 5 \]

\[ J_{12} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \]

\[ \det(J) = \det(J_{11})\det(J_{22}) \]

\[ \det(J_{11}) = 0 \quad \det(J_{22}) = 0 \]
Wrist singularities

- $z_3$ parallel to $z_5$

\[ \theta_5 = 0 \quad \theta_5 = \pi \]

Rotations of equal magnitude about opposite directions on $\theta_4$ and $\theta_6$ do not produce any rotation at the end-effector.
Wrist singularities
Arm singularities

• Anthropomorphic arm

$$\det(J_P) = -a_2a_3s_3(a_2c_2 + a_3c_{23})$$

$$s_3 = 0 \quad a_2c_2 + a_3c_{23} = 0$$

* Elbow singularity

$$\vartheta_3 = 0 \quad \vartheta_3 = \pi$$
Arm singularities

* Shoulder singularity

\[ p_x = p_y = 0 \]
KINEMATIC SINGULARITIES-Movie
For internal hand wiring specifications (-SH**) 

Wrist's downward limit 

Control point (R point) 

Control point (R point) for -SH** specifications 

Motion space at point P 

Wrist's downward singularity boundary 

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

Fotios Dimeas
Charalampos Papakonstantinou
Vassilis C. Moulianitis
Nikos Aspragathos

Dept. of Mechanical Engineering & Aeronautics
University of Patras
Greece

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

http://robotics.mech.upatras.gr
INVERSE DIFFERENTIAL KINEMATICS

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

\[ \dot{q} = J^{-1}(q) \nu \]

\[ q(t) = \int_{0}^{t} \dot{q}(\xi) d\xi + q(0) \]

* (Euler) numerical integration

\[ q(t_{k+1}) = q(t_k) + \dot{q}(t_k) \Delta t \]

It is necessary that the Jacobian be square and of full rank
INVERSE DIFFERENTIAL KINEMATICS ALGORITHMS

• Kinematic inversion

\[ q(t_{k+1}) = q(t_k) + J^{-1}(q(t_k)) \nu(t_k) \Delta t \]

* drift of solution

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

\[ e = x_d - x \]

\[ \dot{e} = \dot{x}_d - \dot{x} \]

\[ = \dot{x}_d - J_A(q) \dot{q} \]

* find \( \dot{q} = \dot{q}(e) : \quad e \to 0 \)
Jacobian (pseudo-) inverse

- Linearization of error dynamics

\[ \dot{q} = J^{-1}_A(q)(\dot{x}_d + Ke) \]

\[ \dot{e} + Ke = 0 \]

The eigenvalues of $K$ determine stability and speed of convergence
Jacobian transpose

• Solve $\dot{q} = J^{-1}(q) v_e$ without linearizing error dynamics

• Lyapunov method

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

where

$$V(e) > 0 \quad \forall e \neq 0 \quad V(0) = 0$$

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

$$= e^T K \dot{x}_d - e^T K J_A(q) \dot{q}$$

* the choice

$$\dot{q} = J_A^T(q) K e$$

leads to

$$\dot{V}(e) = e^T K \dot{x}_d - e^T K J_A(q) J_A^T(q) K e$$
\[ \dot{V}(e) = e^T K \dot{x}_d - e^T K J_A(q) J_A^T(q) K e \]

* if \( \dot{x}_d = 0 \) \( \implies \) \( \dot{V} < 0 \) with \( V > 0 \) (asymptotic stability)

* if \( \mathcal{N}(J_A^T) \neq \emptyset \) \( \implies \) \( \dot{V} = 0 \) if \( K e \in \mathcal{N}(J_A^T) \)

\[ \dot{q} = 0 \text{ with } e \neq 0 \text{ (stuck?)} \]

If \( \dot{x}_d \neq 0 \) then \( e(t) \) bounded and \( e(\infty) \rightarrow 0 \)

Uses only direct kinematics functions!
Example

\[
J_P^T = \begin{bmatrix}
0 & 0 & 0 \\
-c_1(a_2s_2 + a_3s_23) & -s_1(a_2s_2 + a_3s_23) & 0 \\
-a_3c_1s_23 & -a_3s_1s_23 & a_3c_23
\end{bmatrix}
\]
Example

Compute the null space of $J_P^T$

If $\nu_x$, $\nu_y$ and $\nu_z$ denote the components of vector $\mathbf{v}$ along the axes of the base frame, we obtain

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

which means that the direction of $\mathbf{v}$ 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 $K e \notin \mathcal{N}(J_P^T)$. 
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
This is how robots help sort parcels for quicker delivery at a Chinese firm
Samples-Mobile robots

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

Differential  Omni Directional  Steering
Samples-Mobile robots

- 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)

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

Differential Wheel Robot

- Nonholonomic (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 \( \omega \)

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:

\[ v = v_d \cos(\theta_d - \theta) + k_1[\cos\theta(x_d - x) + \sin \theta(y_d - y)] \]
\[ \omega = \omega_d + k_2 \text{sgn}(v_d)\left[\sin \theta(x_d - x) - \cos \theta(y_d - y)\right] + k_3(\theta_d - \theta) \]

\[ v_d = \pm \sqrt{\dot{x}_d^2 + \dot{y}_d^2} \] Desired linear velocity (along the trajectory)

\[ \omega_d = \frac{\ddot{y}_d \dot{x}_d - \dot{y}_d \ddot{x}_d}{\dot{x}_d^2 + \dot{y}_d^2} \] Desired angular velocity

\[ \theta_d = \text{ATAN2}(\dot{y}_d, \dot{x}_d) + k\pi \] Desired direction

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

!The controller fails when \( v_d = 0 \)