Numerical approaches to singularity analysis, singularity index and new concepts

J-P. Merlet

INRIA SOPHIA ANTIPOLIS

You have already seen:

You have already seen:

- an algebraic approach to singularity analysis

You have already seen:

- an algebraic approach to singularity analysis
- a geometrical approach to singularity analysis

INRIA SOPHIA ANTIPOLIS

What you will see now:

What you will see now:

- a numerical approach of singularity analysis

What you will see now:

- a numerical approach of singularity analysis
- what should **not** be used as "distance" to singularity

What you will see now:

- a numerical approach of singularity analysis
- what should **not** be used as "distance" to singularity
- another approach to singularity analysis

want to see a motion from one solution of the forward kinematics to another one without singularity crossing ?

INRIA
SOPHIA ANTIPOLIS

Interval analysis

Interval analysis

Interval $\mathcal{X} = [\underline{x}, \overline{x}]$, width $w(\mathcal{X}) = \overline{x} - \underline{x}$

Interval analysis

Interval $\mathcal{X} = [\underline{x}, \overline{x}]$, width $w(\mathcal{X}) = \overline{x} - \underline{x}$

- a function $f(x)$
- **interval evaluation** of f when $x \in \mathcal{X}$: a range $[\underline{F}, \overline{F}]$ such that

$$\forall\, x \;\in\; \mathcal{X} \;\text{ we have }\; : \underline{F} \leq f(x) \leq \overline{F}$$

Interval analysis

How to construct an interval evaluation ? the simplest one is the **natural evaluation**:

substitute each mathematical operator by its interval equivalent

Interval analysis

Example: $F = x^2 + \cos(x)$, $x \in [0, 1]$

Problem: find $[A, B]$ such that: $A \leq F(x) \leq B \;\forall\, x \in [0, 1]$

INRIA SOPHIA ANTIPOLIS

Interval analysis

$$F = [0,1]^2 + \cos([0,1])$$

Interval analysis

$$F = [0, 1]^2 + \cos([0, 1])$$

Interval analysis

$$F = [0,1]^2 + \cos([0,1]) = [0,1] + \cos([0,1])$$

Interval analysis

$$F = [0,1]^2 + \cos([0,1]) = [0,1] + \cos([0,1])$$

INRIA SOPHIA ANTIPOLIS

Interval analysis

$$F = [0,1]^2 + \cos([0,1]) = [0,1] + [0.54, 1]$$

Interval analysis

$$F = [0,1]^2 + \cos([0,1]) = \text{[0,1]+[0.54,1]}$$

Interval analysis

$$F = [0,1]^2 + \cos([0,1]) = \text{[0,1]+[0.54,1]} = [\mathbf{0.54}, \mathbf{2}]$$

Interval analysis

$$F = [0,1]^2 + \cos([0,1]) = \text{[0,1]+[0.54,1]} = [\mathbf{0.54}, \mathbf{2}]$$

- 0 not included in [0.54,2] $\Rightarrow F \neq 0 \; \forall \, x \in [0,1]$

Interval analysis

$$F = \ [0,1]^2 \ + \ \cos([0,1]) \ \ = \text{[0,1]+[0.54,1]} = \mathbf{[0.54, 2]}$$

- 0 not included in [0.54,2] $\Rightarrow \ F \neq 0 \ \forall \ x \in [0,1]$
- $F > 0 \ \ \forall \ x \in [0,1]$
- $\forall \ x \in [0,1]$ we have $0.54 \leq F \leq 2$ (global optimization)

INRIA SOPHIA ANTIPOLIS

Interval analysis

$$F = [0,1]^2 + \cos([0,1]) = \text{[0,1]+[0.54,1]} = [\mathbf{0.54}, \mathbf{2}]$$

We have calculated an interval evaluation of F for the range [0,1]

Properties

Properties

- **can be implemented to take into account round-off errors**

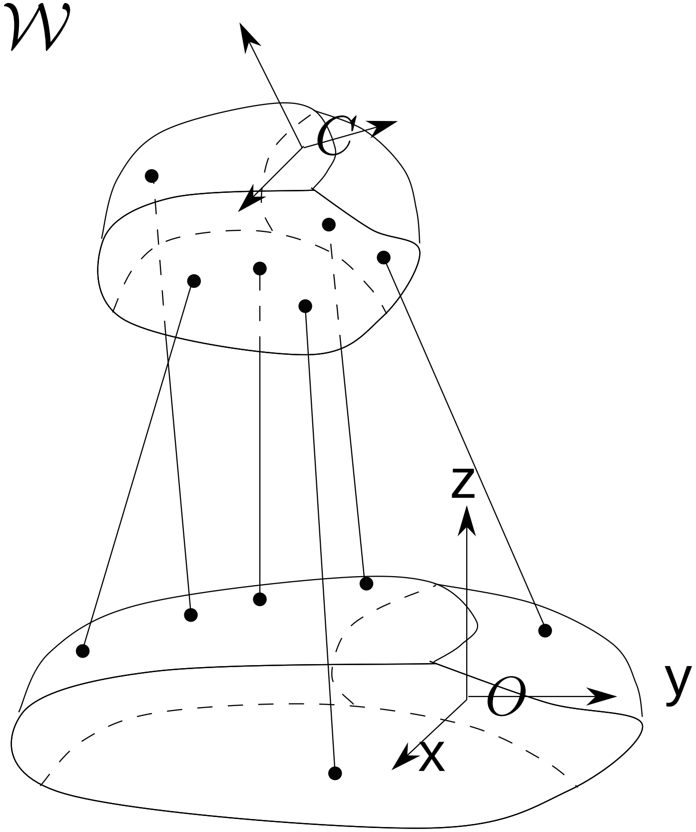
INRIA
SOPHIA ANTIPOLIS

Properties

- **can be implemented to take into account round-off errors**
- **Overestimation** of $\underline{F}, \overline{F}$ but decreases with the size of the ranges

Example: singularity checking

A crucial problem for parallel robot: no singularity in a given workspace $\mathcal{W}$



Example: singularity checking

- τ: force/torque in the actuated joints
- $\mathcal{F}$: external forces/torques applied on the platform

Example: singularity checking

- τ: force/torque in the actuated joints
- $\mathcal{F}$: external forces/torques applied on the platform

Mechanical equilibrium: $\mathcal{F} = \mathbf{J}(\mathbf{X}, \mathbf{P})\tau$

INRIA SOPHIA ANTIPOLIS

Example: singularity checking

- τ: force/torque in the actuated joints
- $\mathcal{F}$: external forces/torques applied on the platform

Mechanical equilibrium: $\mathcal{F} = \mathbf{J}(\mathbf{X}, \mathbf{P})\tau$

linear system in τ

$$\tau_i = \frac{|\ldots|}{|\mathbf{J}|}$$

if $|\mathbf{J}| \longrightarrow 0$ possibly $\tau_i \longrightarrow \infty$

Example: singularity checking

Checking if a workspace $\mathcal{W}$ is singularity free:

$$\forall \mathbf{X} \in \mathcal{W} \quad |\mathbf{J}(\mathbf{X},\mathbf{P}))| \neq 0$$

Example: singularity checking

Checking if a workspace $\mathcal{W}$ is singularity free:

$$\forall \mathbf{X} \in \mathcal{W} \quad |\mathbf{J}(\mathbf{X}, \mathbf{P}))| \neq 0$$

- consider an arbitrary point $\mathbf{M_1}$ in $\mathcal{W}$

Example: singularity checking

Checking if a workspace $\mathcal{W}$ is singularity free:

$$\forall \mathbf{X} \in \mathcal{W} \quad |\mathbf{J}(\mathbf{X}, \mathbf{P}))| \neq 0$$

- consider an arbitrary point $\mathbf{M_1}$ in $\mathcal{W}$
- compute the sign of $|\mathbf{J}(\mathbf{M_1})|$, say > 0 (use IA to certify)

Example: singularity checking

Checking if a workspace $\mathcal{W}$ is singularity free:

$$\forall \mathbf{X} \in \mathcal{W} \quad |\mathbf{J}(\mathbf{X}, \mathbf{P}))| \neq 0$$

- consider an arbitrary point $\mathbf{M_1}$ in $\mathcal{W}$
- compute the sign of $|\mathbf{J}(\mathbf{M_1})|$, say > 0 (use IA to certify)
- if we find a pose $\mathbf{M_2}$ with $|\mathbf{J}(\mathbf{M_2})| < 0$

Example: singularity checking

Checking if a workspace $\mathcal{W}$ is singularity free:

$$\forall \mathbf{X} \in \mathcal{W} \quad |\mathbf{J}(\mathbf{X}, \mathbf{P}))| \neq 0$$

- consider an arbitrary point $\mathbf{M_1}$ in $\mathcal{W}$
- compute the sign of $|\mathbf{J}(\mathbf{M_1})|$, say > 0 (use IA to certify)
- if we find a pose $\mathbf{M_2}$ with $|\mathbf{J}(\mathbf{M_2})| < 0$

$$\Downarrow$$

then any path from $\mathbf{M_1}$ to $\mathbf{M_2}$ crosses a singularity

Example: singularity checking

Checking if a workspace $\mathcal{W}$ is singularity free:

$$\forall \mathbf{X} \in \mathcal{W} \quad |\mathbf{J}(\mathbf{X}, \mathbf{P}))| \neq 0$$

- consider an arbitrary point $\mathbf{M_1}$ in $\mathcal{W}$
- compute the sign of $|\mathbf{J}(\mathbf{M_1})|$, say > 0 (use IA to certify)
- if we find a pose $\mathbf{M_2}$ with $|\mathbf{J}(\mathbf{M_2})| < 0$

$$\Downarrow$$

$\mathcal{W}$ includes a singularity

Singularity checking with IA

Singularity checking with IA

Objective: find a box $\mathcal{B}$ for which $|\mathbf{J}(\mathcal{B})| < 0$

Singularity checking with IA

Objective: find a box $\mathcal{B}$ for which $|\mathbf{J}(\mathcal{B})| < 0$

Evaluation function: compute the sign of $|\mathbf{J}(\mathcal{B})|$

Singularity checking with IA

Objective: find a box $\mathcal{B}$ for which $|\mathbf{J}(\mathcal{B})| < 0$

Evaluation function: compute the sign of $|\mathbf{J}(\mathcal{B})|$

- minor expansion
- Gaussian elimination

Singularity checking with IA

Objective: find a box $\mathcal{B}$ for which $|\mathbf{J}(\mathcal{B})| < 0$

Singularity checking with IA

Ingredients of the algorithm:

- a *box*: a set of ranges for $x, y, z, \psi, \theta, \phi$
- a *list* $\mathcal{S} = \{B_1, B_2, \ldots\}$ of boxes. Initially $\mathcal{S} = \{\mathcal{W}\}$ but boxes will be added
- an *index* i to indicate which box in $\mathcal{S}$ is processed

Singularity checking with IA

Algorithm

1. $F = [\underline{F}, \overline{F}]$=interval evaluation of $|\mathbf{J}(B_i)|$

Singularity checking with IA

Algorithm

1. $F = [\underline{F}, \overline{F}]$=interval evaluation of $|\mathbf{J}(B_i)|$
2. $\underline{F} > 0$: no **x** in B_i with $|\mathbf{J}(\mathbf{X})| < 0$, $i = i + 1$, goto 1

INRIA SOPHIA ANTIPOLIS

Singularity checking with IA

Algorithm

1. $F = [\underline{F}, \overline{F}]$=interval evaluation of $|\mathbf{J}(B_i)|$
2. $\underline{F} > 0$: no **x** in B_i with $|\mathbf{J}(\mathbf{X})| < 0$, $i = i + 1$, goto 1
3. $\overline{F} < 0$: for all **x** in B_i negative determinant, SINGULARITY

Singularity checking with IA

Algorithm

1. $F = [\underline{F}, \overline{F}]$=interval evaluation of $|\mathbf{J}(B_i)|$
2. $\underline{F} > 0$: no **x** in B_i with $|\mathbf{J}(\mathbf{X})| < 0$, $i = i + 1$, goto 1
3. $\overline{F} < 0$: for all **x** in B_i negative determinant, SINGULARITY
4. $\underline{F} < 0, \overline{F} > 0$: split the box in two, add the 2 new boxes at the end of the list, $i = i + 1$, goto 1

INRIA SOPHIA ANTIPOLIS

Singularity checking with IA

Improvement

Extremal matrices: all scalar matrices with as elements either the lower or upper bound of the corresponding elements in $\mathbf{J}$

Singularity checking with IA

Improvement

Extremal matrices: all scalar matrices with as elements either the lower or upper bound of the corresponding elements in $\mathbf{J}$

- **Theorem 1**: if the determinants of all 2^{n^2} extremal matrices have same sign $\Rightarrow$ no singular matrices in $\mathbf{J}$

INRIA SOPHIA ANTIPOLIS

Singularity checking with IA

Improvement

Extremal matrices: all scalar matrices with as elements either the lower or upper bound of the corresponding elements in $\mathbf{J}$

- **Theorem 1**: if the determinants of all 2^{n^2} extremal matrices have same sign $\Rightarrow$ no singular matrices in $\mathbf{J}$
- **Theorem 2**: only 2^{2n-1} such matrices must be checked (Rohn, Rex)

Singularity checking with IA

Improvement

Extremal matrices: all scalar matrices with as elements either the lower or upper bound of the corresponding elements in $\mathbf{J}$

- **Theorem 1**: if the determinants of all 2^{n^2} extremal matrices have same sign $\Rightarrow$ no singular matrices in $\mathbf{J}$
- **Theorem 2**: only 2^{2n-1} such matrices must be checked (Rohn, Rex)

Rohn test can be used as a filter

INRIA SOPHIA ANTIPOLIS

Computation time

Computation time

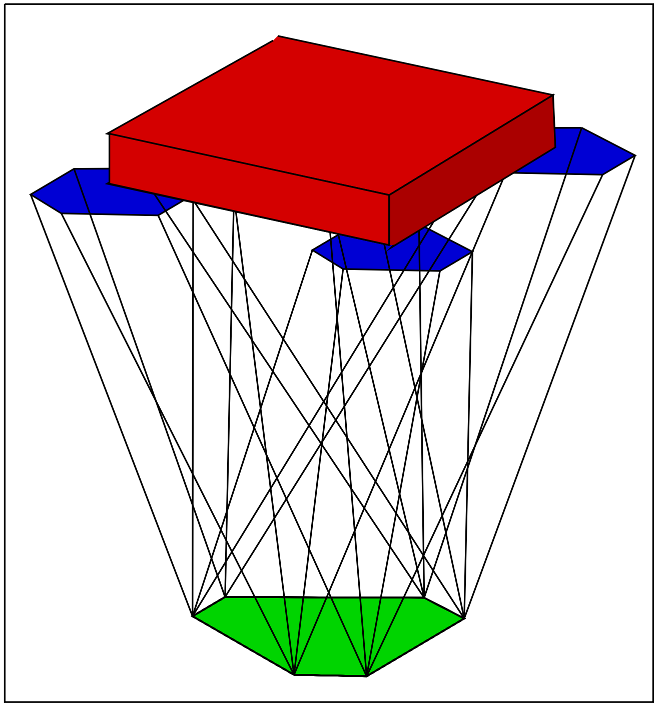
Example: robot with base radius 13, platform radius 8

Workspace: $x, y \in [-15, 15], z \in [45, 50]$, $\psi, \theta, \phi \in [-15^\circ, 15^\circ]$

Computation time

Example: robot with base radius 13, platform radius 8

Workspace: $x, y \in [-15, 15], z \in [45, 50]$, $\psi, \theta, \phi \in [-15°, 15°]$



Computation time

Example: robot with base radius 13, platform radius 8

Workspace: $x, y \in [-15, 15], z \in [45, 50]$, $\psi, \theta, \phi \in [-15°, 15°]$

- computation time to show that $\mathcal{W}$ is singularity-free: **0.01s**

INRIA SOPHIA ANTIPOLIS

Computation time

Example: robot with base radius 13, platform radius 8

Workspace: $x, y \in [-15, 15], z \in [45, 50]$, $\psi, \theta, \phi \in [-15°, 15°]$

- computation time to show that $\mathcal{W}$ is singularity-free: **0.01s**

the algorithm is able to handle uncertainty in the location of the anchor points (+36 variables)

Computation time

Example: robot with base radius 13, platform radius 8

Workspace: $x, y \in [-15, 15], z \in [45, 50]$, $\psi, \theta, \phi \in [-15°, 15°]$

- computation time to show that $\mathcal{W}$ is singularity-free: **0.01s**

the algorithm is able to handle uncertainty in the location of the anchor points (+36 variables)

- $\pm$ 0.1 uncertainties on the location of the anchor points: **515s**

Computation time

Example: robot with base radius 13, platform radius 8

Workspace: $x, y \in [-15, 15], z \in [45, 50]$, $\psi, \theta, \phi \in [-15°, 15°]$

- computation time to show that $\mathcal{W}$ is singularity-free: **0.01s**

the algorithm is able to handle uncertainty in the location of the anchor points (+36 variables)

- $\pm$ 0.1 uncertainties on the location of the anchor points: **515s**

meaning that ALL robots in this family are singularity-free

INRIA
SOPHIA ANTIPOLIS

Singularity index

Singularity index

How can can we define a singularity index ?

Singularity index

How can can we define a singularity index ?

- that is invariant with units choice

Singularity index

How can can we define a singularity index ?

- that is invariant with units choice
- that has a physical meaning

Singularity index

Singularity index

Classical indices rely on the jacobian of the robot

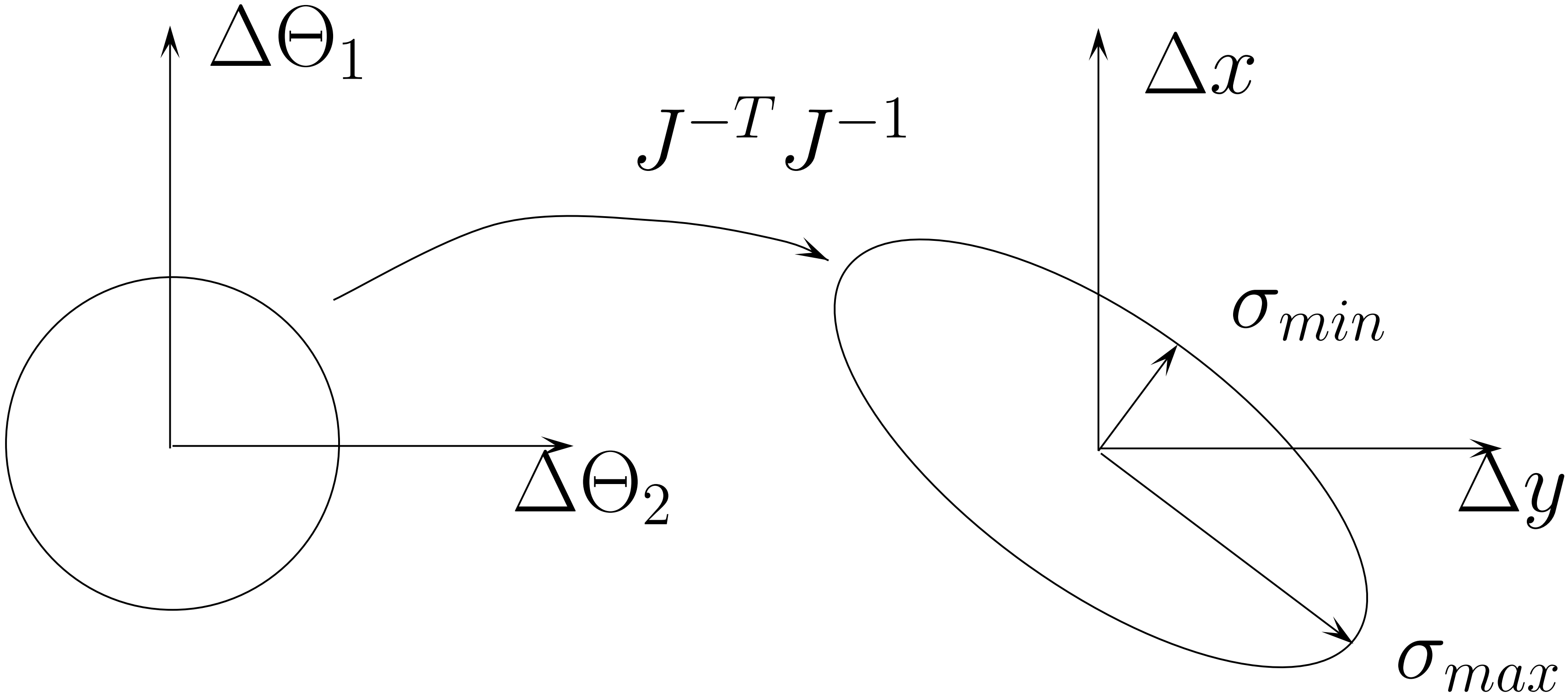
Singularity index

Example: manipulability ellipsoid

Singularity index

Example: manipulability ellipsoid

$$\mathbf{\Delta\Theta}^T\mathbf{\Delta\Theta} \leq 1 \quad \Rightarrow \quad \mathbf{\Delta X}^T J^{-T} J^{-1} \mathbf{\Delta X} \leq 1$$



possible accuracy index I: σ_{max}

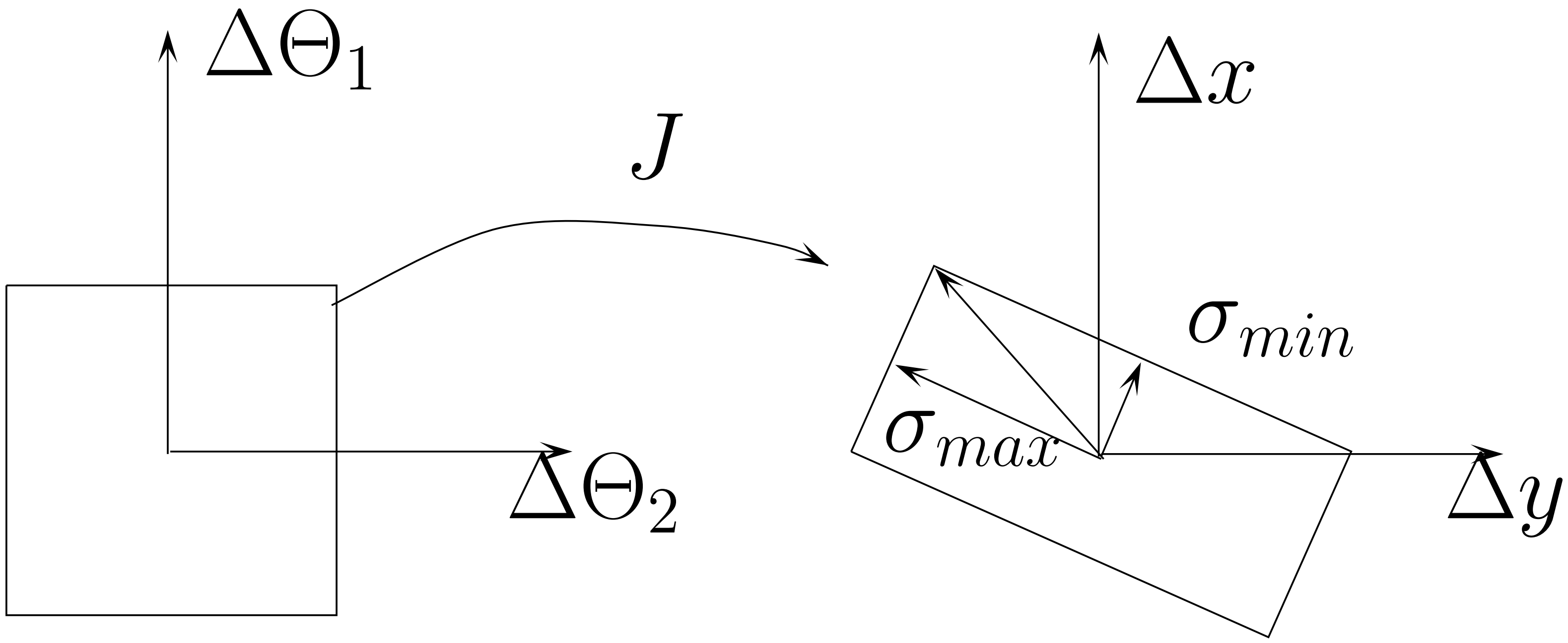
INRIA SOPHIA ANTIPOLIS

Singularity index

In fact $\boldsymbol{\Delta\Theta}$ bounded means: $|\Delta\Theta_i| \leq 1$

- each $\Delta\Theta_i$ is independent
- the possible region for $\boldsymbol{\Delta\Theta}$ is a square
- the linear mapping induced by J^{-1} transforms this square into a tilted rectangle

Singularity index



- σ_{max} is not the maximal positioning error

Singularity index

Another example: condition number

Singularity index

Another example: condition number

$$\mathbf{J^{-1}}(\mathbf{X})\mathbf{\Delta X} = \Delta\mathbf{\Theta}$$

$$\frac{||\mathbf{\Delta X}||}{||\mathbf{X}||} \leq ||\mathbf{J^{-1}}||||\mathbf{J}||\frac{||\Delta\mathbf{\Theta}||}{||\mathbf{\Theta}||}$$

condition number κ: $||\mathbf{J^{-1}}||||\mathbf{J}|| \Rightarrow$ relative amplification factor

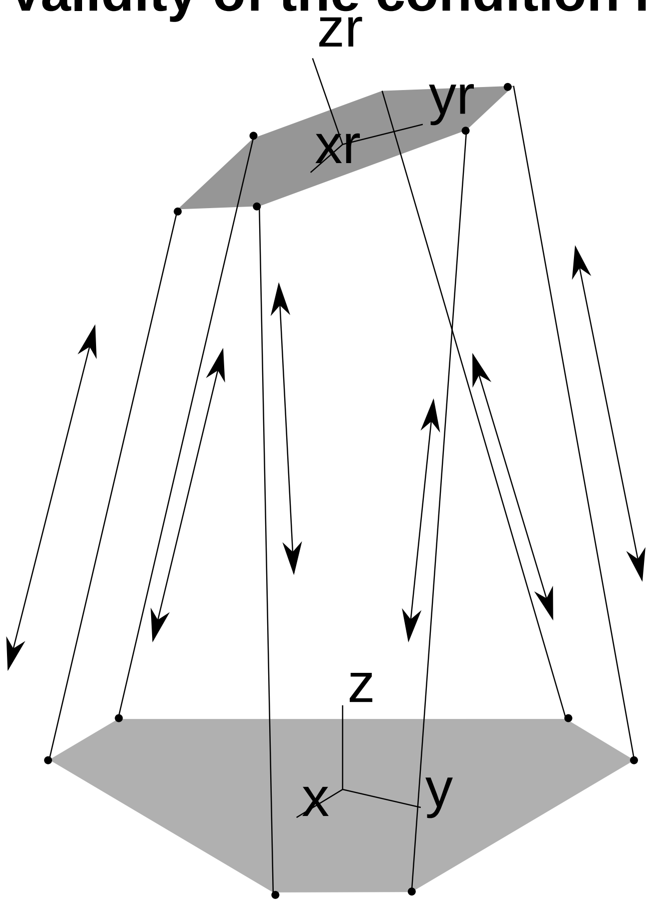
Singularity index

Condition number:

- local index
- has usually no closed-form
- value changes with the norm choice
- usual norms: 2-norm, Frobenius norm $\Rightarrow \kappa \in [1, \infty]$
- $1/\kappa$ often used, $1/\kappa \in [0, 1]$, 0 at a singularity
- **meaning when the robot has both translation and orientation d.o.f. ?**

Singularity index

Validity of the condition number



3 reference poses P_1, P_2, P_3

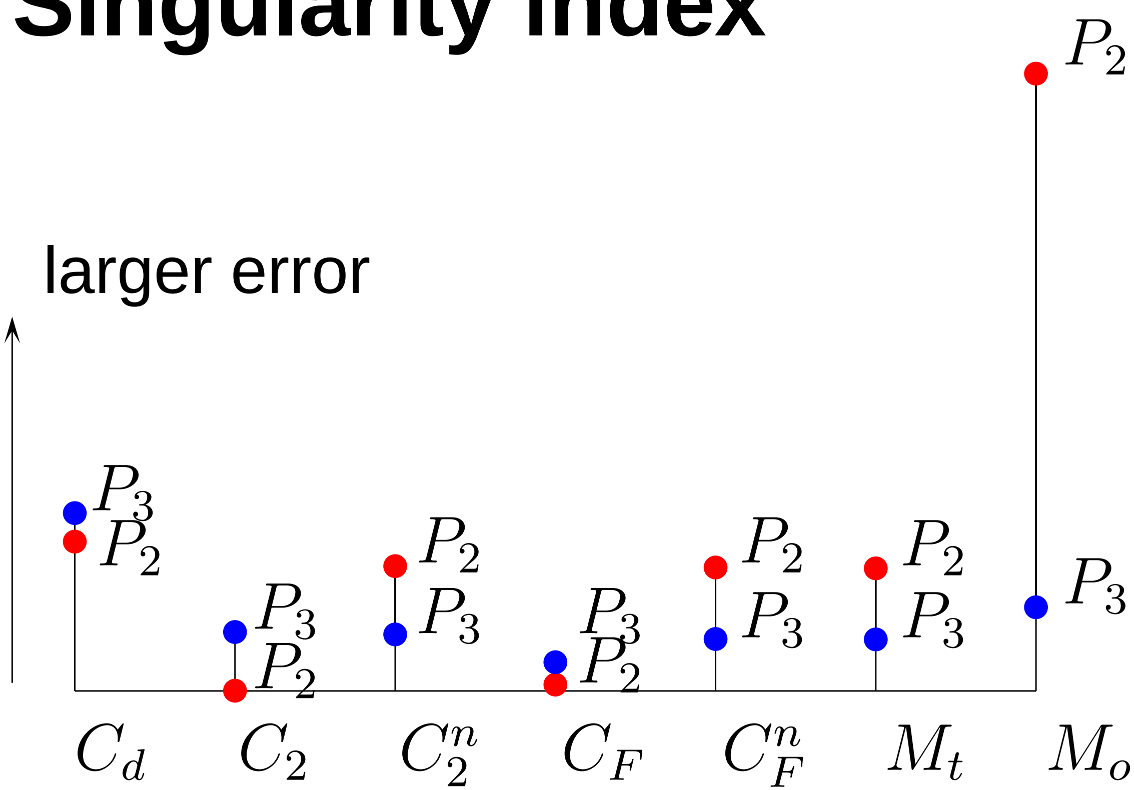
$$|\Delta\Theta^a| \leq 1$$

$$\Delta X_{x,y,z,\theta_x,\theta_y,\theta_z} = \sum_{k=1}^{k=6} |J^k_{x,y,z,\theta_x,\theta_y,\theta_z}|$$

ranking according to accuracy: $P_1 \gg P_2 > P_3$

INRIA SOPHIA ANTIPOLIS

Singularity index



C_d: $|\mathbf{J}^{-1}|$ (manipulability)

C_2: κ, 2-norm

C_2^n: κ, 2-norm, normalized $\mathbf{J}^{-1}$

C_F: κ, Frobenius

C_F^n: κ, Frobenius, normalized $\mathbf{J}^{-1}$

M_t: κ, translation part J

M_o: κ, orientation part J

INRIA SOPHIA ANTIPOLIS

Singularity index

Not really consistent with accuracy ranking !

Singularity index

Global conditioning indices

To characterize the dexterity over a given workspace W

$$\mathrm{GCI} = \frac{\displaystyle\int_W \left(\frac{1}{\kappa} \right) dW}{\displaystyle\int_W dW}$$

Problem: how to compute it ?

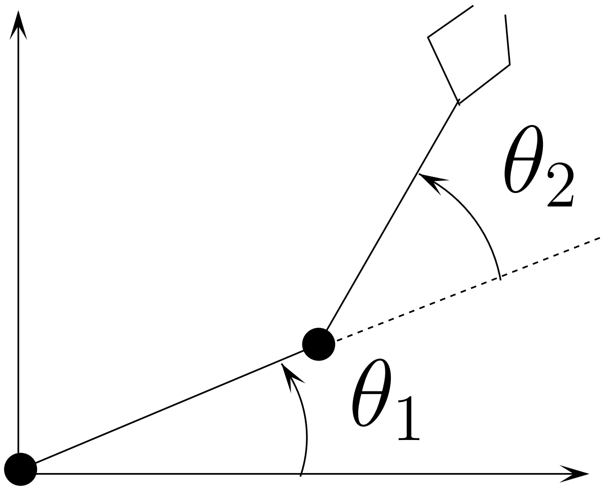
INRIA SOPHIA ANTIPOLIS

Singularity index

- sample each W axis, step $l \to m$ poses, GCI = $\frac{\sum \kappa_i}{m}$,
- computation time: $O(l^6)$, error ?
- assumption: if $\frac{\text{GCI}(m+50)-\text{GCI}(m)}{\text{GCI}(m+50)} \leq 0.5\%$, then $\frac{|\text{GCI}-\text{GCI}(m+50)|}{\text{GCI}} \leq 0.5\%$ wrong!

Singularity index

Counter example: planar 2R robot



condition number only function of θ_2:

GCI can be calculated exactly

$$\frac{\mathrm{GCI}(60)-\mathrm{GCI}(50)}{\mathrm{GCI}(60)} = \mathbf{0.3768\%} \quad \text{while} \quad \frac{|\mathrm{GCI}-\mathrm{GCI}(60)|}{\mathrm{GCI}} = \mathbf{1.7514\%}$$

Another approach

Another approach

$$\tau_j = \frac{|\mathbf{M}|}{|\mathbf{J}^{-\mathbf{T}}|}$$

Another approach

$$\tau_j = \frac{|\mathbf{M}|}{|\mathbf{J}^{-\mathbf{T}}|}$$

$|\mathbf{J}^{-\mathbf{T}}| = 0$ leads to a problem ...

Another approach

$$\tau_j = \frac{|\mathbf{M}|}{|\mathbf{J^{-T}}|}$$

$|\mathbf{J^{-T}}| = 0$ leads to a problem . . .

except if we have also $|\mathbf{M}| = 0$

INRIA SOPHIA ANTIPOLIS

Another approach

$$\tau_j = \frac{|\mathbf{M}|}{|\mathbf{J}^{-\mathbf{T}}|}$$

$|\mathbf{J}^{-\mathbf{T}}| = 0$ leads to a problem ...

except if we have also $|\mathbf{M}| = 0$

At a singularity we may still have finite articular forces

Another approach

$$\tau_j = \frac{|\mathbf{M}|}{|\mathbf{J^{-T}}|}$$

$|\mathbf{J^{-T}}| = 0$ leads to a problem ...

except if we have also $|\mathbf{M}| = 0$

At a singularity we may still have finite articular forces

but in the vicinity of a singularity we may have very large articular forces

Another approach

$$\tau_j = \frac{|\mathbf{M}|}{|\mathbf{J^{-T}}|}$$

$|\mathbf{J^{-T}}| = 0$ leads to a problem ...

except if we have also $|\mathbf{M}| = 0$

At a singularity we may still have finite articular forces

but in the vicinity of a singularity we may have very large articular forces

Hence we have to consider also the vicinity of a singularity

- we define a maximal force/torque τ_{max} for the kinematic chains

- we define a maximal force/torque τ_{max} for the kinematic chains
- the closeness to a singularity is defined as

$$d = Min(\tau_{max} - |\tau_j|)$$

- we define a maximal force/torque τ_{max} for the kinematic chains
- the closeness to a singularity is defined as

$$d = Min(\tau_{max} - |\tau_j|)$$

- $d \geq 0$ if $|\tau_j| \leq \tau_{max}$
- $d < 0 \Rightarrow$ breakdown of the mechanism

INRIA SOPHIA ANTIPOLIS

- we define a maximal force/torque τ_{max} for the kinematic chains
- the closeness to a singularity is defined as

$$d = Min(\tau_{max} - |\tau_j|)$$

- $d \geq 0$ if $|\tau_j| \leq \tau_{max}$
- $d < 0 \Rightarrow$ breakdown of the mechanism

closeness is unit invariant, has a physical meaning

INRIA SOPHIA ANTIPOLIS

Static workspace

Static workspace

Static workspace: location of the EE such that $d \geq 0$

INRIA SOPHIA ANTIPOLIS

Static workspace

Static workspace: location of the EE such that $d \geq 0$

$$\Downarrow$$

no breakdown of the mechanism

Static workspace

Static workspace: location of the EE such that $d \geq 0$

How can we compute the static workspace ?

Static workspace

Static workspace: location of the EE such that $d \geq 0$

How can we compute the static workspace ?

- for a given load
- for a set of loads

INRIA SOPHIA ANTIPOLIS

Static workspace: 2D case

Static workspace: 2D case

- for a 2 dof robot
- for a given load

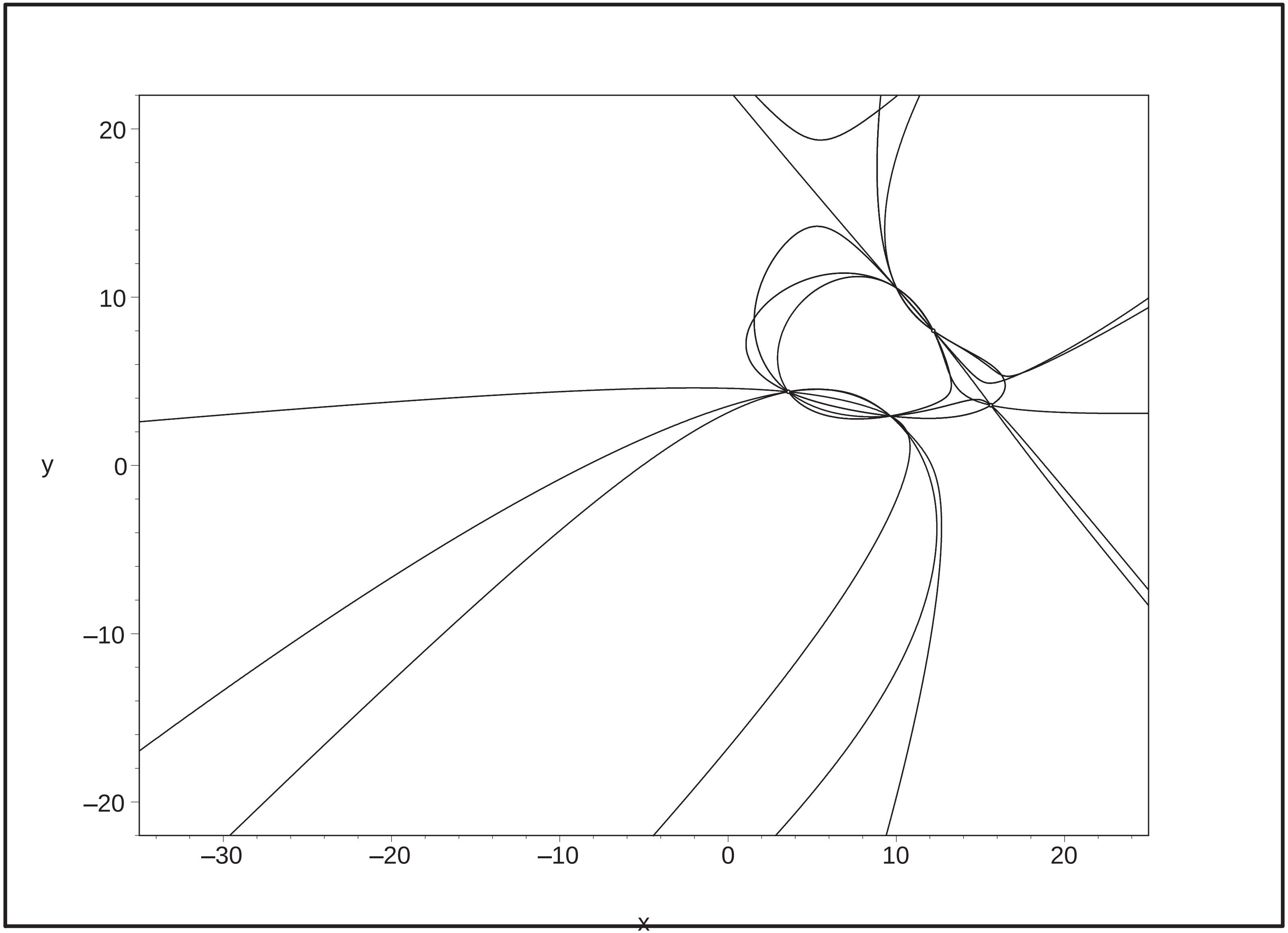
INRIA SOPHIA ANTIPOLIS

Static workspace: 2D case

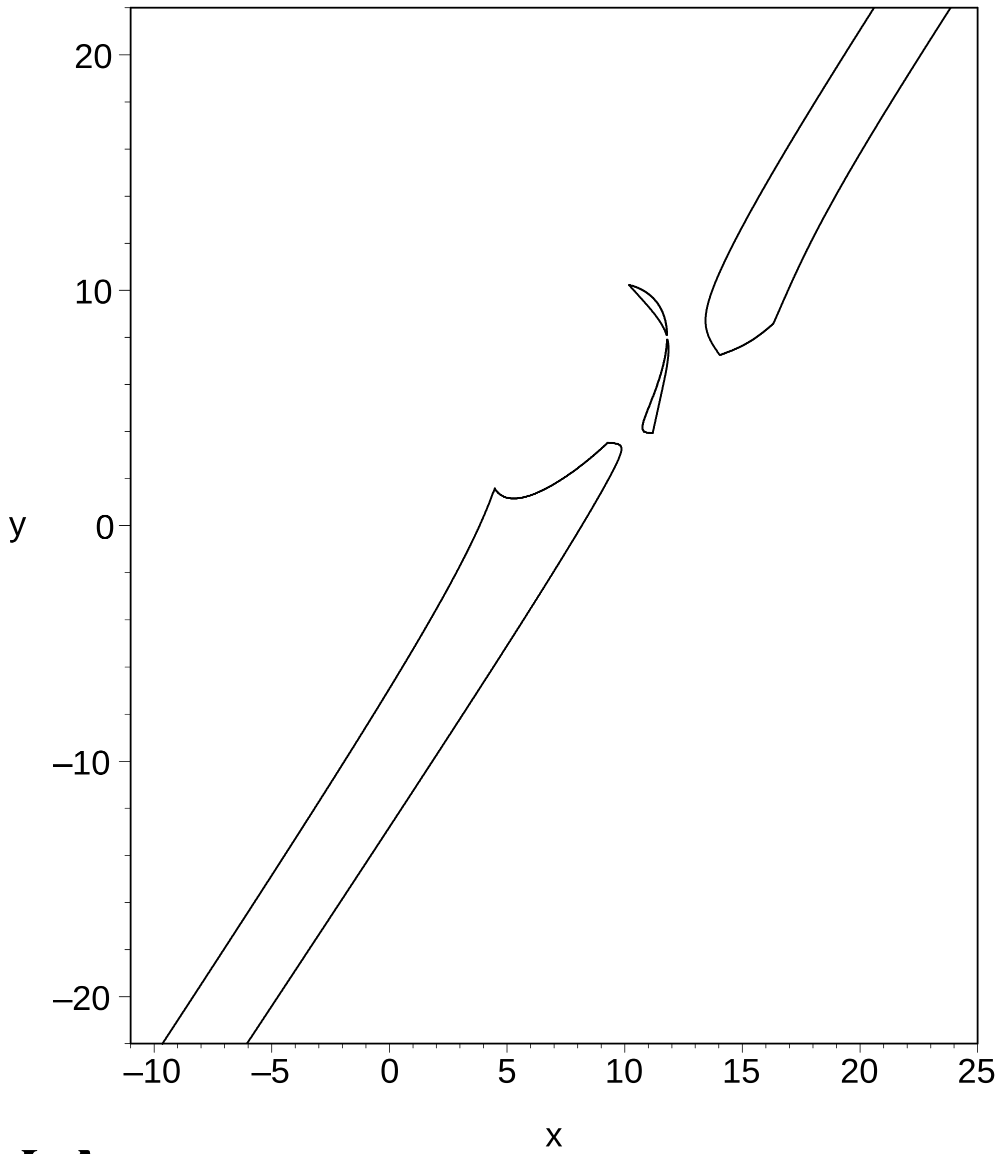
- for a 2 dof robot
- for a given load

we can calculate in closed-form part of analytic curves that will be part of the border of the static workspace

Static workspace: 2D case



Static workspace: 2D case



Static workspace: general case

Static workspace: general case

- for a n dof robot
- for a set of loads
- with uncertainties on the robot's geometry

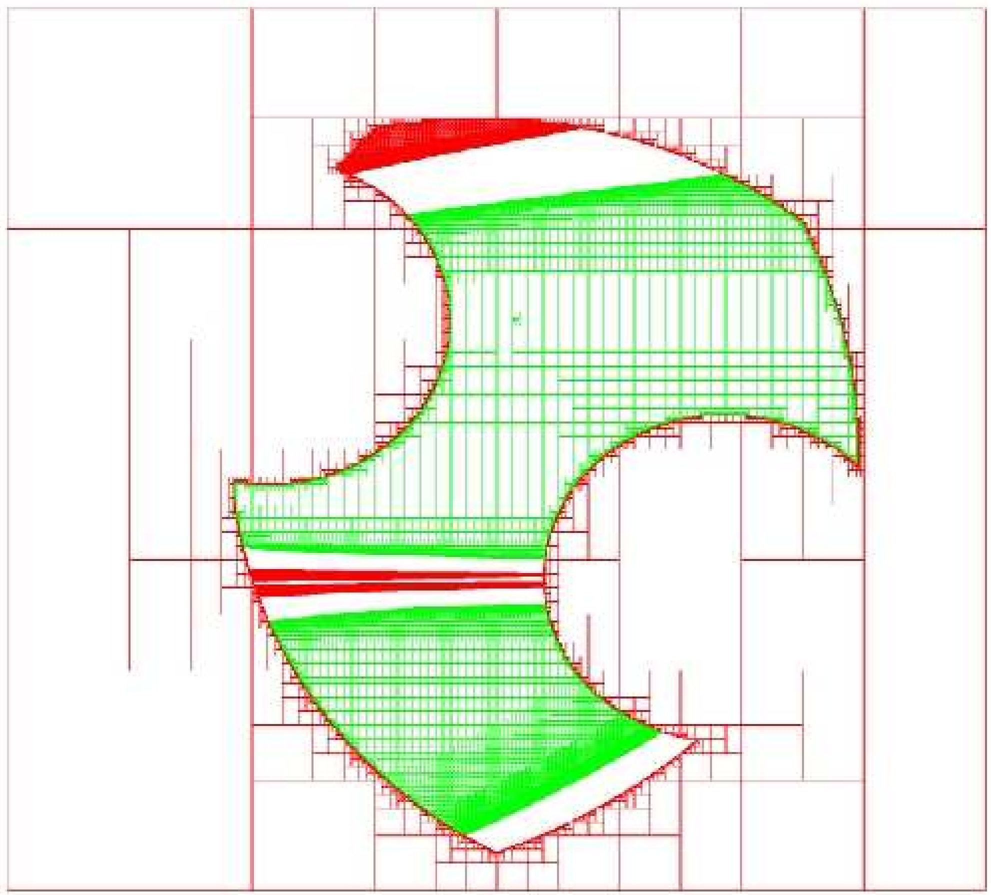
Static workspace: general case

- for a n dof robot
- for a set of loads
- with uncertainties on the robot's geometry

Interval analysis algorithm allows to obtain an accurate approximation of the static workspace

Static workspace: general case

Example: cross-section of a 6D static workspace



Conclusion

- checking if a singularity is present in a given workspace is feasible efficiently
- just forget about so-called singularity, dexterity indices
- the best singularity index is the one that will prohibit your robot to do its job

INRIA
SOPHIA ANTIPOLIS