LECTURE 5

Speed and separation monitoring

Paolo Rocco

Speed and separation monitoring

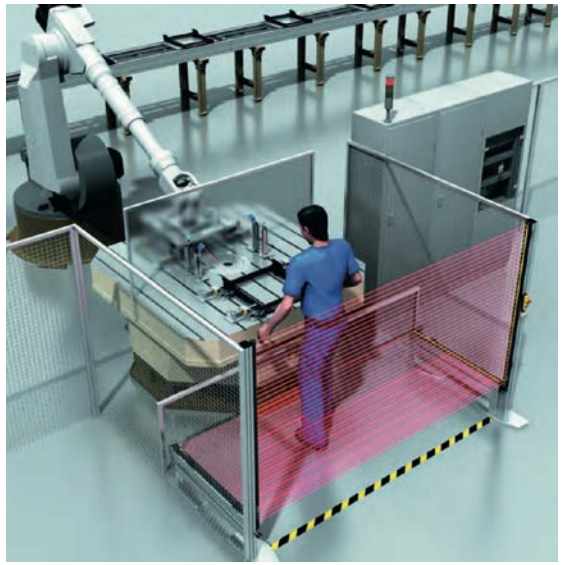
In this collaboration mode, the human can be close to the robot, provided that the speed of the robot is always such that the robot can come to a complete stop before colliding with the human.

Then:

- The closer the operator is, the slower the robot must move
- A safety-rated device is needed to monitor the distance of the operator from the robot

Sensor technologies: light-based

- Used to monitor 2D areas
- Light curtains return whether one or more beam has been interrupted
- Laser scanners measure the distance between the objects that fall into its sensing field

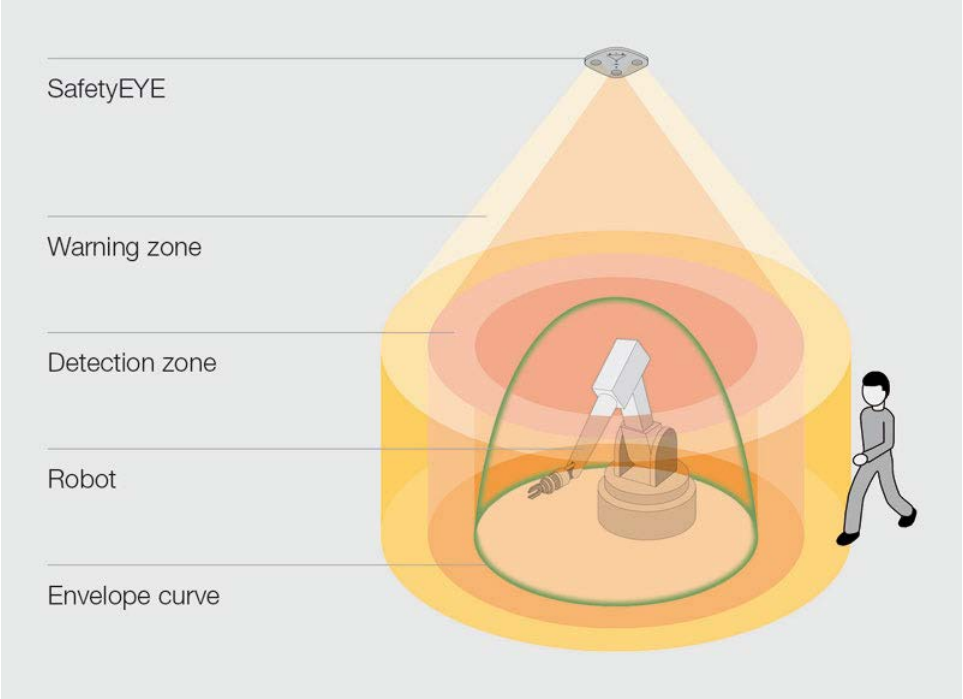


Source: ABB

Link to the video: https://www.youtube.com/watch?v=2ad_oI_4eJ8

Sensor technologies: vision-based

- Camera systems have long been used in various industrial sectors for monitoring tasks as well as quality control measurements.
- For camera systems to be used in the field of safe automation, they must satisfy safety-relevant specifications.



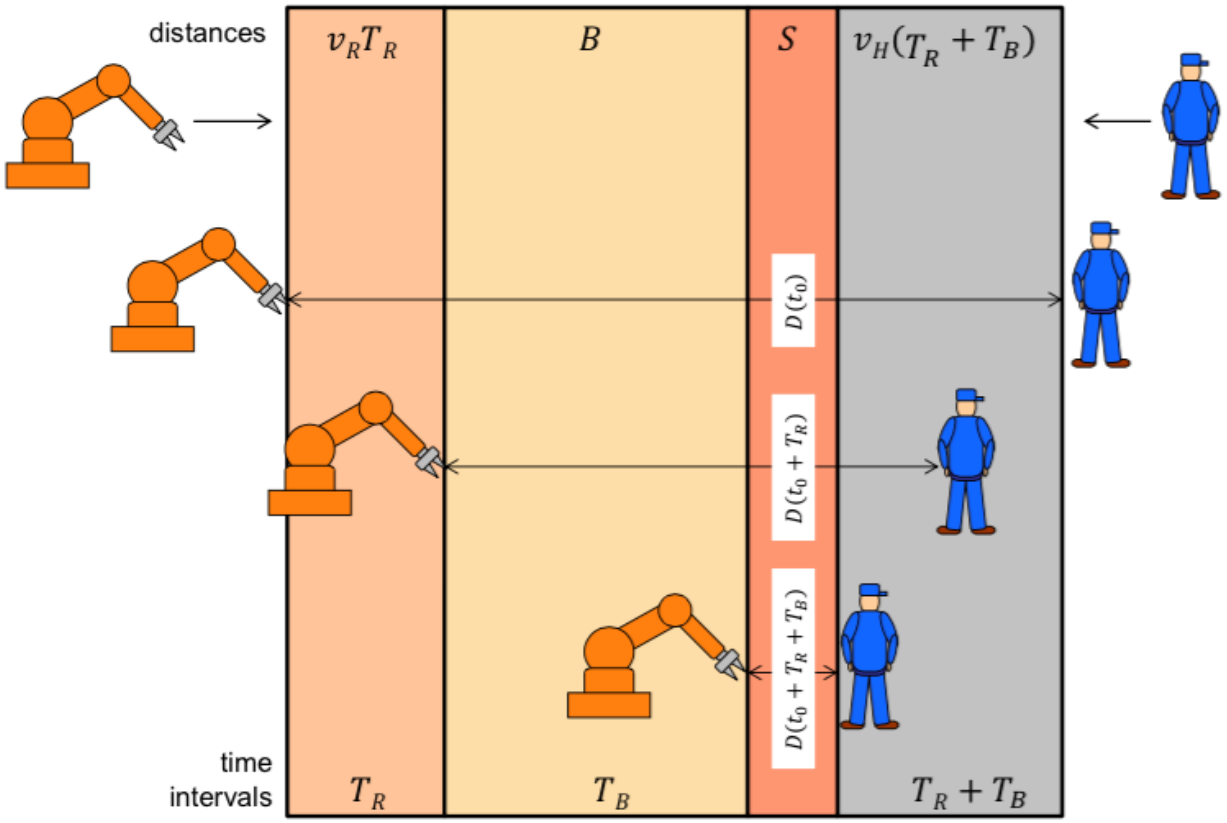
https://www.pilz.com/en-INT/eshop/00106002207042/SafetyEYE-Safe-camera-system

Sensor technologies: capacitive skin

Source: Bosch

Link to the video: https://www.youtube.com/watch?v=stFYQaHULLk

Speed and separation monitoring



$$D(t_0) - v_R(T_R + T_B) - v_H(T_R + T_B) \geq S$$

v_R = robot speed

v_H = human speed

T_R = robot controller reaction time

T_B = robot stopping time

S = minimum separation distance

$D(t_0)$ = separation distance at time t_0

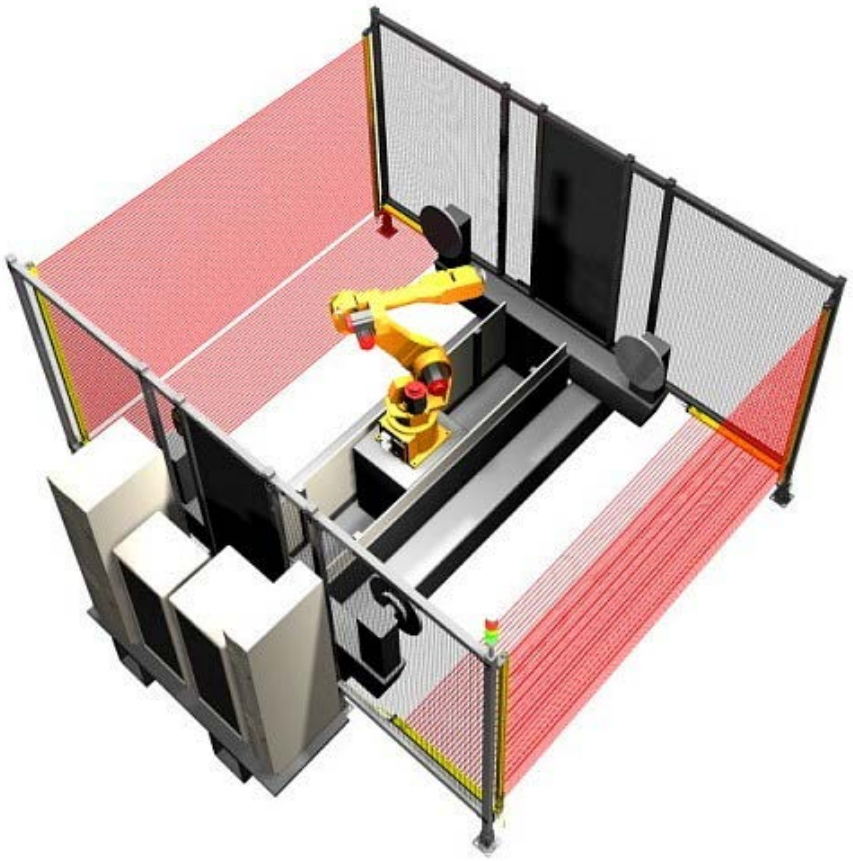
Source: ABB Corporate Research

Speed and separation monitoring

$$v_R \leq \frac{D(t_0) - S}{T_R + T_B} - v_H$$

- v_H is in the order of 1.6 m/s (walking speed)
- T_R in the order of 10 ms (reaction time of the robot controller)
- T_B in the order of 1 s (braking time of the robot actuators)

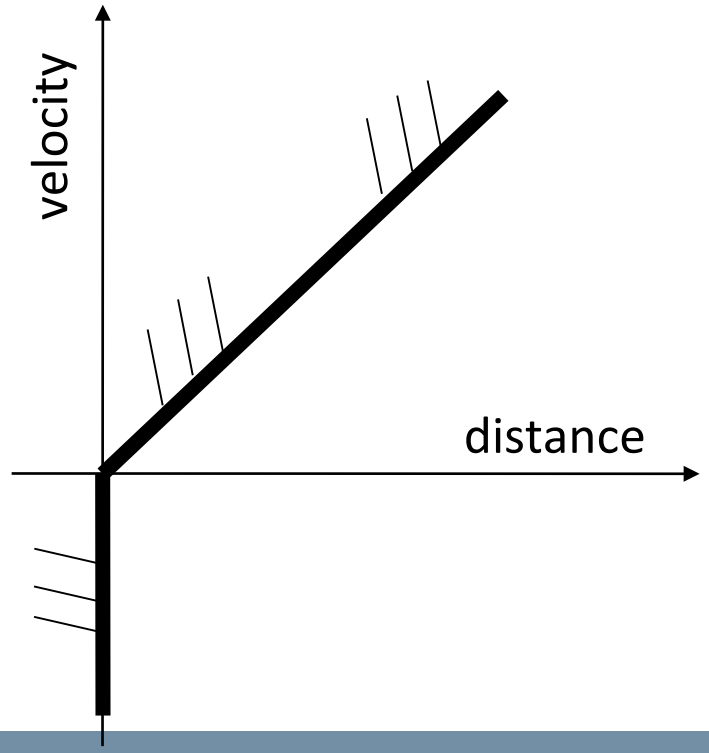
$$D(t_0) \geq v_H(T_R + T_B) = 1.6 \times 1.01 = 1.616\, m$$



Safety as a motion constraint

Consistent with the speed and separation monitoring safety standard, we can think of safety as a constraint for robot motion.

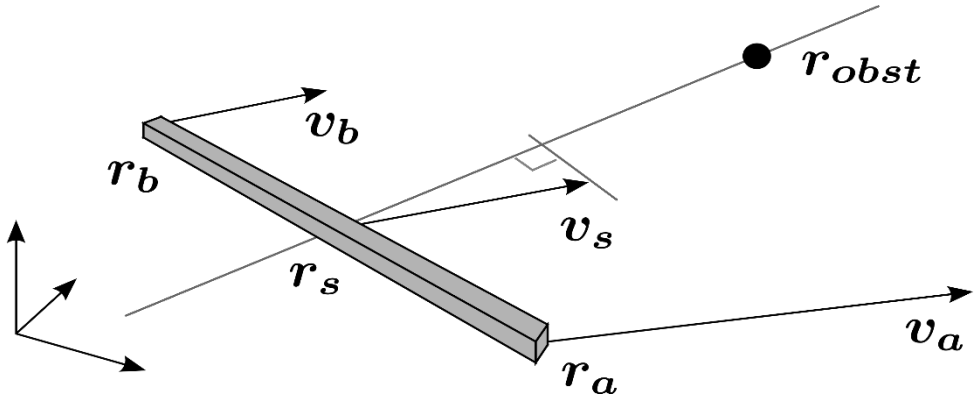
Distance ≥ Velocity * StoppingTime



Safety constraints might require the robot to slow down on path

Safety as a motion constraint

A **multi-body** version of the safety constraints:



$$r_s = r_a + s\left(r_b - r_a\right)$$

$$\left\|r_{obst} - r_s\right\| \geq t_b \frac{\left(r_{obst} - r_s\right)^T}{\left\|r_{obst} - r_s\right\|} v_s$$

distance velocity

$$\boxed{\left\|r_{obst} - r_s\right\|^2 - t_b\left(r_{obst} - r_s\right)^T v_s \geq 0, \forall s \in [0,1]}$$

Safety as a motion constraint

Safety constraints can be rewritten as follows:

$$\|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|^2 + \beta s + \gamma \geq 0, \forall s \in [0, 1]$$

A sufficient condition is the following one:

$$\min_s \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|^2 + \beta s + \gamma \geq 0$$

no longer depends on *s*

linear in *s*

which can be split into two constraints:

$$\begin{cases} \gamma + \min_s \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|^2 \geq 0 \\ \beta + \gamma + \min_s \|\boldsymbol{r_{obst}} - \boldsymbol{r_s}\|^2 \geq 0 \end{cases}$$

Safety as a motion constraint

Introducing the following quantities:

$$\boldsymbol{E}_i = \begin{bmatrix} (\boldsymbol{r}_{obst} - \boldsymbol{r}_a)^T \boldsymbol{J}_a \\ (\boldsymbol{r}_{obst} - \boldsymbol{r}_a)^T \boldsymbol{J}_b - (\boldsymbol{r}_b - \boldsymbol{r}_a)^T \boldsymbol{J}_a \end{bmatrix}$$

$$\boldsymbol{f}_i = \min_s \|\boldsymbol{r}_{obst} - \boldsymbol{r}_s\|^2 \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

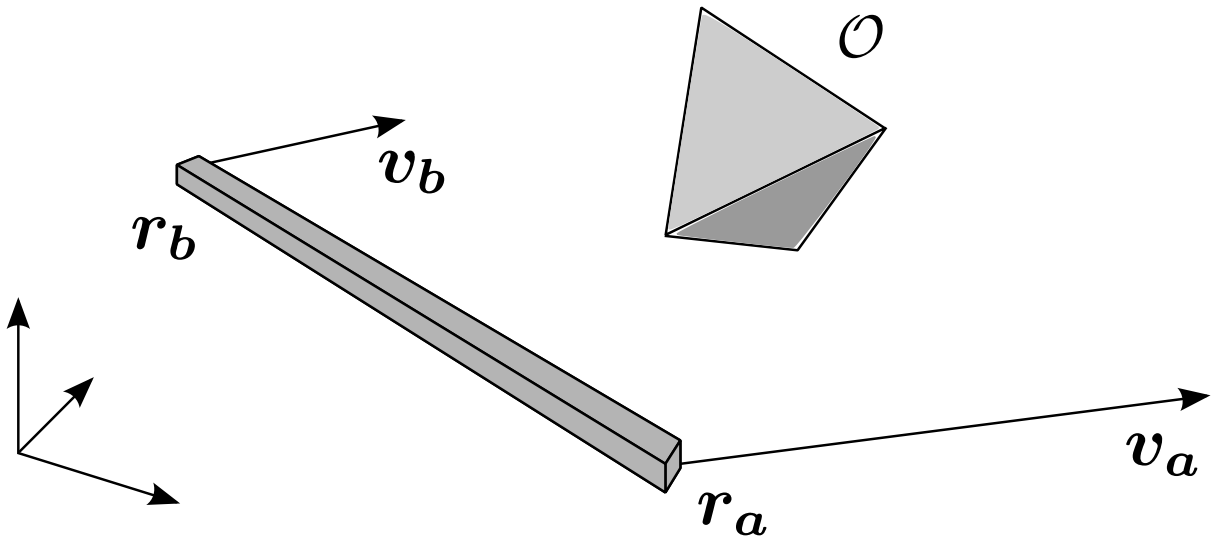
The safety constraints (for each link-obstacle pair) can be finally written in terms of a **linear constraint** in the robot velocities.

$$\boxed{t_b \boldsymbol{E}_i \dot{\boldsymbol{q}} \leq \boldsymbol{f}_i}$$

Zanchettin and Rocco, IROS 2013

Extension to generic obstacles

Generic obstacles, as **human co-workers**, can be considered as well:



Extension to an arbitrarily shaped obstacle can be obtained applying the inequality to **all its points**.

Extension to generic obstacles

A **sufficient condition** is the following one

$$t_b \boldsymbol{E_i}\left(\boldsymbol{r}_{obst}\right)\dot{\boldsymbol{q}} \leq \min_{\boldsymbol{r}_{obst}} \left\|\boldsymbol{f_i}\left(\boldsymbol{r}_{obst}\right)\right\|_\infty \begin{bmatrix}1\\1\end{bmatrix}, \forall \boldsymbol{r}_{obst} \in \mathrm{vert}\left(\mathcal{O}\right)$$

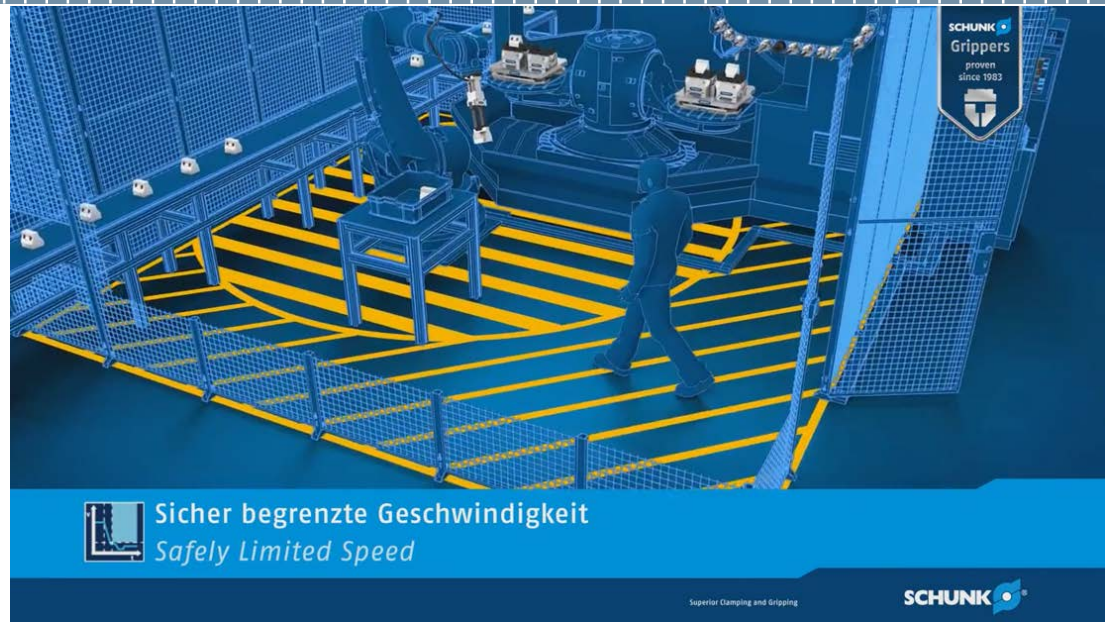
Meaning that we just need to:

- account for vertices (convex obstacles)
- compute the link-obstacle distance (e.g. using GJK)

Prediction of human space occupancy

Human workers moving in a robotic cell can be considered as moving obstacles.

To predict their motion we can try to infer the occupied space whilst the robot is stopping or slowing down.



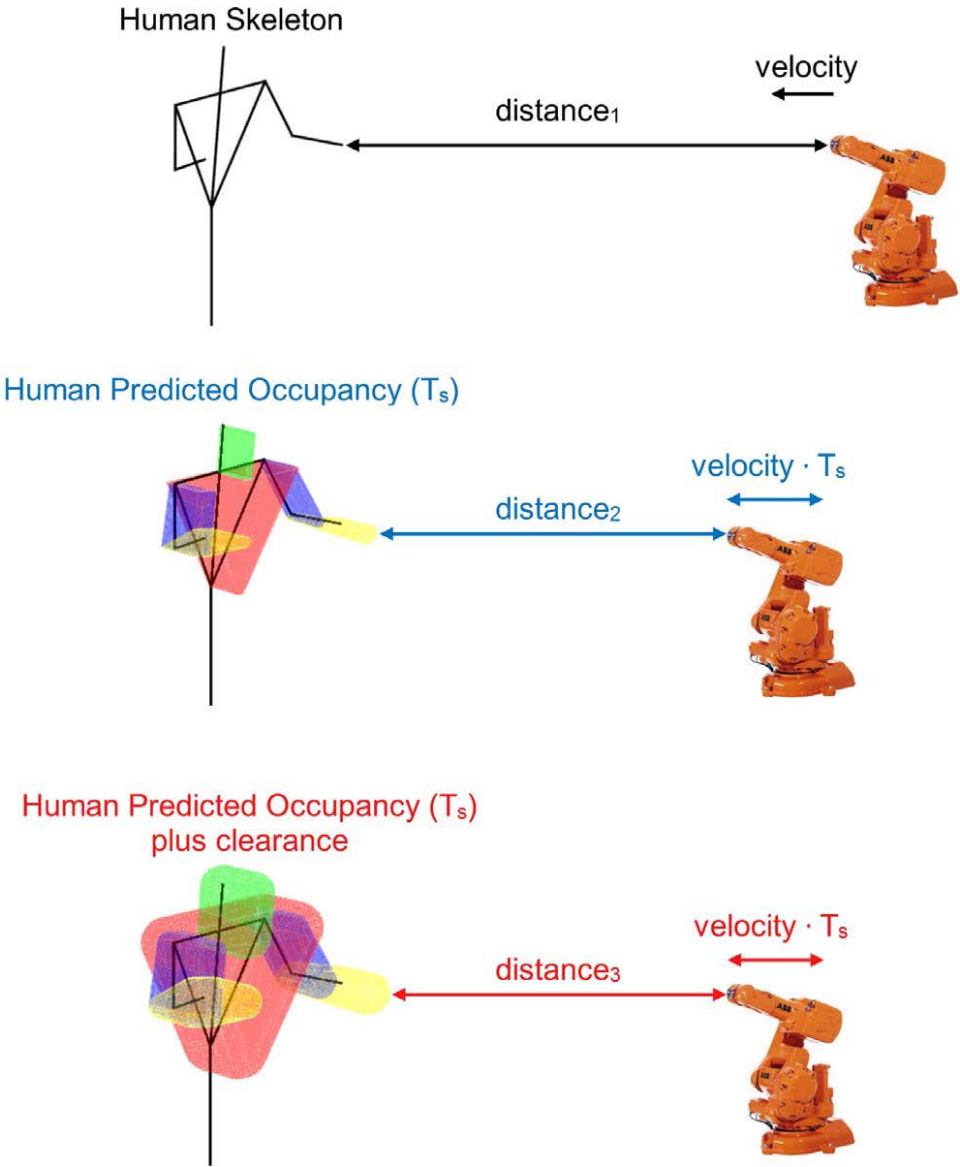
The main building blocks of this motion prediction strategy are:

1. kinematic model of human motion
2. reachable sets
3. swept volumes

Prediction of human space occupancy

In addition to prediction of human occupancy, a clearance factor can also be considered.

The clearance parameter allows to take into account both robot and obstacle dimensions, sensor uncertainties and ultimately an actual clearance.

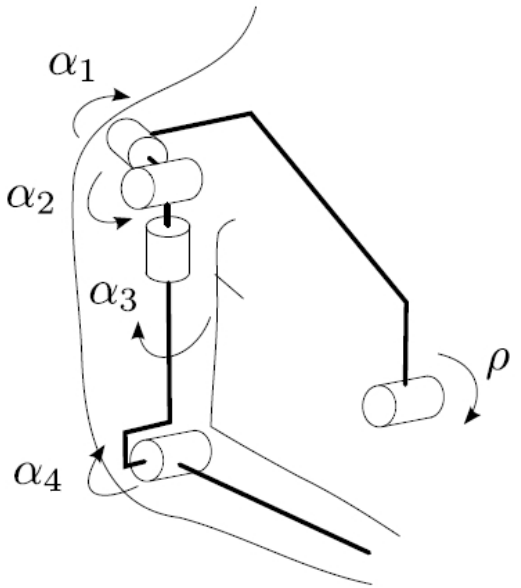


Prediction of human space occupancy

Kinematic model of human motion:

1. walking kinematics: $\dot{x} = v\cos\theta \quad \dot{y} = v\sin\theta \quad \dot{\theta} = \omega$

2. arm kinematics:



$$\boldsymbol{p} = \begin{bmatrix} x & y & \theta & \rho & \boldsymbol{\alpha}^{right} & \boldsymbol{\alpha}^{left} \end{bmatrix}^T$$

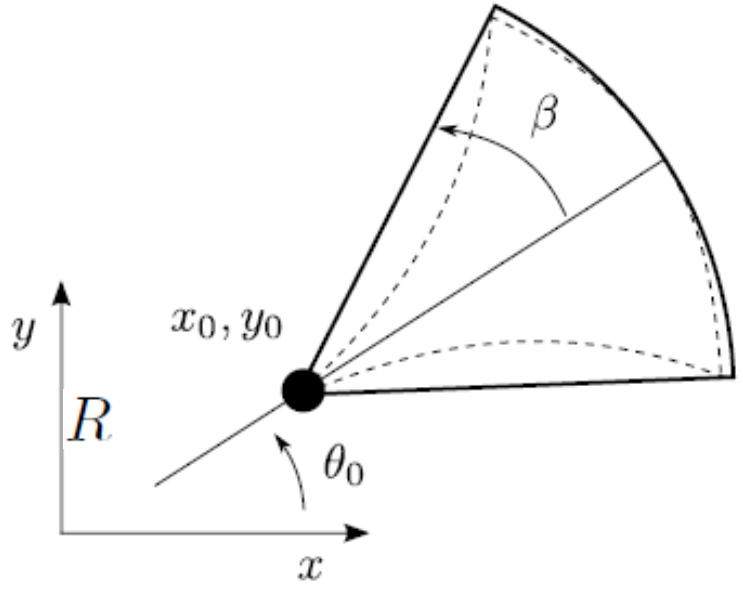
Prediction of human space occupancy

Unicycle Reachable Set: given the following velocity constraints

$$0 \leq v \leq v^{max} \quad |\omega| \leq \omega^{max}$$

the reachable set of the unicycle can be conservatively represented by a circular sector:

- angle: $$\beta = 2\cos^{-1}\left(\frac{\sin\left(\omega^{max}\boldsymbol{t_b}\right)}{\sqrt{2-2\cos\left(\omega^{max}\boldsymbol{t_b}\right)}}\right)$$
- radius: $$R = v^{max}\boldsymbol{t_b}$$



Prediction of human space occupancy

Arm Reachable Set: given bounds on torso and arms joint velocities, the reachable set of the torso and upper limbs can be determined as:

$$\boldsymbol{\alpha_{t_b} = \alpha \pm t_b \dot{\alpha}^{max}} \qquad \rho t_b = \rho \pm \boldsymbol{t_b} \dot{\rho}^{max}$$

Then the following constraints are enforced:

$$-9 \leq \alpha_1 \leq 160$$

$$-43 + \frac{\alpha_1}{3} \leq \alpha_2 \leq 153 - \frac{\alpha_1}{6}$$

$$-90 + \frac{7\alpha_1}{9} - \frac{\alpha_2}{9} + \frac{2\alpha_1\alpha_2}{810} \leq \alpha_3 \leq 60 + \frac{4\alpha_1}{9} - \frac{5\alpha_2}{9} + \frac{5\alpha_1\alpha_2}{810}$$

$$20 \leq \alpha_4 \leq 180$$

$$-30 \leq \rho \leq 90$$

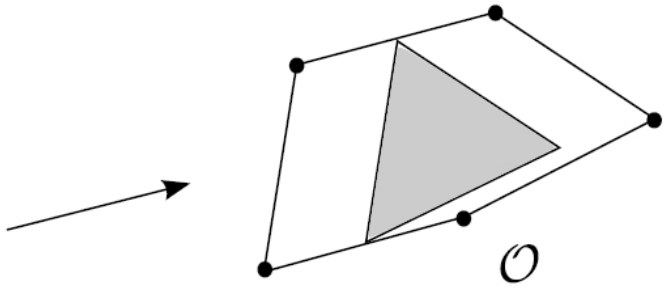
Prediction of human space occupancy

Swept Volumes calculation:

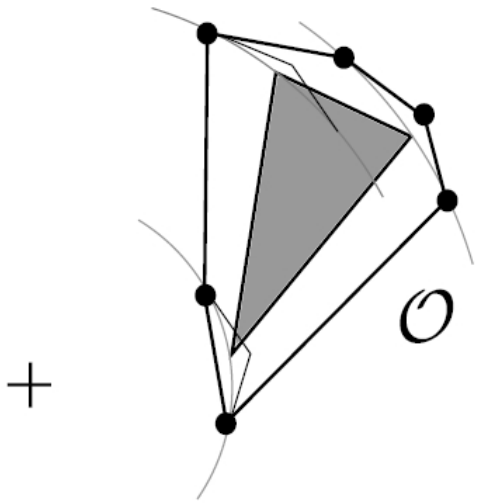
1. consider each human limb in terms of its vertex set;
2. select the DoFs connecting the current limb to the world frame;
3. for each selected DoF:
   I. apply translational / rotational sweep according to reachable sets:

$$p_{inf} \leq p \leq p^{sup}$$

   II. update the vertex set;
4. compute the convex hull of the obtained set of points;



translational sweep



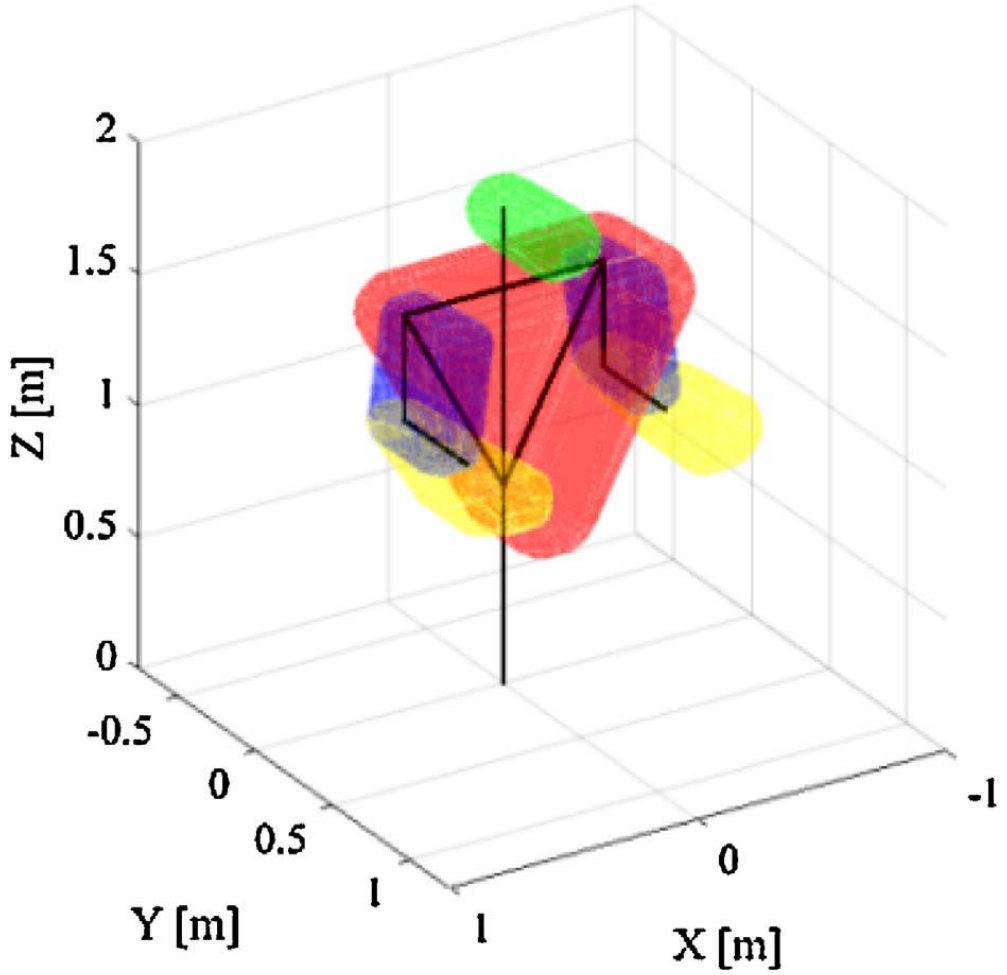
rotational sweep

Prediction of human space occupancy

Swept Volumes calculation:

- the procedure is repeated for each human limb (head – G, torso – R, upper arms – B, lower arms – Y);

- each limb is characterized by a "clearance" parameter → a sphere of corresponding radius is mounted on each swept volume point;



Safety-aware velocity scaling

$$\max_{\delta,\dot{\boldsymbol{q}}} \delta$$

maximize productivity
(minimize cycle time)

$$t_{b,i}\boldsymbol{E_i}\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}} \leq \boldsymbol{f_i}\left(\boldsymbol{q}\right), \forall i = 0, \ldots, n-1$$

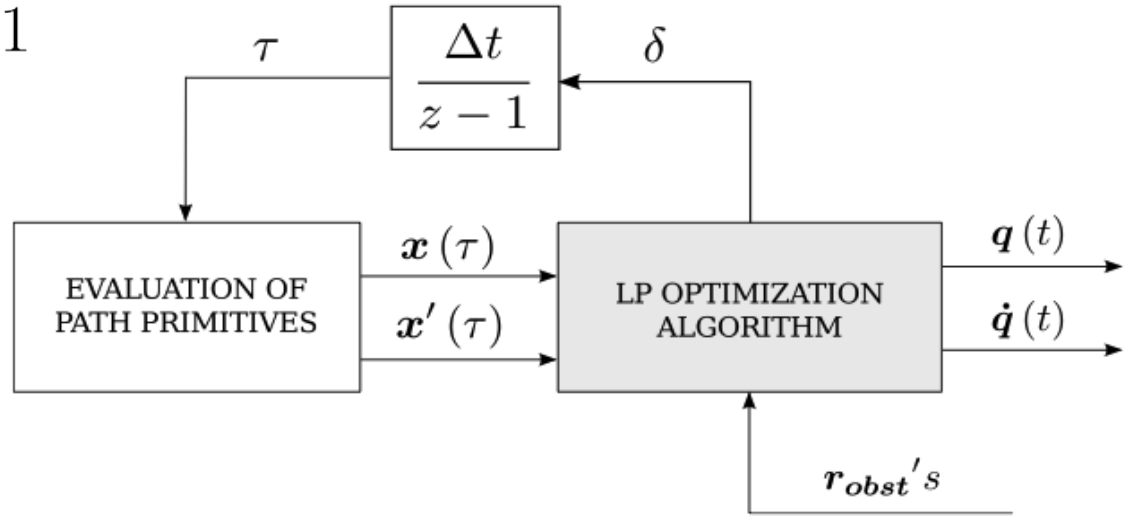
safety constraints

$$\boldsymbol{J}\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}} = \delta\dot{\boldsymbol{x}}$$

task (path) constraints

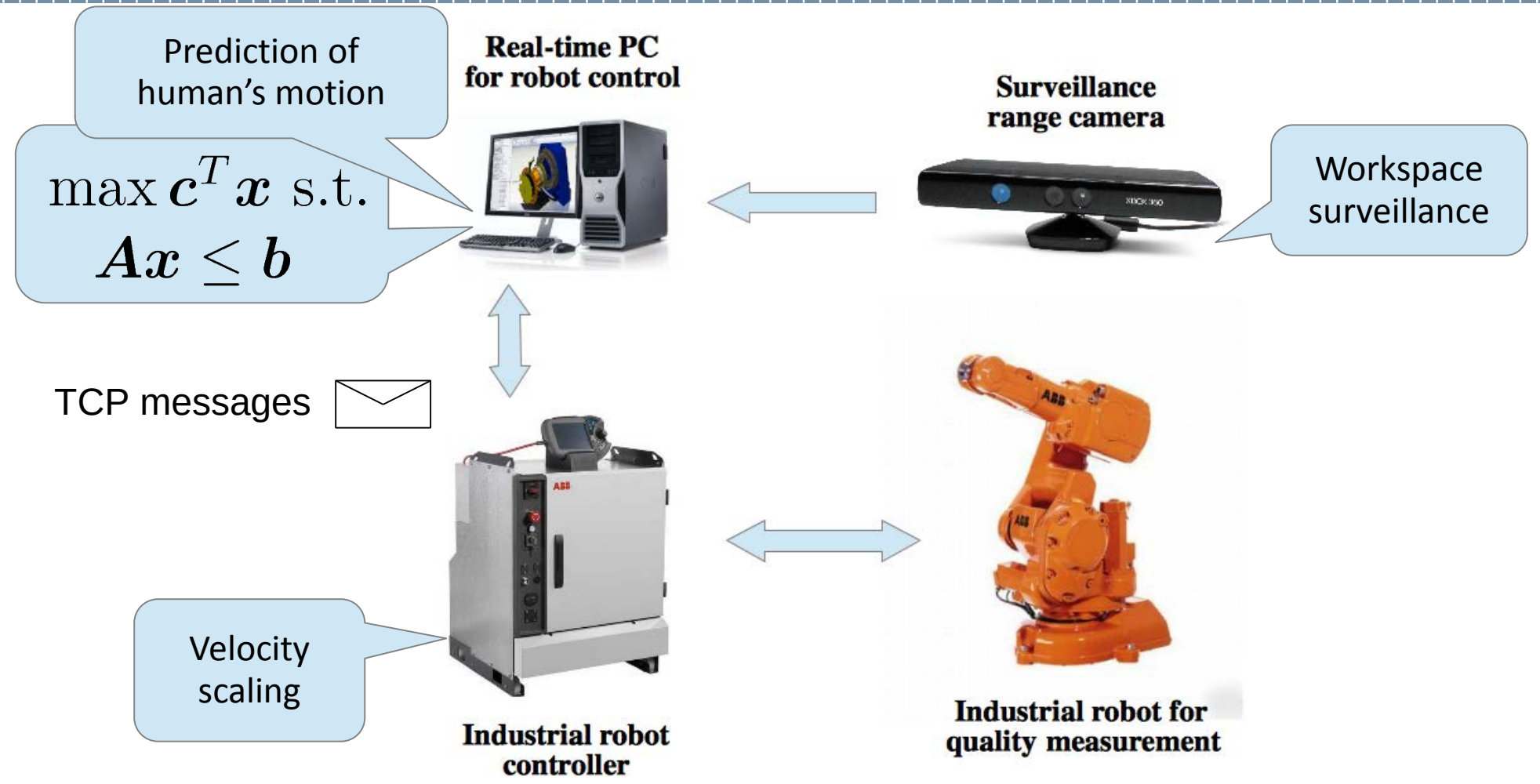
$$0 \leq \delta \leq 1$$

$$\tau_{k+1} = \tau_k + \Delta t \delta_k$$



Ragaglia, Zanchettin and Rocco, ICAR 2015

Safety-aware velocity scaling



Safety-aware velocity scaling

Safety-aware velocity scaling: redundant dofs

In case of redundant degrees of freedom, the following optimization problem

$$\max_{\delta,\dot{\boldsymbol{q}}} \delta$$

$$t_{b,i}\boldsymbol{E_i}\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}} \leq \boldsymbol{f_i}\left(\boldsymbol{q}\right), \forall i = 0, \ldots, n-1$$

$$\boldsymbol{J}\left(\boldsymbol{q}\right)\dot{\boldsymbol{q}} = \delta\dot{\boldsymbol{x}}$$

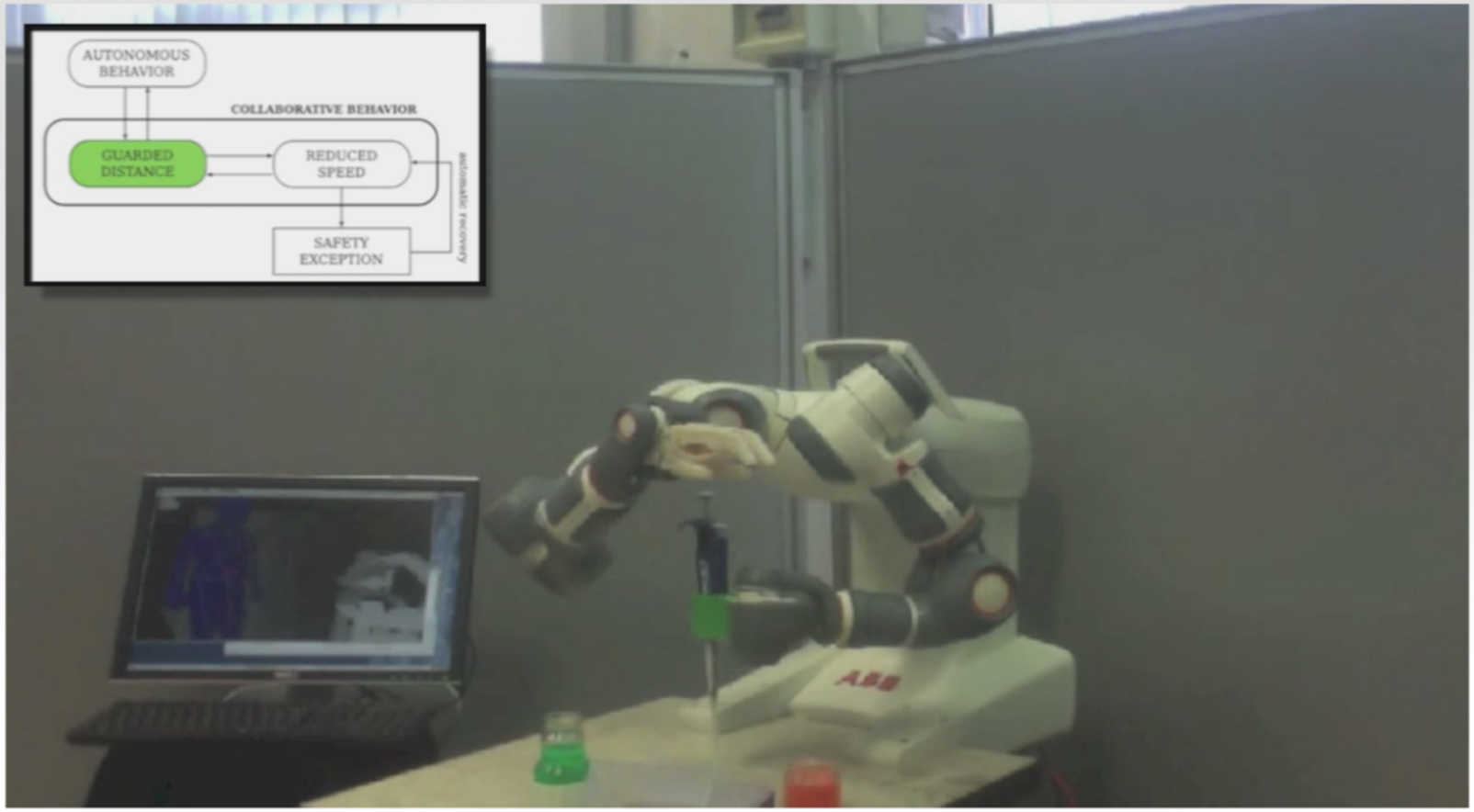
$$0 \leq \delta \leq 1$$

usually has **infinite solutions** in joint space.

The idea is to define an **additional optimization layer** to cope with null space velocities.

Zanchettin, Ceriani, Rocco, Ding and Matthias, IEEE T-ASE 2016

Safety-aware velocity scaling: redundant dofs



Reactive constraint-based motion planning

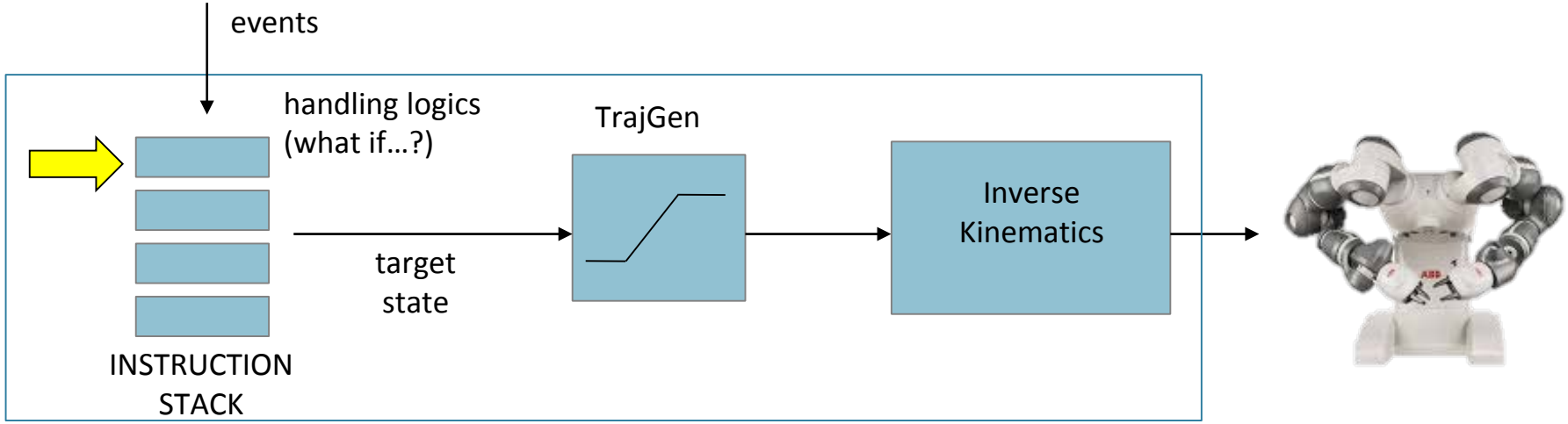
So far we have modulated the robot velocity along the specified path, consistently with this path.

A more general approach might be to **rethink the trajectory generation**, simultaneously accounting for various constraints, including the speed and sepation distance constraint.

This requires a sort of paradigm shift.

Industrial state of the art

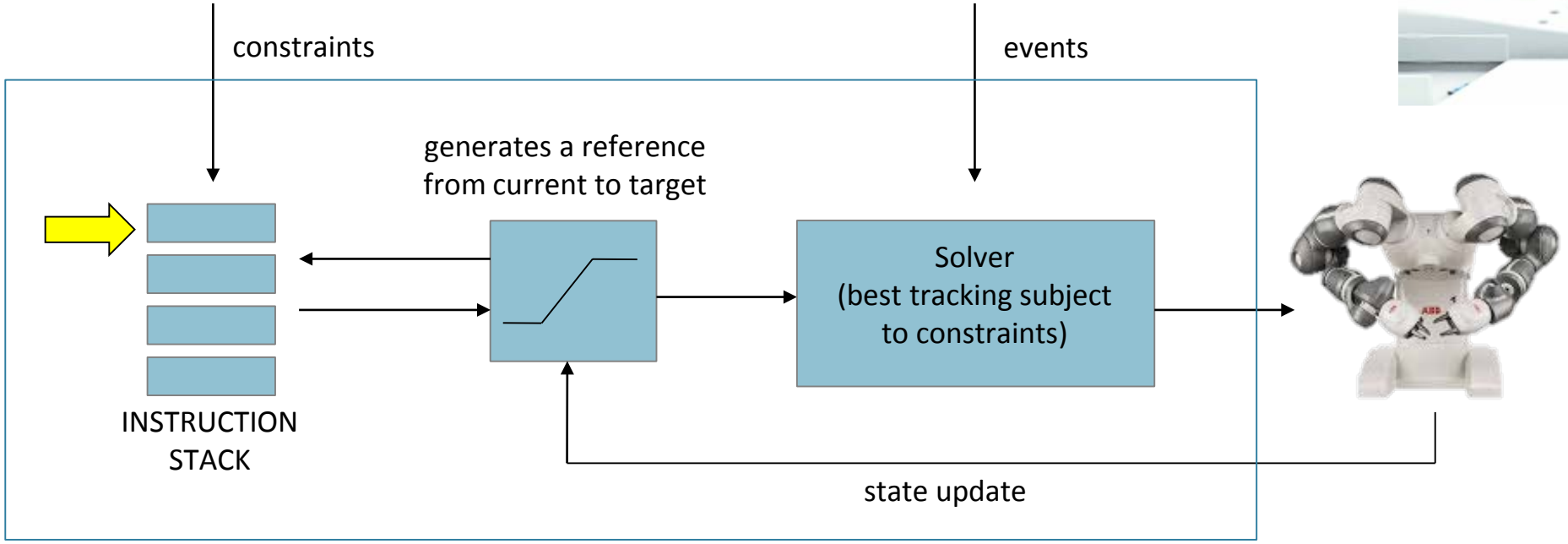
- all logics handled at programming level
- imperative motion programming (if-then-else)
- reduced reactivity (especially for real-time behaviour)



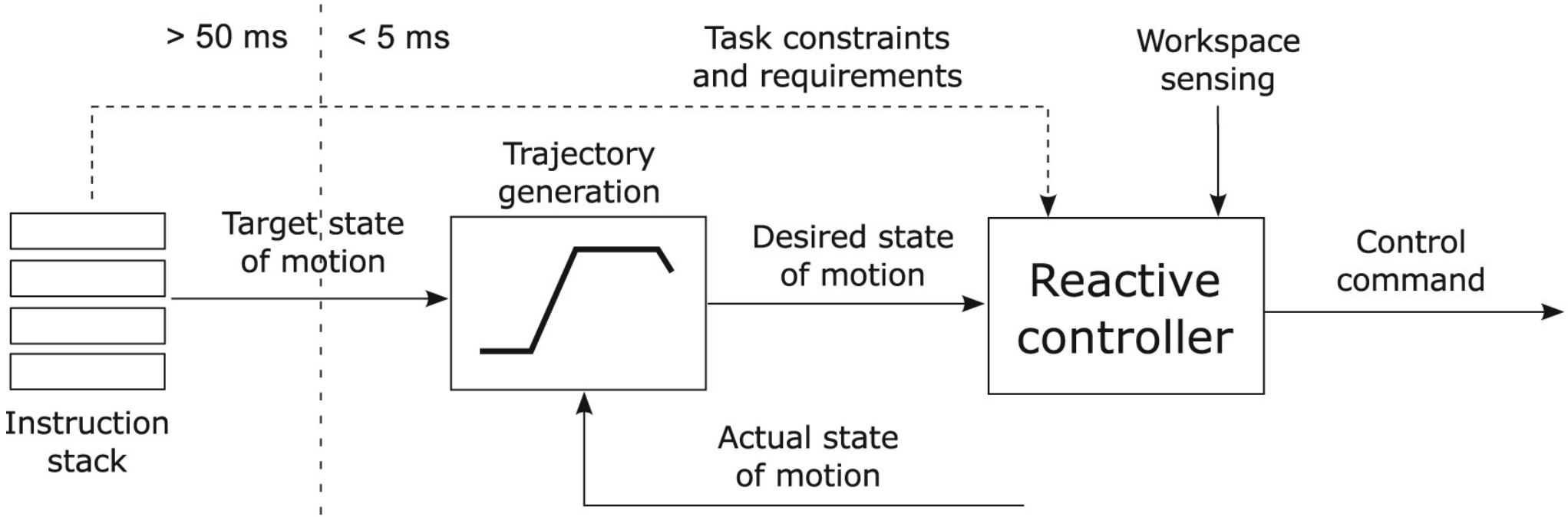
Reactive constraint-based motion planning

Our vision

- no need for logics
- declarative programming
- increased reactivity (especially for real-time behaviour)



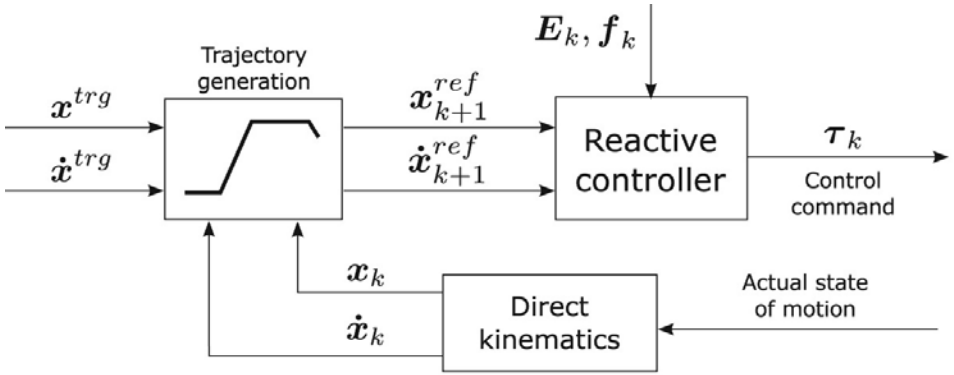
Reactive constraint-based motion planning



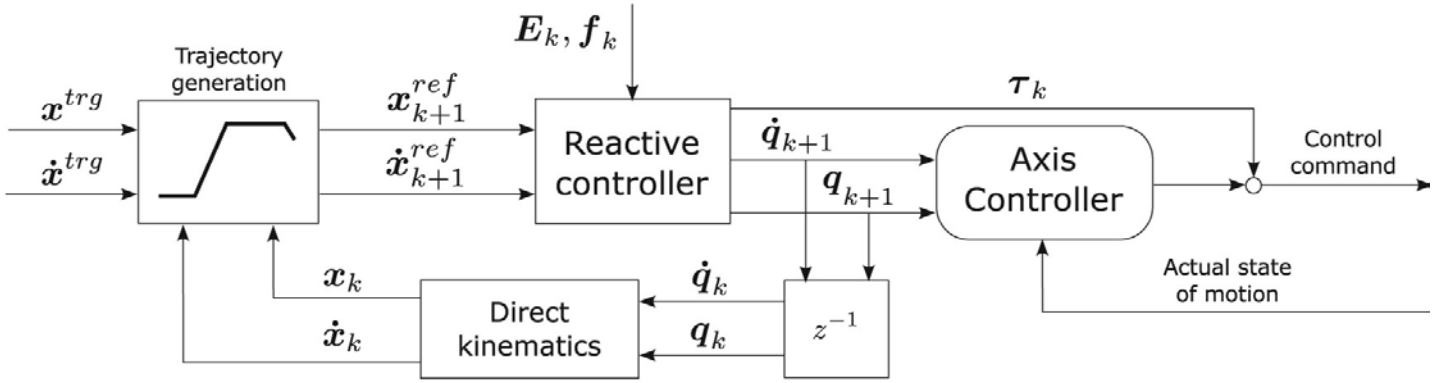
Zanchettin and Rocco, CEP 2017

Reactive constraint-based motion planning

As an actual feedback robot controller



As a trajectory generation algorithm on top of an existing axis controller



Most common arrangement

Optimization stages

$$\min_{\boldsymbol{u}_k} \left(\left\| \dot{\boldsymbol{x}}_{k+1} - \dot{\boldsymbol{x}}_{k+1}^{ref} \right\|_{\boldsymbol{Q}_v}^2 + \left\| \boldsymbol{x}_{k+1} - \boldsymbol{x}_{k+1}^{ref} \right\|_{\boldsymbol{Q}_p}^2 \right)$$

Cost metric: default trajectory tracking

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t\, \boldsymbol{J}_k \dot{\boldsymbol{q}}_k + \frac{\Delta t^2}{2} \left(\dot{\boldsymbol{J}}_k \dot{\boldsymbol{q}}_k + \boldsymbol{J}_k \boldsymbol{u}_k \right)$$

$$\dot{\boldsymbol{x}}_{k+1} = \boldsymbol{J}_k \dot{\boldsymbol{q}}_k + \Delta t \left(\dot{\boldsymbol{J}}_k \dot{\boldsymbol{q}}_k + \boldsymbol{J}_k \boldsymbol{u}_k \right)$$

2nd order forward kinematics discretization

$$\boldsymbol{q}^{inf} \leq \boldsymbol{q}_k + \Delta t\, \dot{\boldsymbol{q}}_k + \frac{1}{2} \Delta t^2\, \boldsymbol{u}_k \leq \boldsymbol{q}^{sup}$$

$$\dot{\boldsymbol{q}}^{inf} \leq \dot{\boldsymbol{q}}_k + \Delta t\, \boldsymbol{u}_k \leq \dot{\boldsymbol{q}}^{sup}$$

$$\boldsymbol{u}^{inf} \leq \boldsymbol{u}_k \leq \boldsymbol{u}^{sup}$$

Joint-space (feasibility) constraints

$$-\dot{\boldsymbol{x}}^{max} \leq \dot{\boldsymbol{x}}_{k+1} \leq \dot{\boldsymbol{x}}^{max}$$

$$-\ddot{\boldsymbol{x}}^{max} \leq \dot{\boldsymbol{J}}_k \dot{\boldsymbol{q}}_k + \boldsymbol{J}_k \boldsymbol{u}_k \leq \ddot{\boldsymbol{x}}^{max}$$

Task-space constraints

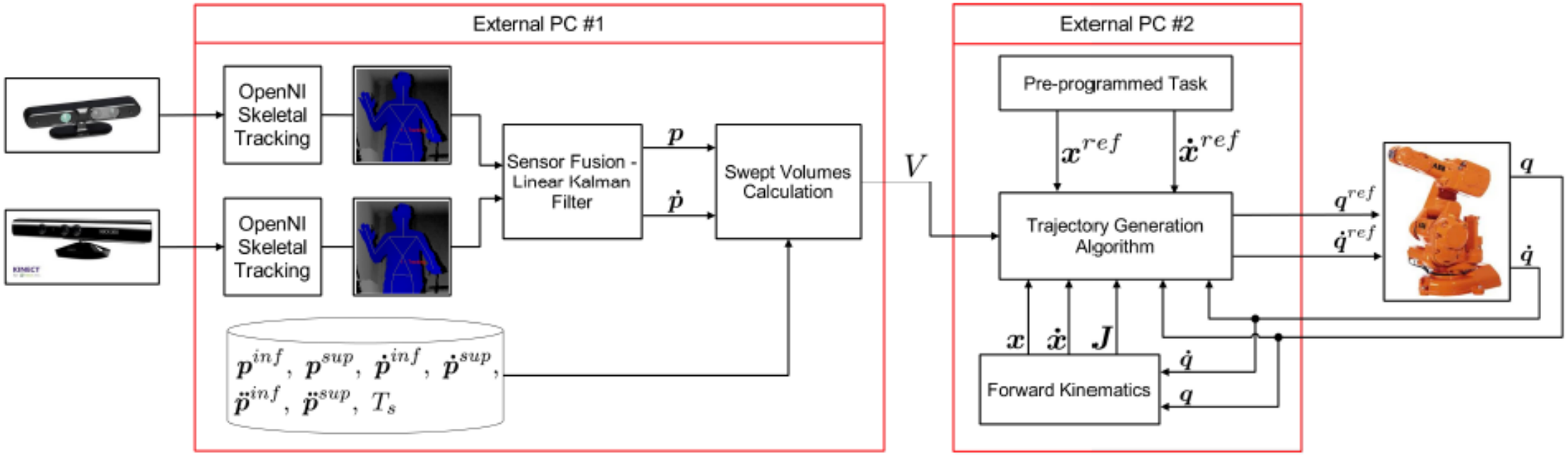
$$\forall\, \mathcal{V}_l \in \mathcal{SV},\ \forall\, \boldsymbol{r}_l^j \in \mathcal{V}_l \implies$$
$$\boldsymbol{E}_k \left(\boldsymbol{r}_l^j \right) \left(\dot{\boldsymbol{q}}_k + \Delta t\, \boldsymbol{u}_k \right) \leq \boldsymbol{f}_k \left(\boldsymbol{r}_l^j \right)$$

Safety Constraints

$$\dot{\boldsymbol{q}}_{k+1}^{ref} = \dot{\boldsymbol{q}}_k^{ref} + \Delta t\, \boldsymbol{u}_k$$
$$\boldsymbol{q}_{k+1}^{ref} = \boldsymbol{q}_k^{ref} + \Delta t\, \dot{\boldsymbol{q}}_k^{ref} + \frac{\Delta t^2}{2} \boldsymbol{u}_k$$

Joint references update

Reactive trajectory generation



- given a sequence of way points, a trajectory is computed that is collision free with respect to obstacles and human workers
- the manipulator can deviate from the pre-programmed trajectory in order to enforce the safety constraints

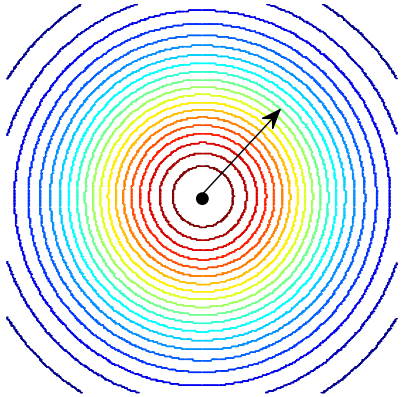
Ragaglia, Zanchettin and Rocco, Mechatronics 2018

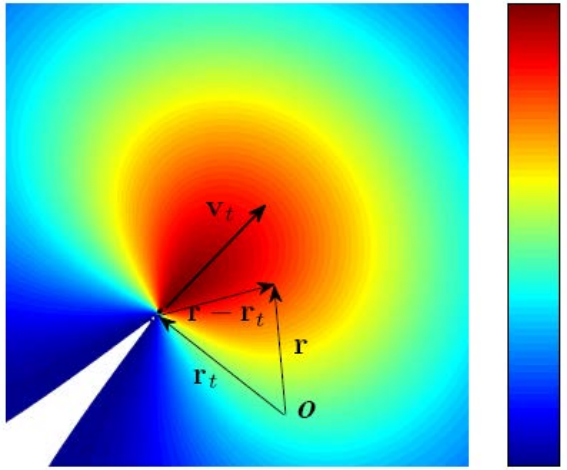
Reactive trajectory generation

Safety assessments

- How dangerous is a robot?
- Can we express its dangerousness quantitatively?
- Can we derive safety oriented control laws from such safety assessments?

Danger field





Static:
A decreasing function w.r.t. the distance from the "source of danger"

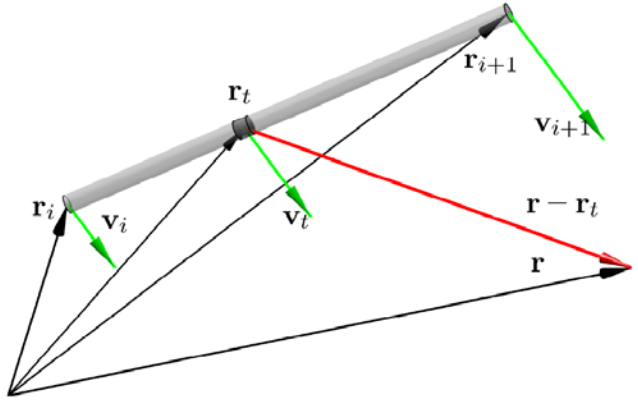
Kinetostatic:
Takes into account a velocity vector of the source of danger as well

An instance of KSDF:

$$DF(\mathbf{r}, \mathbf{r}_t, \mathbf{v}_t) = \frac{k_1}{\|\mathbf{r} - \mathbf{r}_t\|} + \frac{k_2 \|\mathbf{v}_t\| \left[\gamma + \cos \angle(\mathbf{r} - \mathbf{r}_t, \mathbf{v}_t)\right]}{\|\mathbf{r} - \mathbf{r}_t\|^2}$$

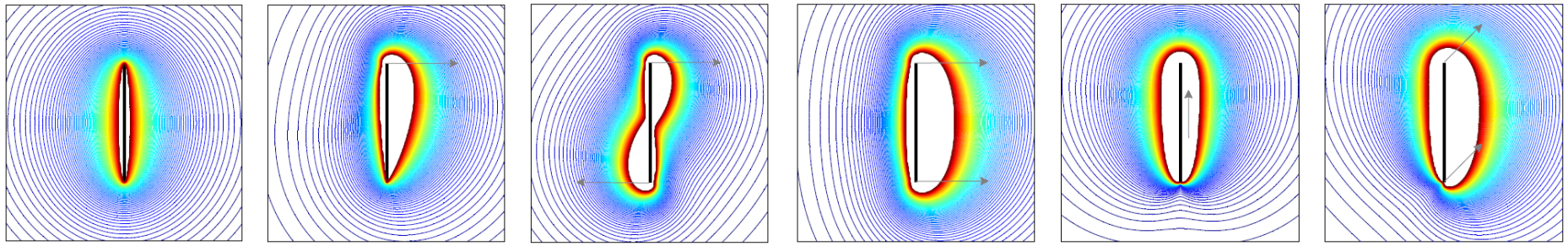
Cumulative danger field

$$CDF(\mathbf{r}, \mathbf{r}_i, \mathbf{v}_i, \mathbf{r}_{i+1}, \mathbf{v}_{i+1}) = k_1 \int_0^1 \frac{dt}{\sqrt{at^2 + bt + c}} + k_2\gamma \int_0^1 \frac{\sqrt{At^2 + Bt + C}}{at^2 + bt + c} dt + k_2 \int_0^1 \frac{Mt^2 + Nt + P}{(at^2 + bt + c)^{3/2}} dt$$



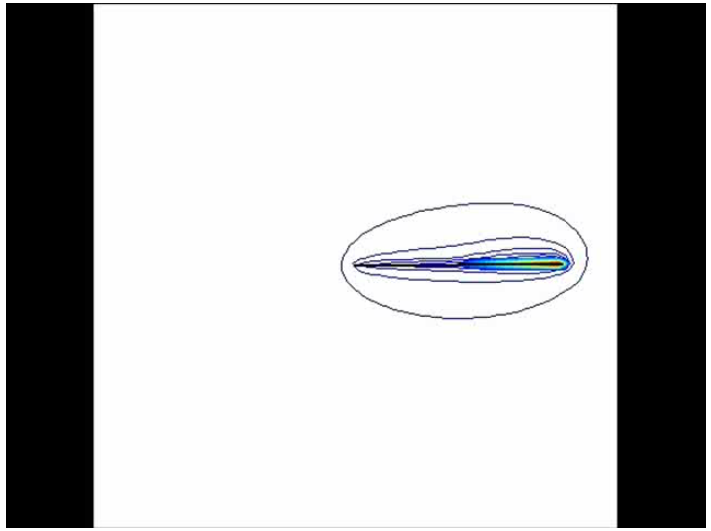
A line integral of the elementary danger field

Danger field induced by a single link:

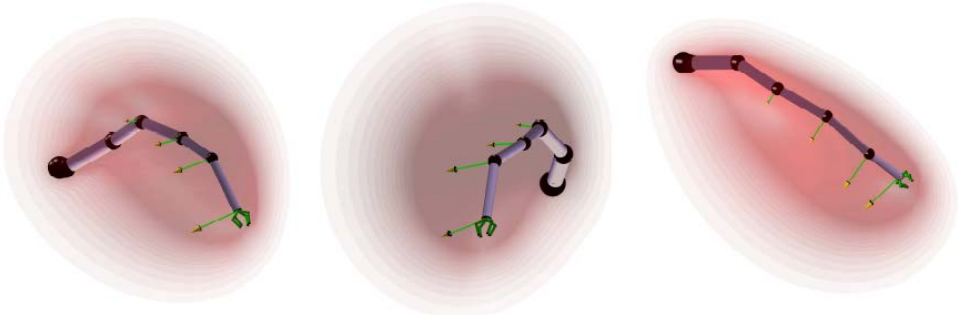


Cumulative danger field

Danger field induced by a 2 DOF robot:



Danger field induced by a 6 DOF robot:



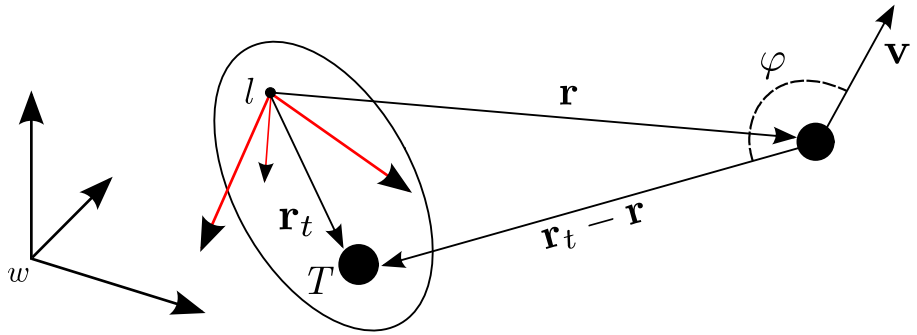
- Computable in closed form for the wire model of the robot
- Captures both position and velocity of the source of danger
- Captures the kinematic properties of the whole kinematic chain

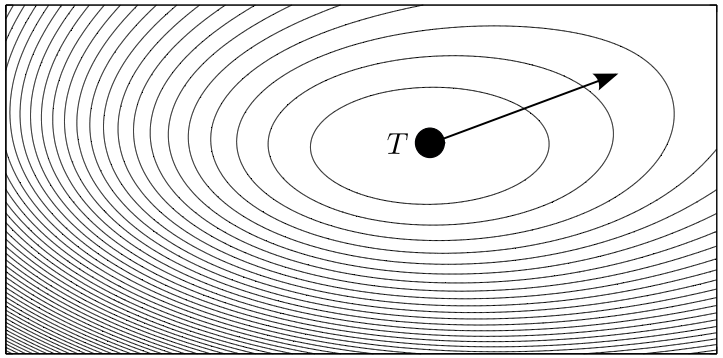
Lacevic, Rocco and Zanchettin, IEEE T-RO 2013

Safety field

From **danger** field to **safety** field

Reference frame **local** to the moving rigid body "source of danger"





Elementary Kinetostatic Safety Field (KSSF):

$$SF(\mathbf{r}_t, \mathbf{r}, \mathbf{v}) = \|\mathbf{r}_t - \mathbf{r}\|^2(\gamma - \langle \mathbf{r}_t - \mathbf{r}, \mathbf{v} \rangle)$$

Safety field

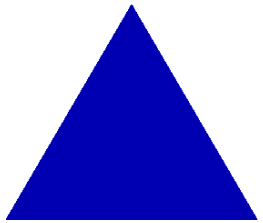
Property	Danger Field	Safety Field
Relative position	✓	✓
Relative velocity	X	✓
Length of danger source	✓	✓
Closed form solution	X	✓
Shape of danger source	X	✓

$$CSF(\mathbf{r}, \mathbf{v}) = c_1 x^2 + c_2 x + c_3 y^2 + c_4 y + c_5 z^2 + c_6 z + \\ + c_7 x^3 v_x + c_8 y^3 v_y + c_9 z^3 v_z + \ldots + c_{43}$$

Coefficients are **costant** and can be computed **off-line**

Cumulative kinetostatic safety field

$$CSF(\mathbf{r},\mathbf{v}) = \frac{1}{A}\iint_S SF(\mathbf{r}_S,\mathbf{r},\mathbf{v})\ dS$$



One triangle

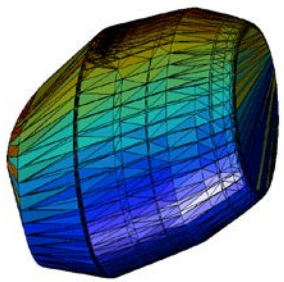
Closed form solution!

$$CSF(\mathbf{r},\mathbf{v}) = c_1x^2 + c_2x + c_3y^2 + c_4y + c_5z^2 + c_6z + \\ + c_7x^3v_x + c_8y^3v_y + c_9z^3v_z + \ldots + c_{43}$$

- Constant coefficients, depending on the specific triangular surface
- Coefficients computable off-line
- **r** and **v** implicitly contain the information about the source of danger absolute position and velocity

Cumulative kinetostatic safety field

$$CSF(\mathbf{r}, \mathbf{v}) = \frac{1}{N} \sum_{i=1}^{N} CSF_i(\mathbf{r}, \mathbf{v})$$



Mesh of N triangles

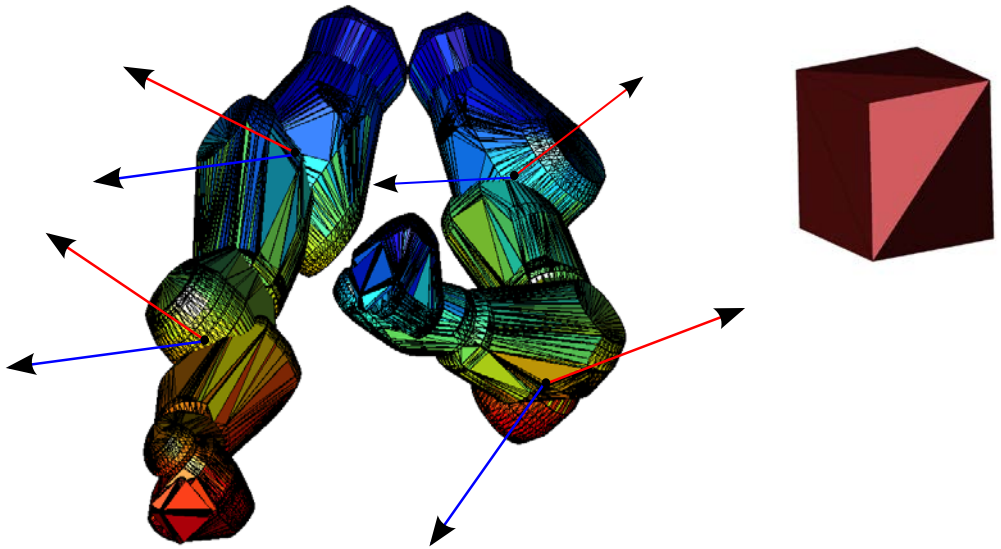
- Constant coefficients depending on the rigid body shape and size, computable off-line
- No upper bound in the complexity of the triangular mesh of the rigid body

SF COMPUTATION TIME

Source of Danger	Mesh elements	Time
Box	12	1,53 μs
Robot link	10.662	1,41 μs

Online computation time does not depend on the complexity of the shape!

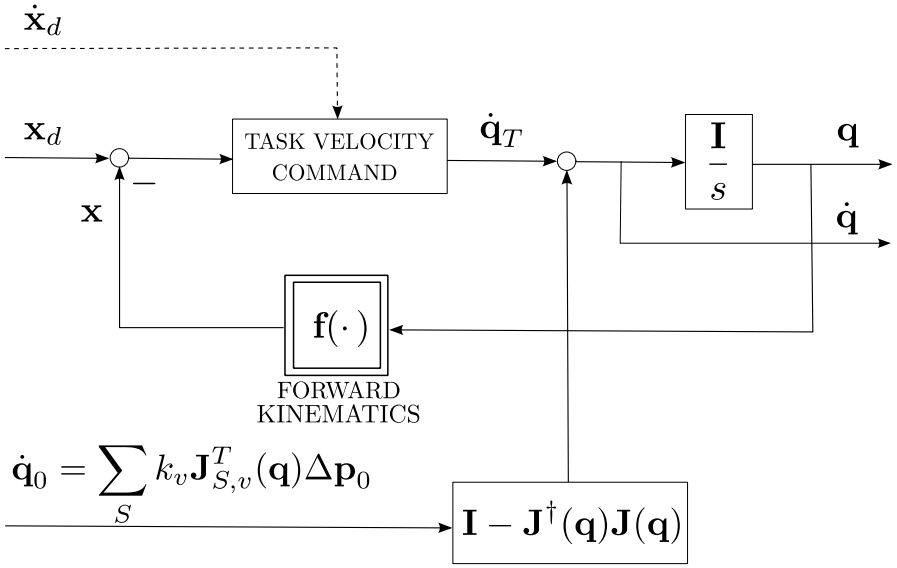
Safety field based control

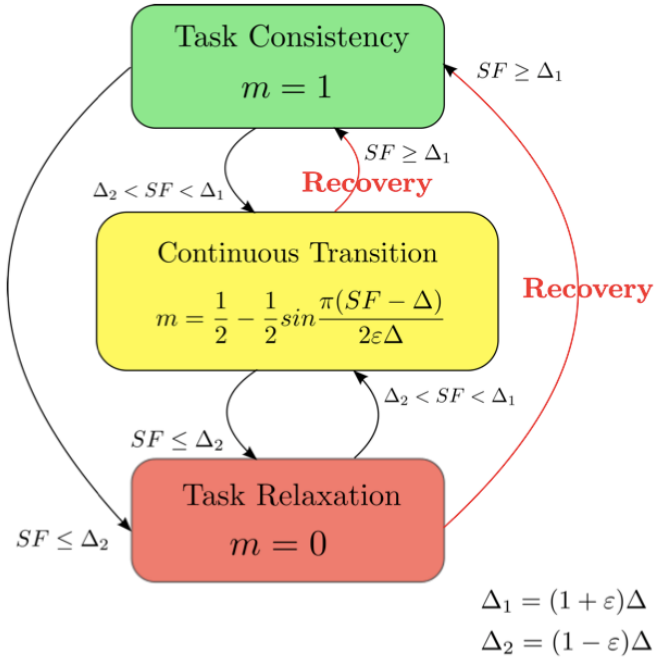


Evasive displacement $\Delta \mathbf{p}_0 = \mathbf{R}_{w \leftarrow l} \, \frac{1}{CSF(\mathbf{r}, \mathbf{v})} \frac{\boldsymbol{\nabla} CSF(\mathbf{r}, \mathbf{v})}{\|\boldsymbol{\nabla} CSF(\mathbf{r}, \mathbf{v})\|}$

n sources of danger $\Delta \mathbf{p}_0 = \sum_{i=1}^{n} \Delta \mathbf{p}_{0,i}$

Safety field based control





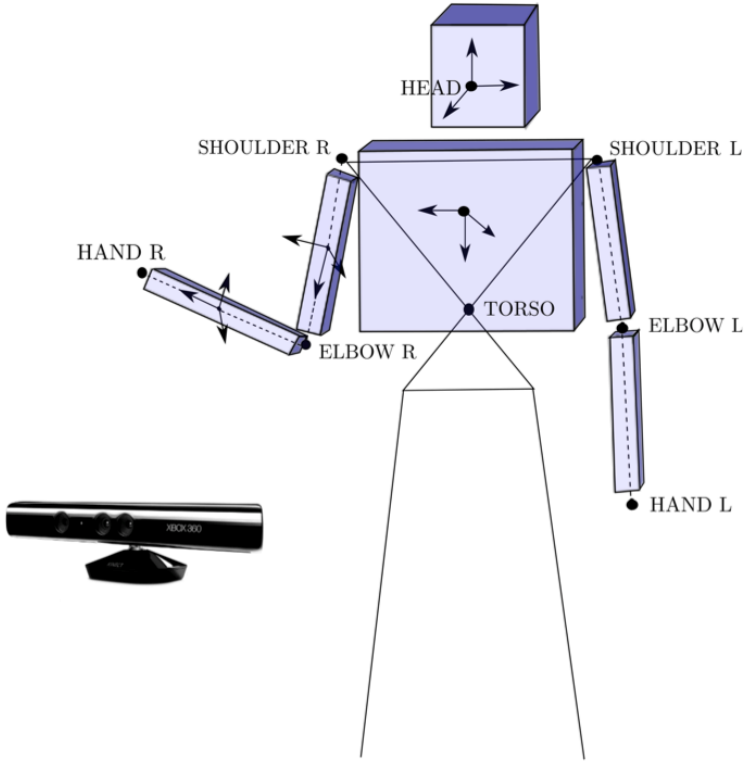
Control features

- Obstacle avoidance based on null-space projection
- Task can be temporarily relaxed to avoid obstacle

$$\dot{\mathbf{q}} = m\dot{\mathbf{q}}_T + [\mathbf{I} - m\mathbf{J}^\dagger(\mathbf{q})\mathbf{J}(\mathbf{q})]\dot{\mathbf{q}}_0$$

Parigi Polverini, Zanchettin and Rocco, RCIM 2017

Safety field based control



$$\dot{\mathbf{q}}_0 = \dot{\mathbf{q}}_0^{arm} + \dot{\mathbf{q}}_0^{human}$$

Safety field based control

POLITECNICO
MILANO 1863