Some considerations on optimisation-based control in robotics

Many problems, some ideas towards solutions

Vincent Padois – vincent.padois@inria.fr
Senior research scientist
Inria Bordeaux Sud-Ouest, Auctus

R4 2021,
Talence – 2021/05/10

Interactive robots do not exist for real

Real-world ...

Basic locomotion and manipulation skills

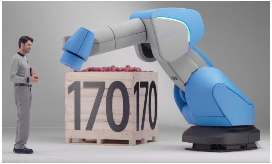
Advanced locomotion skills

Cognitive and physical interactions

Interactive robots do not exist for real

... vs Laboratory science and technology

Advanced control but no living bodies around



How many (trully) collaborative robots have you seen in the industry ?

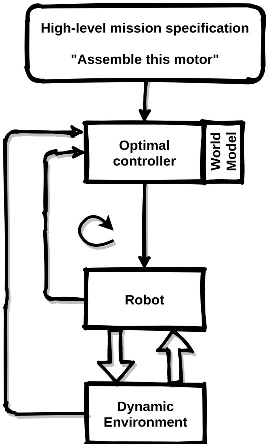
Why is it so ?

The world is dynamic, complex and hard to predict (impact in 6s)

Outline of the presentation

(Reactive) Optimal control

Ideally, solve reactively ...



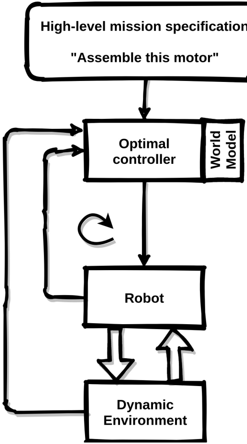
$$\min_{t_0, t_f, \boldsymbol{x}(t), \boldsymbol{u}(t)} \underbrace{J_b(t_0, t_f, x(t_0), x(t_f))}_{boundary\ \ objective\ \ function} + \underbrace{\int_{t_0}^{t_f} J_i(s, x(s), u(s))ds}_{integral\ \ objective\ \ function}$$

subject to :

- Dynamics : $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t))$
- Path constraints : $\boldsymbol{h}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \leq \boldsymbol{0}$
- State constraints : $\boldsymbol{x}_l(t) \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_u(t)$
- Control bounds : $\boldsymbol{u}_l(t) \leq \boldsymbol{u}(t) \leq \boldsymbol{u}_u(t)$

(Reactive) Optimal control

Ideally, solve reactively ...



$$\min_{t_0, t_f, \boldsymbol{x}(t), \boldsymbol{u}(t)} \underbrace{J_b(t_0, t_f, x(t_0), x(t_f))}_{boundary\ \ objective\ \ function} + \underbrace{\int_{t_0}^{t_f} J_i(s, x(s), u(s))ds}_{integral\ \ objective\ \ function}$$

... but in practice

- infinite dimensional problem
- can generally not be solved, even once

$\hookrightarrow$ transformed in a finite dimensional problem : non linear program / constrained parameter optimization

$\hookrightarrow$ hard to solve, cannot be solved reactively

Looking closer

In dynamic environments, $\boldsymbol{x}(t) = \{\boldsymbol{x}_{rob}(t), \boldsymbol{x}_{env}(t)\}$
$\hookrightarrow$ requires **perception** for the state of the environment $\boldsymbol{x}_{env}(t)$
$\hookrightarrow$ no control over $\boldsymbol{x}_{env}(t) \rightarrow$ reactive planning needed

Looking closer

In dynamic environments, $\boldsymbol{x}(t) = \{\boldsymbol{x}_{rob}(t), \boldsymbol{x}_{env}(t)\}$
$\hookrightarrow$ requires **perception** for the state of the environment $\boldsymbol{x}_{env}(t)$
$\hookrightarrow$ no control over $\boldsymbol{x}_{env}(t)$ $\rightarrow$ reactive planning needed

$\hookrightarrow$ compute an optimal control input trajectory $\boldsymbol{\tau}(t)$ **at each control instant** given

Looking closer

In dynamic environments, $\boldsymbol{x}(t) = \{\boldsymbol{x}_{rob}(t), \boldsymbol{x}_{env}(t)\}$
$\hookrightarrow$ requires **perception** for the state of the environment $\boldsymbol{x}_{env}(t)$
$\hookrightarrow$ no control over $\boldsymbol{x}_{env}(t)$ $\rightarrow$ reactive planning needed

$\hookrightarrow$ compute an optimal control input trajectory $\boldsymbol{\tau}(t)$ **at each control instant** given

- Control objectives : $\{\boldsymbol{H}_{1,f}, \dots, \boldsymbol{H}_{n_o,f}\}$

Looking closer

In dynamic environments, $\boldsymbol{x}(t) = \{\boldsymbol{x}_{rob}(t), \boldsymbol{x}_{env}(t)\}$
$\hookrightarrow$ requires **perception** for the state of the environment $\boldsymbol{x}_{env}(t)$
$\hookrightarrow$ no control over $\boldsymbol{x}_{env}(t) \rightarrow$ reactive planning needed

$\hookrightarrow$ compute an optimal control input trajectory $\boldsymbol{\tau}(t)$ **at each control instant** given

- Control objectives : $\{\boldsymbol{H}_{1,f}, \dots, \boldsymbol{H}_{n_o,f}\}$
- (Non-linear) Dynamics of the system :
  - $\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \boldsymbol{b}(\boldsymbol{q}, \boldsymbol{\nu}) = \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{\tau}\ (+\sum_i^{n_c} \boldsymbol{J}_{c_i}^T(\boldsymbol{q})\boldsymbol{f}_{c_i})$
  - $\boldsymbol{v}_i = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{\nu}} \ \ \forall i \in [1, n_o]$ and $\boldsymbol{v}_i := \dot{\boldsymbol{H}}_i$

Looking closer

In dynamic environments, $\boldsymbol{x}(t) = \{\boldsymbol{x}_{rob}(t), \boldsymbol{x}_{env}(t)\}$
$\hookrightarrow$ requires **perception** for the state of the environment $\boldsymbol{x}_{env}(t)$
$\hookrightarrow$ no control over $\boldsymbol{x}_{env}(t)$ $\rightarrow$ reactive planning needed

$\hookrightarrow$ compute an optimal control input trajectory $\boldsymbol{\tau}(t)$ **at each control instant** given

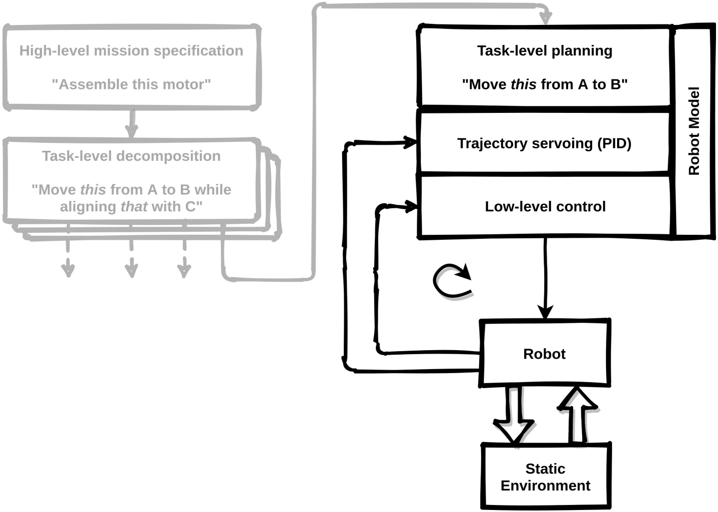
- Control objectives : $\{\boldsymbol{H}_{1,f}, \ldots, \boldsymbol{H}_{n_o,f}\}$
- (Non-linear) Dynamics of the system :
  - $\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \boldsymbol{b}(\boldsymbol{q}, \boldsymbol{\nu}) = \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{\tau}\ (+\sum_i^{n_c} \boldsymbol{J}_{c_i}^T(\boldsymbol{q})\boldsymbol{f}_{c_i})$
  - $\boldsymbol{v}_i = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{\nu}}\ \ \forall i \in [1, n_o]$ and $\boldsymbol{v}_i := \dot{\boldsymbol{H}}_i$
- Constraints :
  - $\boldsymbol{\tau}_l \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_u$
  - $\dot{\boldsymbol{\tau}}_l \leq \dot{\boldsymbol{\tau}} \leq \dot{\boldsymbol{\tau}}_u$
  - $\boldsymbol{q}_l \leq \boldsymbol{q} \leq \boldsymbol{q}_u$
  - $\dot{\boldsymbol{\nu}}_l \leq \dot{\boldsymbol{\nu}} \leq \dot{\boldsymbol{\nu}}_u$
  - $\boldsymbol{h}(\boldsymbol{x}_{env}, \boldsymbol{q}) \leq \boldsymbol{0}$
  - ...

$\hookrightarrow$ **very complex and computationnally demanding control / optimization problem**

Outline of the presentation

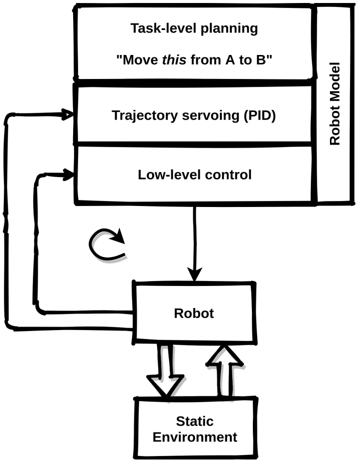
Optimal control vs real-life

Historically in the industry, the problem left to robots is simplified



Optimal control vs real-life

Static environment → reactivity not required at the task planning level ...

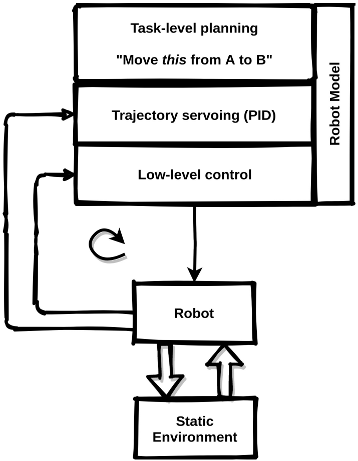


... as constraints are met

- offline, through planning
- a posteriori through emergency stops or stereotypical safety zones definition

Optimal control vs real-life

Static environment → reactivity not required at the task planning level ...



... as constraints are met

- offline, through planning
- a posteriori through emergency stops or stereotypical safety zones definition

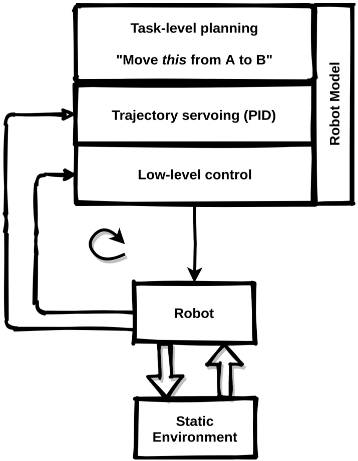
Yet finding a control trajectory is complex

↪ Decouple planning and control

- Plan for $\boldsymbol{q}(t)$ or $\boldsymbol{H}(t)$
- Perform trajectory servoing and low level-control

Optimal control vs real-life

Static environment $\rightarrow$ reactivity not required at the task planning level ...



... as constraints are met

- offline, through planning
- a posteriori through emergency stops or stereotypical safety zones definition

Yet finding a control trajectory is complex

$\hookrightarrow$ Decouple planning and control

- Plan for $\boldsymbol{q}(t)$ or $\boldsymbol{H}(t)$
- Perform trajectory servoing and low level-control

Still too complex !

- Simplification based on an underestimation of the true robot capacities

$\hookrightarrow$ the industry is full of oversized and dangerous robots

- Highly expert manual tuning required

$\hookrightarrow$ robots are not the promised versatile tools

Illustration with the Franka Emika Panda Robot

Constants

Limits in the Cartesian space are as follows:

Name	Translation	Rotation	Elbow
$\dot{p}_{max}$	1.7000 $\frac{\text{m}}{\text{s}}$	2.5000 $\frac{\text{rad}}{\text{s}}$	2.1750 $\frac{rad}{\text{s}}$
$\ddot{p}_{max}$	13.0000 $\frac{\text{m}}{\text{s}^2}$	25.0000 $\frac{\text{rad}}{\text{s}^2}$	10.0000 $\frac{rad}{\text{s}^2}$
$\dddot{p}_{max}$	6500.0000 $\frac{\text{m}}{\text{s}^3}$	12500.0000 $\frac{\text{rad}}{\text{s}^3}$	5000.0000 $\frac{rad}{\text{s}^3}$

Joint space limits are:

Name	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Unit
q_{max}	2.8973	1.7628	2.8973	-0.0698	2.8973	3.7525	2.8973	rad
q_{min}	-2.8973	-1.7628	-2.8973	-3.0718	-2.8973	-0.0175	-2.8973	rad
$\dot{q}_{max}$	2.1750	2.1750	2.1750	2.1750	2.6100	2.6100	2.6100	$\frac{\text{rad}}{\text{s}}$
$\ddot{q}_{max}$	15	7.5	10	12.5	15	20	20	$\frac{\text{rad}}{\text{s}^2}$
$\dddot{q}_{max}$	7500	3750	5000	6250	7500	10000	10000	$\frac{\text{rad}}{\text{s}^3}$
$\tau_{j\,max}$	87	87	87	87	12	12	12	Nm
$\dot{\tau}_{j\,max}$	1000	1000	1000	1000	1000	1000	1000	$\frac{\text{Nm}}{\text{s}}$

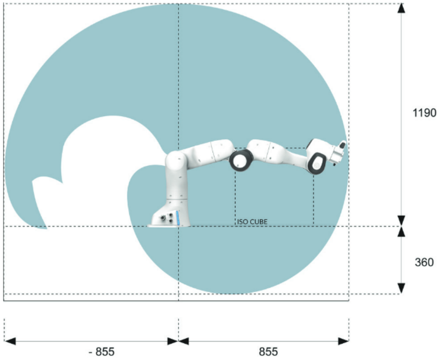


Illustration with the Franka Emika Panda Robot

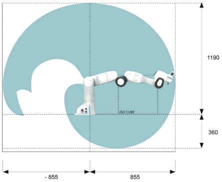
Constants

Limits in the Cartesian space are as follows:

Name	Translation	Rotation	Elbow
$\dot{p}_{max}$	1.7000 $\frac{m}{s}$	2.5000 $\frac{rad}{s}$	2.1750 $\frac{rad}{s}$
$\ddot{p}_{max}$	13.0000 $\frac{m}{s^2}$	25.0000 $\frac{rad}{s^2}$	10.0000 $\frac{rad}{s^2}$
$\dddot{p}_{max}$	6500.0000 $\frac{m}{s^3}$	12500.0000 $\frac{rad}{s^3}$	5000.0000 $\frac{rad}{s^3}$

Joint space limits are:

Name	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Unit
q_{max}	2.8973	1.7628	2.8973	-0.0698	2.8973	3.7525	2.8973	rad
q_{min}	-2.8973	-1.7628	-2.8973	-3.0718	-2.8973	-0.0175	-2.8973	rad
$\dot{q}_{max}$	2.1750	2.1750	2.1750	2.1750	2.6100	2.6100	2.6100	$\frac{rad}{s}$
$\ddot{q}_{max}$	15	7.5	10	12.5	15	20	20	$\frac{rad}{s^2}$
$\dddot{q}_{max}$	7500	3750	5000	6250	7500	10000	10000	$\frac{rad}{s^3}$
$\tau_{j\,max}$	87	87	87	87	12	12	12	Nm
$\dot{\tau}_{j\,max}$	1000	1000	1000	1000	1000	1000	1000	$\frac{Nm}{s}$

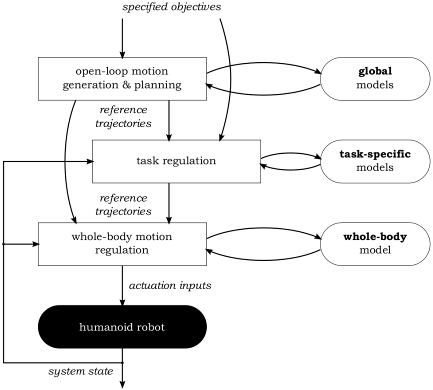


↪ **Curse of "collaborative" robotics**

- Safety in the collaboration requires small robots and controlled stops
- Small robots capabilities are small
- Underestimating the capabilities of small robots leads to "not much" capabilities
- Potentially safe robots are mostly useless

Optimal control vs complex robots (e.g. humanoids)

For systems making intermittent contacts with the environment (e.g. humanoids walking)...



[Ibanez 2017]

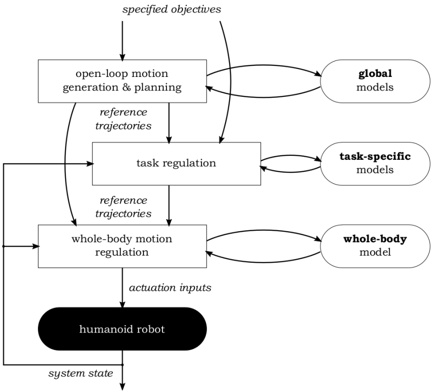
... mostly two solutions

- Sequential simplified planning problem solving from contact sequence to center of mass trajectory under balance constraints and in purely static environment (plan once)
- Stereotypical walking gaits (planned once) on flat grounds and online planar trajectory adaptation

\+ Trajectory servoing and multi-task whole-body control

Optimal control vs complex robots (e.g. humanoids)

For systems making intermittent contacts with the environment (e.g. humanoids walking)...



[Ibanez 2017]

Difficulties

- Planning performed with advanced models is costly $\rightarrow$ no reactivity
- Simplified models do not account for the true capabilities of the system

$\hookrightarrow$ underestimation / overstimation $\rightarrow$ manual tuning

- Humanoids can't do much in real life

Outline of the presentation

Outline of the presentation

Robot low-level control as an optimisation problem

In a dynamic environment, performance and safety requires to embed constraints in the low-level control problem : at each control instant, find the actuation torque $\boldsymbol{\tau}^*$ optimizing under constraints some objective related task $\boldsymbol{v}^* = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{\nu}$

Robot low-level control as an optimisation problem

In a dynamic environment, performance and safety requires to embed constraints in the low-level control problem : at each control instant, find the actuation torque $\boldsymbol{\tau}^*$ optimizing under constraints some objective related task $\boldsymbol{v}^* = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{\nu}$

- Equation of motion and joint space to task space mappings : **equalities**
  $\hookrightarrow$ can be solved using Linear Algebra
  - $\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \boldsymbol{b}(\boldsymbol{q},\boldsymbol{\nu}) = \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{\tau} \ (+\sum_i^{n_c} \boldsymbol{J}_{c_i}^T(\boldsymbol{q})\boldsymbol{f}_{c_i})$
  - $\boldsymbol{v}_i = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{\nu}} \ \ \forall i \in [1, n_o]$ and $\boldsymbol{v}_i := \dot{\boldsymbol{H}}_i$

Robot low-level control as an optimisation problem

In a dynamic environment, performance and safety requires to embed constraints in the low-level control problem : at each control instant, find the actuation torque $\boldsymbol{\tau}^*$ optimizing under constraints some objective related task $\boldsymbol{v}^* = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{\nu}$

- Equation of motion and joint space to task space mappings : **equalities**
  $\hookrightarrow$ can be solved using Linear Algebra
  - $\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \boldsymbol{b}(\boldsymbol{q}, \boldsymbol{\nu}) = \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{\tau}\ (+\sum_i^{n_c} \boldsymbol{J}_{c_i}^T(\boldsymbol{q})\boldsymbol{f}_{c_i})$
  - $\boldsymbol{v}_i = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{\nu}} \ \ \forall i \in [1, n_o]$ and $\boldsymbol{v}_i := \dot{\boldsymbol{H}}_i$
- **Standard** IVK and operational space control **approaches***
  $\hookrightarrow$ solution based on $\boldsymbol{J}^+$ and null-space projections $\dot{\boldsymbol{\nu}} = \boldsymbol{J}^+(\boldsymbol{q})\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{J}^+\boldsymbol{J})\dot{\boldsymbol{\nu}}_0$

. *see the work of [Liégeois 1977], [Khatib 1987], [Siciliano 1991], [Chiaverini 1997], [Mansard 2009], [Flacco 2012],...

Robot low-level control as an optimisation problem

In a dynamic environment, performance and safety requires to embed constraints in the low-level control problem : at each control instant, find the actuation torque $\boldsymbol{\tau}^*$ optimizing under constraints some objective related task $\boldsymbol{v}^* = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{\nu}$

- Equation of motion and joint space to task space mappings : **equalities**
  $\hookrightarrow$ can be solved using Linear Algebra
  - $\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \boldsymbol{b}(\boldsymbol{q}, \boldsymbol{\nu}) = \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{\tau}\ (+\sum_i^{n_c} \boldsymbol{J}_{c_i}^T(\boldsymbol{q})\boldsymbol{f}_{c_i})$
  - $\boldsymbol{v}_i = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{\nu}}\ \ \forall i \in [1, n_o]$ and $\boldsymbol{v}_i := \dot{\boldsymbol{H}}_i$
- **Standard** IVK and operational space control **approaches***
  $\hookrightarrow$ solution based on $\boldsymbol{J}^+$ and null-space projections $\dot{\boldsymbol{\nu}} = \boldsymbol{J}^+(\boldsymbol{q})\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{J}^+\boldsymbol{J})\dot{\boldsymbol{\nu}}_0$
- Some limits on the system cannot or should never be crossed : **inequalities**
  $\hookrightarrow$ cannot be accounted for properly using Linear Algebra only

$$\boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\nu})\mathbb{X} \leq \boldsymbol{h}(\boldsymbol{q}, \boldsymbol{\nu})$$

. *see the work of [Liégeois 1977], [Khatib 1987], [Siciliano 1991], [Chiaverini 1997], [Mansard 2009], [Flacco 2012],...

Robot low-level control as an optimisation problem

In a dynamic environment, performance and safety requires to embed constraints in the low-level control problem : at each control instant, find the actuation torque $\boldsymbol{\tau}^*$ optimizing under constraints some objective related task $\boldsymbol{v}^* = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{\nu}$

- Equation of motion and joint space to task space mappings : **equalities**
  $\hookrightarrow$ can be solved using Linear Algebra
  - $\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \boldsymbol{b}(\boldsymbol{q},\boldsymbol{\nu}) = \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{\tau}\ (+\sum_i^{n_c} \boldsymbol{J}_{c_i}^T(\boldsymbol{q})\boldsymbol{f}_{c_i})$
  - $\boldsymbol{v}_i = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{\nu}}\ \ \forall i \in [1, n_o]$ and $\boldsymbol{v}_i := \dot{\boldsymbol{H}}_i$
- **Standard** IVK and operational space control **approaches***
  $\hookrightarrow$ solution based on $\boldsymbol{J}^+$ and null-space projections $\dot{\boldsymbol{\nu}} = \boldsymbol{J}^+(\boldsymbol{q})\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{J}^+\boldsymbol{J})\dot{\boldsymbol{\nu}}_0$
- Some limits on the system cannot or should never be crossed : **inequalities**
  $\hookrightarrow$ cannot be accounted for properly using Linear Algebra only

$$\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\nu})\mathbb{X} \leq \boldsymbol{h}(\boldsymbol{q},\boldsymbol{\nu})$$

- These constraints are linear wrt control variables : **convex solution space**
  $\hookrightarrow$ convex optimization (LQP) is a powerful tool to solve optimally the reactive control problem.

. *see the work of [Liégeois 1977], [Khatib 1987], [Siciliano 1991], [Chiaverini 1997], [Mansard 2009], [Flacco 2012],...

Robot low-level control as an optimisation problem

3 reasons why Quadratic Programs are better than explicit Jacobian inversions

Robot low-level control as an optimisation problem

3 reasons why Quadratic Programs are better than explicit Jacobian inversions

1. Leave your robot alone
   - Methods based on $\boldsymbol{J}^{+}$ forces constraints to be treated as tasks $\rightarrow$ active avoidance
   - QP allows to consider constraints as such $\rightarrow$ passive avoidance

Robot low-level control as an optimisation problem

3 reasons why Quadratic Programs are better than explicit Jacobian inversions

1. Leave your robot alone
   - Methods based on $\boldsymbol{J}^{+}$ forces constraints to be treated as tasks $\rightarrow$ active avoidance
   - QP allows to consider constraints as such $\rightarrow$ passive avoidance
2. More constraints than DoFs : choose which one to consider at each time
   - Methods based on $\boldsymbol{J}^{+}$ use context specific heuristics to do so
   - QP comes with an optimal active constraints determination algorithm

Robot low-level control as an optimisation problem

3 reasons why Quadratic Programs are better than explicit Jacobian inversions

1. Leave your robot alone
   - Methods based on $\boldsymbol{J}^{+}$ forces constraints to be treated as tasks $\rightarrow$ active avoidance
   - QP allows to consider constraints as such $\rightarrow$ passive avoidance
2. More constraints than DoFs : choose which one to consider at each time
   - Methods based on $\boldsymbol{J}^{+}$ use context specific heuristics to do so
   - QP comes with an optimal active constraints determination algorithm
3. Infeasibilty can't be ignored
   - Methods based on $\boldsymbol{J}^{+}$ can solve infeasible problems $\rightarrow$ constraints violation
   - QP can't be solved if infeasible $\rightarrow$ deal with this problem first
     [Rubrecht 2012, Meguenani 2017b, Del Prete 2018]

Constraints compliance as a control feature

For example :

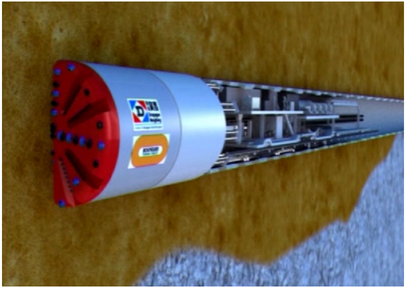
$$\boldsymbol{\tau}_{k+1}{}^{*} = \underset{\boldsymbol{\tau}_{k+1}, \ddot{\boldsymbol{q}}_{k+1}}{\arg\min} \left\| \boldsymbol{obj}\left(\ddot{\boldsymbol{q}}_{k+1}, \ddot{\boldsymbol{x}}^{*}_{k+1}\right) \right\|^2_{\boldsymbol{Q}_t} + \epsilon \left\| \begin{bmatrix} \boldsymbol{\tau}_{k+1} \\ \ddot{\boldsymbol{q}}_{k+1} \end{bmatrix} \right\|^2_{\boldsymbol{Q}_r}$$

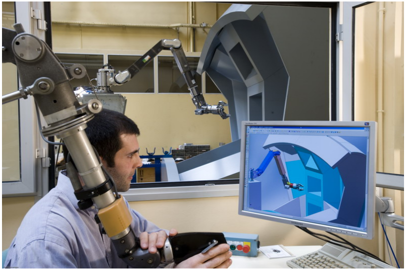
$$\begin{aligned} \text{such that} \quad \boldsymbol{M}(\boldsymbol{q}_k)\ddot{\boldsymbol{q}}_{k+1} + \boldsymbol{b}(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k) &= \boldsymbol{S}^T(\boldsymbol{q}_k)\boldsymbol{\tau}_{k+1} \\ \boldsymbol{\tau}_{min} \leq \boldsymbol{\tau}_{k+1} &\leq \boldsymbol{\tau}_{max} \\ \boldsymbol{q}_{min} \leq \boldsymbol{q}_{k+1} &\leq \boldsymbol{q}_{max} \\ \dot{\boldsymbol{q}}_{min} \leq \dot{\boldsymbol{q}}_{k+1} &\leq \dot{\boldsymbol{q}}_{max} \\ 0 \leq d^{rob,obj_j}_{k+1} & \quad \forall j \in \{1, ..., n_{obj}\} \end{aligned}$$

$$\boldsymbol{obj}\left(\ddot{\boldsymbol{q}}_{k+1}, \ddot{\boldsymbol{x}}^{*}_{k+1}\right) = \underbrace{\ddot{\boldsymbol{x}}^{des}_{k+1} + PD(\boldsymbol{x}_k, \boldsymbol{x}^{des}_{k+1})}_{\ddot{\boldsymbol{x}}^{*}_{k+1}} - \boldsymbol{J}(\boldsymbol{q}_k)\ddot{\boldsymbol{q}}_{k+1} - \dot{\boldsymbol{J}}(\boldsymbol{q}_k)\dot{\boldsymbol{q}}_k$$

Constraints compliance as a control feature : the teleoperation case

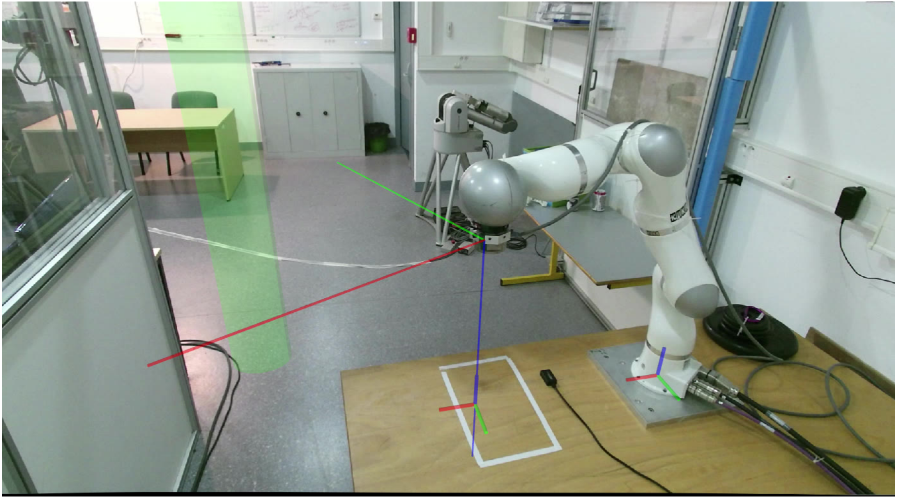
- PhD thesis Sébastien Rubrecht, ANR TELEMACH, CIFRE Bouygues Construction [Rubrecht 2010, Rubrecht 2011, Rubrecht 2012]
- Context : Teleoperation in tunnel boring machine cutter-heads
- Static environment, interactive task definition





Constraints compliance as a control feature

- PhD work of Lucas Joseph, CIFRE GE Healthcare [Joseph 2018c]
- Dynamic environment : perception in the loop and reactive constraints adaptation



Outline of the presentation

Where is redundancy hiding ?

- Classically, it's considered to be related to the null-space of the Jacobian
  $\dot{\boldsymbol{\nu}} = \boldsymbol{J}^{+}(\boldsymbol{q})\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{\nu}}_0$ or $\boldsymbol{\tau} = \boldsymbol{J}^{T}(\boldsymbol{q})\boldsymbol{f} + (\boldsymbol{I} - \boldsymbol{J}^{T}\boldsymbol{J}^{T+})\boldsymbol{\tau}_0$

Where is redundancy hiding ?

- Classically, it's considered to be related to the null-space of the Jacobian $\dot{\boldsymbol{\nu}} = \boldsymbol{J}^{+}(\boldsymbol{q})\boldsymbol{v} + (I - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{\nu}}_0$ or $\boldsymbol{\tau} = \boldsymbol{J}^{T}(\boldsymbol{q})\boldsymbol{f} + (I - \boldsymbol{J}^{T}\boldsymbol{J}^{T+})\boldsymbol{\tau}_0$

- In a QP, it does not appear explicitely. Three possibilities :

1. Write the cost function as a weighted sum of individual task constraints [Salini 2011],[Bouyarmane 2011]

$$\boldsymbol{\tau}^* = \arg\min_{\mathbb{X}} \quad T(\mathbb{X}) = \sum_{i=1}^{n_o} T_i(\mathbb{X}, \boldsymbol{W}_i) + w_0 T_0 \tag{1}$$

$$\text{subject to} \quad \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \boldsymbol{b}(\boldsymbol{q}, \boldsymbol{\nu}) = \boldsymbol{S}^{T}(\boldsymbol{q})\boldsymbol{\tau} + \sum_{i=1}^{n_c} \boldsymbol{J}_{c_i}^{\top}(\boldsymbol{q})\boldsymbol{f}_{c_i} \tag{2}$$

$$\boldsymbol{A}(\boldsymbol{q}, \boldsymbol{\nu})\mathbb{X} = \boldsymbol{b}(\boldsymbol{q}, \boldsymbol{\nu}) \tag{3}$$

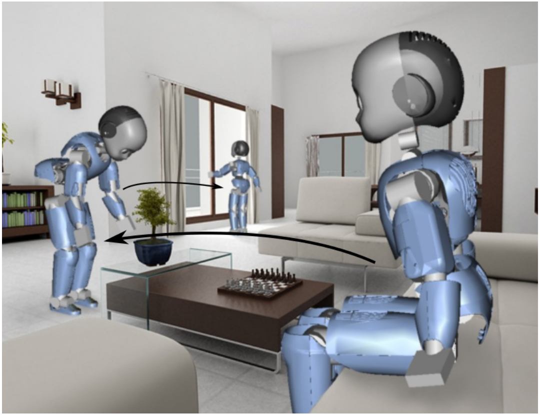
$$\boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\nu})\mathbb{X} \leq \boldsymbol{h}(\boldsymbol{q}, \boldsymbol{\nu}) \tag{4}$$

Where is redundancy hiding ?

- Classically, it's considered to be related to the null-space of the Jacobian $\dot{\boldsymbol{\nu}} = \boldsymbol{J}^{+}(\boldsymbol{q})\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{\nu}}_0$ or $\boldsymbol{\tau} = \boldsymbol{J}^{T}(\boldsymbol{q})\boldsymbol{f} + (\boldsymbol{I} - \boldsymbol{J}^{T}\boldsymbol{J}^{T+})\boldsymbol{\tau}_0$

- In a QP, it does not appear explicitely. Three possibilities :

1. Write the cost function as a weighted sum of individual task constraints [Salini 2011],[Bouyarmane 2011]



Where is redundancy hiding ?

- Classically, it's considered to be related to the null-space of the Jacobian $\dot{\boldsymbol{\nu}} = \boldsymbol{J}^{+}(\boldsymbol{q})\boldsymbol{v} + (I - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{\nu}}_0$ or $\boldsymbol{\tau} = \boldsymbol{J}^{T}(\boldsymbol{q})\boldsymbol{f} + (I - \boldsymbol{J}^{T}\boldsymbol{J}^{T+})\boldsymbol{\tau}_0$

- In a QP, it does not appear explicitely. Three possibilities :

1. Write the cost function as a weighted sum of individual task constraints [Salini 2011],[Bouyarmane 2011]

2. Solve a cascade of n_o QPs to ensure a strict hierarchy [Kanoun 2009], [Escande 2014]

$$\boldsymbol{\tau}^* = \underset{\mathbb{X}}{\arg\min} \quad T_i(\mathbb{X}) \tag{1}$$

$$\text{subject to} \quad \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \boldsymbol{b}(\boldsymbol{q},\boldsymbol{\nu}) = \boldsymbol{S}^{T}(\boldsymbol{q})\boldsymbol{\tau} + \sum_{i=1}^{n_c} \boldsymbol{J}_{c_i}^{\top}(\boldsymbol{q})\boldsymbol{f}_{c_i} \tag{2}$$

$$\boldsymbol{A}(\boldsymbol{q},\boldsymbol{\nu})\mathbb{X} = \boldsymbol{b}(\boldsymbol{q},\boldsymbol{\nu}) \tag{3}$$

$$\boldsymbol{D}(\boldsymbol{q},\boldsymbol{\nu})\mathbb{X} \leq \boldsymbol{h}(\boldsymbol{q},\boldsymbol{\nu}) \tag{4}$$

$$T_j(\mathbb{X}) = T_j^* \quad \forall j < i \tag{5}$$

Where is redundancy hiding ?

- Classically, it's considered to be related to the null-space of the Jacobian $\dot{\boldsymbol{\nu}} = \boldsymbol{J}^{+}(\boldsymbol{q})\boldsymbol{v} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\dot{\boldsymbol{\nu}}_0$ or $\boldsymbol{\tau} = \boldsymbol{J}^{T}(\boldsymbol{q})\boldsymbol{f} + (\boldsymbol{I} - \boldsymbol{J}^{T}\boldsymbol{J}^{T+})\boldsymbol{\tau}_0$

- In a QP, it does not appear explicitely. Three possibilities :

1. Write the cost function as a weighted sum of individual task constraints [Salini 2011],[Bouyarmane 2011]

2. Solve a cascade of n_o QPs to ensure a strict hierarchy [Kanoun 2009], [Escande 2014]

3. Solve a QP allowing the formulation and the smooth transition between both soft and strict hierarchy – Generalized Hierarchical Control [Liu 2016]

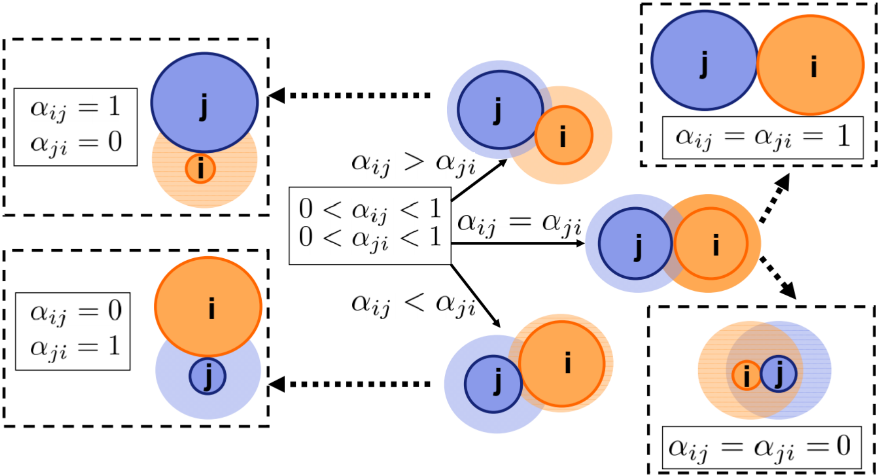
$$\boldsymbol{\tau}^* = \underset{\boldsymbol{\tau}, \boldsymbol{f}_c, \dot{\boldsymbol{\nu}}'}{\arg\min} \quad T(\mathbb{X}) = \sum_{i=1}^{n_o} T_i(\boldsymbol{\tau}, \boldsymbol{f}_c, \dot{\boldsymbol{\nu}}'_i) \tag{1}$$

$$\text{subject to} \quad \boldsymbol{M}(\boldsymbol{q})\boldsymbol{P}\dot{\boldsymbol{\nu}}' + \boldsymbol{b}(\boldsymbol{q}, \boldsymbol{\nu}) = \boldsymbol{S}^{T}(\boldsymbol{q})\boldsymbol{\tau} + \sum_{i=1}^{n_c} \boldsymbol{J}_{c_i}^{\top}(\boldsymbol{q})\boldsymbol{f}_{c_i} \tag{2}$$

$$\boldsymbol{A}(\boldsymbol{q}, \boldsymbol{\nu})[\boldsymbol{\tau}^{T}, \boldsymbol{f}_c^{T}, \boldsymbol{P}\dot{\boldsymbol{\nu}}'^{T}]^{T} = \boldsymbol{b}(\boldsymbol{q}, \boldsymbol{\nu}) \tag{3}$$

$$\boldsymbol{D}(\boldsymbol{q}, \boldsymbol{\nu})[\boldsymbol{\tau}^{T}, \boldsymbol{f}_c^{T}, \boldsymbol{P}\dot{\boldsymbol{\nu}}'^{T}]^{T} \leq \boldsymbol{h}(\boldsymbol{q}, \boldsymbol{\nu}) \tag{4}$$

Priorities in Generalized Hierarchical Control [Liu 2016]



- Task activation $\alpha_{ii} = 1 \longrightarrow \alpha_{ii} = 0$
- Task deactivation $\alpha_{ii} = 0 \longrightarrow \alpha_{ii} = 1$

Redundancy as a key to simple adaptive behaviours

- Redundancy also hides in the regularization task T_0

Redundancy as a key to simple adaptive behaviours

- Redundancy also hides in the regularization task T_0
- Often treated by default $\rightarrow$ converge towards a "good posture"

Redundancy as a key to simple adaptive behaviours

- Redundancy also hides in the regularization task T_0
- Often treated by default $\rightarrow$ converge towards a "good posture"
- "Good postures" can help convergence of NLP at the planning phase

Redundancy as a key to simple adaptive behaviours

- Redundancy also hides in the regularization task T_0
- Often treated by default $\rightarrow$ converge towards a "good posture"
- "Good postures" can help convergence of NLP at the planning phase

$\hookrightarrow$ But they mostly artificially constrain the solution space

Redundancy as a key to simple adaptive behaviours

- Redundancy also hides in the regularization task T_0
- Often treated by default $\rightarrow$ converge towards a "good posture"
- "Good postures" can help convergence of NLP at the planning phase

$\hookrightarrow$ But they mostly artificially constrain the solution space

- There are some alternatives : gravity compensation, viscous friction, middle of the constraints,...

Redundancy as a key to simple adaptive behaviours

- Redundancy also hides in the regularization task T_0
- Often treated by default $\rightarrow$ converge towards a "good posture"
- "Good postures" can help convergence of NLP at the planning phase

$\hookrightarrow$ But they mostly artificially constrain the solution space

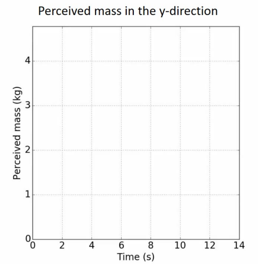
- There are some alternatives : gravity compensation, viscous friction, middle of the constraints,...

Redundancy as a key to simple adaptive behaviours

- Redundancy also hides in the regularization task T_0
- Often treated by default $\rightarrow$ converge towards a "good posture"
- "Good postures" can help convergence of NLP at the planning phase

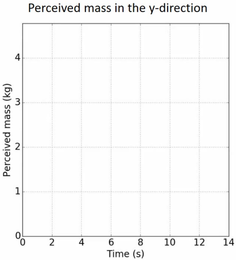
$\hookrightarrow$ But they mostly artificially constrain the solution space

- There are some alternatives : gravity compensation, viscous friction, middle of the constraints,...



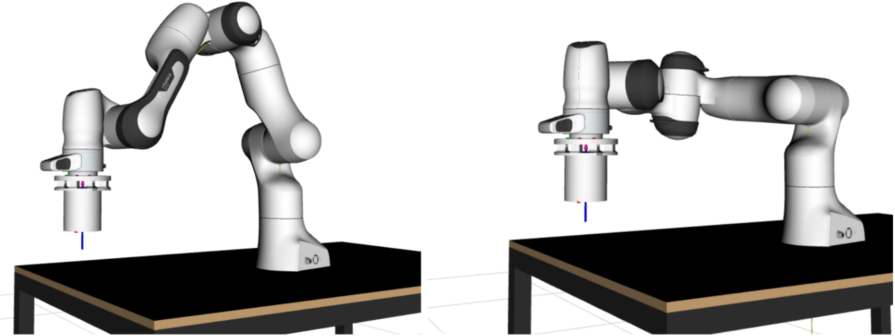
Redundancy as a key to simple adaptive behaviours

- Apparent mass minimization in the potential direction of interaction [Joseph 2018a]



Redundancy as a key to simple adaptive behaviours

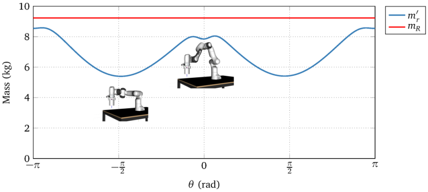
- Apparent mass minimization in the potential direction of interaction [Joseph 2018a]
- Makes a significative difference at impact time (H2020 COVR HARRY2 project)

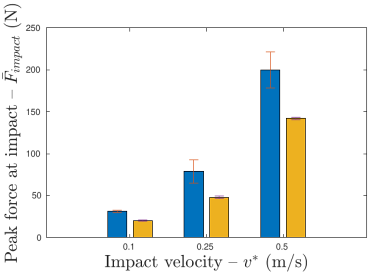


(a) Configuration $\boldsymbol{q}_1$ (b) Configuration $\boldsymbol{q}_2$

Redundancy as a key to simple adaptive behaviours

- Apparent mass minimization in the potential direction of interaction [Joseph 2018a]
- Makes a significative difference at impact time (H2020 COVR HARRY2 project)





(b) Comparison of the averaged maximum peak force at impact time as a function of impact velocity and in two different configurations $\boldsymbol{q}_1$ (blue) and $\boldsymbol{q}_2$ (yellow). Standard deviation is plotted as a red whisker.

Outline of the presentation

Energetic approach to safety

Important observations

- Fixed-based robot can't escape and Human motion and intention is hard to predict

$\hookrightarrow$ Collisions will occur

Energetic approach to safety

Important observations

- Fixed-based robot can't escape and Human motion and intention is hard to predict

$\hookrightarrow$ Collisions will occur

- Dissipated **Kinetic Energy** at **impact** $\rightarrow$ source of **danger** :

$$\int_u F_{impact}\,du = E_{dissipated} = E_c^{hum} + E_c^{rob}$$

- Robot Kinetic Energy (expressed at the end-effector) :

$$E_{c,k} = \frac{1}{2}\dot{\boldsymbol{x}}_k^T \boldsymbol{\Lambda}(\boldsymbol{q}_k)\dot{\boldsymbol{x}}_k \qquad \text{with} \qquad \boldsymbol{\Lambda}(\boldsymbol{q}) = (\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}))^{-1}$$

Energetic approach to safety

Important observations

- Fixed-based robot can't escape and Human motion and intention is hard to predict

$\hookrightarrow$ Collisions will occur

- Dissipated **Kinetic Energy** at **impact** $\rightarrow$ source of **danger** :

$$\int_u F_{impact}\,du = E_{dissipated}$$
$$= E_c^{hum} + E_c^{rob}$$

- Robot Kinetic Energy (expressed at the end-effector) :

$$E_{c,k} = \frac{1}{2}\dot{\boldsymbol{x}}_k^T \boldsymbol{\Lambda}(\boldsymbol{q}_k)\dot{\boldsymbol{x}}_k \qquad \text{with} \qquad \boldsymbol{\Lambda}(\boldsymbol{q}) = \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\right)^{-1}$$

- Future Kinetic Energy : $E_{c,k+1} = E_{c,k} + \Delta E_c$

$$\Delta E_c = \underbrace{\left(\dot{\boldsymbol{x}}_k \Delta t + 0.5\ddot{\boldsymbol{x}}_{k+1}^c (\Delta t)^2\right)^T}_{\text{Expected task motion}} \underbrace{\boldsymbol{\Lambda}(\boldsymbol{q}_k)\ddot{\boldsymbol{x}}_{k+1}}_{\text{Equivalent actuation force}}$$

Energetic approach to safety

Important observations

- Fixed-based robot can't escape and Human motion and intention is hard to predict

$\hookrightarrow$ Collisions will occur

- Dissipated **Kinetic Energy** at **impact** $\rightarrow$ source of **danger** :

$$\int_u F_{impact}\, du = E_{dissipated}$$
$$= E_c^{hum} + E_c^{rob}$$

- Robot Kinetic Energy (expressed at the end-effector) :

$$E_{c,k} = \frac{1}{2}\dot{\boldsymbol{x}}_k^T \boldsymbol{\Lambda}(\boldsymbol{q}_k)\dot{\boldsymbol{x}}_k \qquad \text{with} \qquad \boldsymbol{\Lambda}(\boldsymbol{q}) = (\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}))^{-1}$$

- Future Kinetic Energy : $E_{c,k+1} = E_{c,k} + \Delta E_c$

$$\Delta E_c = \underbrace{\Delta \boldsymbol{x}_{k+1}^T}_{\text{Expected task motion}} \underbrace{\boldsymbol{\Lambda}(\boldsymbol{q}_k)\boldsymbol{J}(\boldsymbol{q}_k)\boldsymbol{M}^{-1}(\boldsymbol{q}_k)(\boldsymbol{S}^T(\boldsymbol{q}_k)\boldsymbol{\tau}_{k+1} - \boldsymbol{b}(\boldsymbol{q}_k, \boldsymbol{\nu}_k)) + \dot{\boldsymbol{J}}(\boldsymbol{q}_k)\boldsymbol{\nu}_k}_{\text{Equivalent actuation force}}$$

Energetic approach to safety

Important observations

- Fixed-based robot can't escape and Human motion and intention is hard to predict

$\hookrightarrow$ Collisions will occur

- Dissipated **Kinetic Energy** at **impact** $\rightarrow$ source of **danger** :

$$\int_u F_{impact}\, du = E_{dissipated}$$
$$= E_c^{hum} + E_c^{rob}$$

- Robot Kinetic Energy (expressed at the end-effector) :

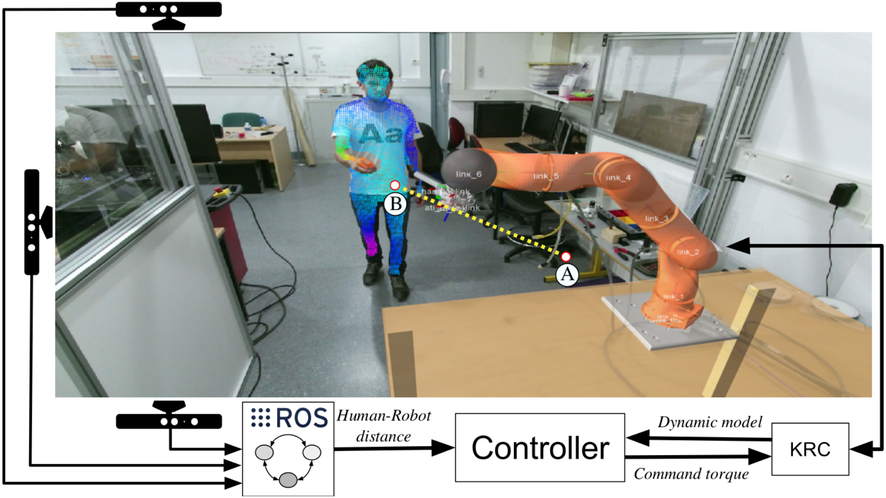
$$E_{c,k} = \frac{1}{2}\dot{\boldsymbol{x}}_k^T \boldsymbol{\Lambda}(\boldsymbol{q}_k)\dot{\boldsymbol{x}}_k \quad \text{with} \quad \boldsymbol{\Lambda}(\boldsymbol{q}) = (\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q}))^{-1}$$

- Future Kinetic Energy : $E_{c,k+1} = E_{c,k} + \Delta E_c$

$$\Delta E_c = \underbrace{\Delta \boldsymbol{x}_{k+1}^T}_{\text{Expected task motion}} \underbrace{\boldsymbol{\Lambda}(\boldsymbol{q}_k)\boldsymbol{J}(\boldsymbol{q}_k)\boldsymbol{M}^{-1}(\boldsymbol{q}_k)(\boldsymbol{S}^T(\boldsymbol{q}_k)\boldsymbol{\tau}_{k+1} - \boldsymbol{b}(\boldsymbol{q}_k, \boldsymbol{\nu}_k)) + \dot{\boldsymbol{J}}(\boldsymbol{q}_k)\boldsymbol{\nu}_k}_{\text{Equivalent actuation force}}$$

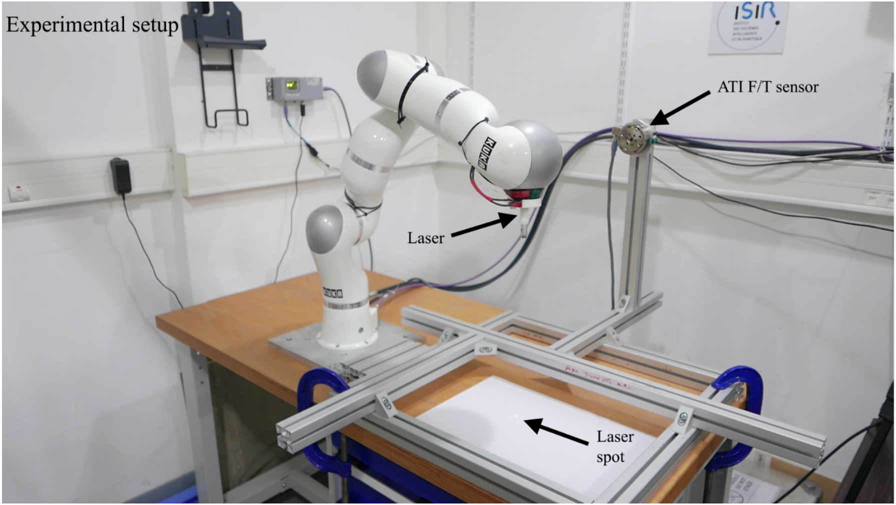
- We can write a constraint on Kinetic energy at each time [ISO 2016]

Energetic approach to safety



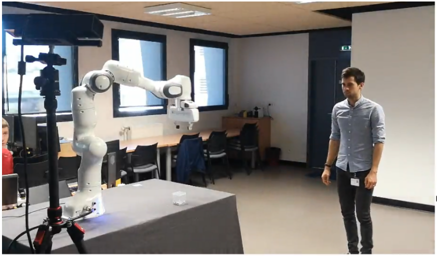
. [Meguenani 2017a],[Joseph 2018b]

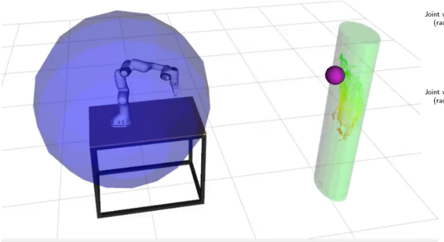
Energetic approach to safety

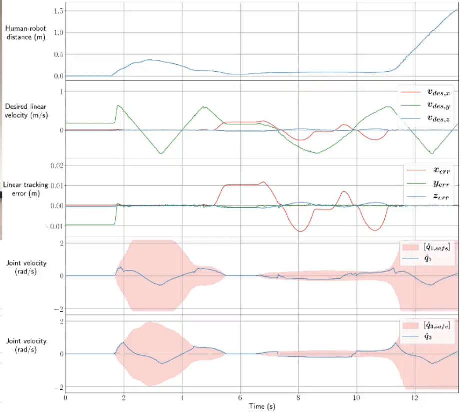


. [Meguenani 2017a],[Joseph 2018b]

Energetic approach to safety [Joseph 2020]





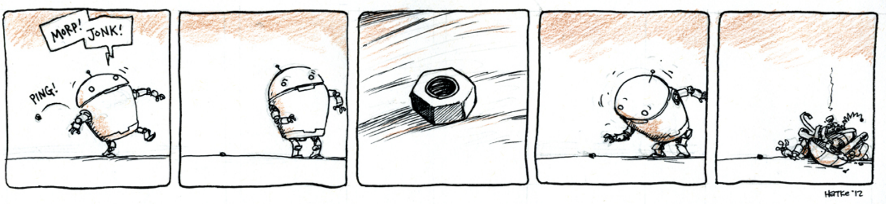


The robot safely continues operating when in proximity to the human.

Linear tracking error is minimized by modulating the desired linear velocity of the trajectory.

This allows to maximize the robot's performance without jeopardizing the human's safety

– To be continued –



References

Karim Bouyarmane and Abderrahmane Kheddar.
Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations.
In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4414–4419. IEEE, 2011.

S. Chiaverini.
Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators.
Robotics and Automation, IEEE Transactions on, vol. 13, no. 3, pages 398–410, 1997.

Andrea Del Prete.
Joint Position and Velocity Bounds in Discrete-Time Acceleration/Torque Control of Robot Manipulators.
IEEE Robotics and Automation Letters, vol. 3, no. 1, pages pp.281–288, January 2018.

Adrien Escande, Nicolas Mansard and Pierre-Brice Wieber.
Hierarchical quadratic programming : Fast online humanoid-robot motion generation.
The International Journal of Robotics Research, 2014.

F. Flacco, A. De Luca and O. Khatib.
Prioritized multi-task motion control of redundant robots under hard joint constraints.
In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 3970–3977, 2012.

A. Ibanez, P. Bidaud and V. Padois.
Optimization-based control approaches to humanoid balancing.
In Ambarish Goswami and Prahlad Vadakkepat, editeurs, Humanoid Robotics : a Reference, pages 1–27. Springer Netherlands, 2017.

ISO.
ISO TS-15066, Robots and robotic devices - Collaborative robots, 2016.

L. Joseph, V. Padois and G. Morel.
Experimental validation of an energy constraint for a safer collaboration with robots.
In Proceedings of the International Symposium on Experimental Robotics, Buenos Aires, Argentina, November 2018.

L. Joseph, V. Padois and G. Morel.
Towards X-ray medical imaging with robots in the open : safety without compromising performances.
In Proceedings of the IEEE International Conference on Robotics and Automation, pages 6604–6610, Brisbane, Australia, May 2018.

Lucas Joseph.
An energetic approach to safety in robotic manipulation.
Theses, Sorbonne Université, December 2018.

Lucas Joseph, Joshua Pickard, Vincent Padois and David Daney.
Online velocity constraint adaptation for safe and efficient human-robot workspace sharing.
In In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, USA, October 2020.

Oussama Kanoun, Florent Lamiraux, Pierre-Brice Wieber, Fumio Kanehiro, Eiichi Yoshida and Jean-Paul Laumond.
Prioritizing linear equality and inequality systems : application to local motion planning for redundant robots.
In Proceedings of the IEEE International Conference on Robotics and Automation, pages 2939–2944, 2009.

O. Khatib.
A unified approach for motion and force control of robot manipulators : The operational space formulation.
IEEE Journal of Robotics and Automation, vol. 3, no. 1, pages 43–53, 1987.

A. Liégeois.
Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms.
IEEE Transactions on Systems, Man and Cybernetics, vol. 7, no. 12, pages 868–871, dec. 1977.

M. Liu, Y. Tan and V. Padois.
Generalized hierarchical control.
Autonomous Robots, vol. 40, no. 1, pages 17–31, 2016.

N. Mansard, O. Khatib and A. Kheddar.
A Unified Approach to Integrate Unilateral Constraints in the Stack of Tasks.
IEEE Transactions on Robotics, vol. 25, no. 3, pages 670–685, june 2009.

A. Meguenani, V. Padois, J. Da Silva, A. Hoarau and P. Bidaud.
Energy-based control for safe Human-robot physical interactions.
In D. Kulic, G. Venture, Y. Nakamura and O. Khatib, editeurs, Springer Proceedings in Advanced Robotics - The 2016 International Symposium on Experimental Robotics. Springer International Publishing AG, 2017.

Anis Meguenani.
Safe control of robotic manipulators in dynamic contexts.
Theses, Université Pierre et Marie Curie - Paris VI, October 2017.

S. Rubrecht, V. Padois, P. Bidaud and M. de Broissia.
Constraint Compliant Control for a Redundant Manipulator in a Cluttered Environment.
In Jadran Lenarčič and Michael M. Stanisic, editeurs, Advances in Robot Kinematics : Motion in Man and Machine, pages 367–376. Springer, 2010.

S. Rubrecht.
Contributions to the control of constrained robot
PhD thesis, Université Pierre et Marie Curie - Paris VI, September 2011.

S. Rubrecht, V. Padois, P. Bidaud, M. de Broissia and M. Da Silva Simoes.
Motion safety and constraints compatibility for multibody robots.
Autonomous Robots, vol. 32, no. 3, pages 333–349, 2012.

J. Salini, V. Padois and P. Bidaud.
Synthesis of Complex Humanoid Whole-Body Behavior : a Focus on Sequencing and Tasks Transitions.
In Proceedings of the IEEE International Conference on Robotics and Automation, pages 1283–1290, Shanghai, China, May 2011.

B. Siciliano and J.-J.E. Slotine.
A general framework for managing multiple tasks in highly redundant robotic systems.
In Fifth International Conference on Advanced Robotics, pages 1211 –1216 vol.2, june 1991.