# Hybrid Force Position Control of a Kuka LWR4

Corso di LM in Ingegneria Robotica ed Automazione
Controllo dei Robot


Students:
Nicola Piga
Giulio Romualdi

Supervisors:
Prof. Antonio Bicchi
Ing. Manuel Bonilla


17/05/2017

Università di Pisa

# Project description

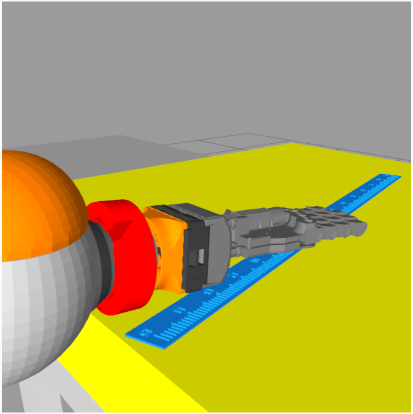
## Grasp of thin objects

**Problem**

Performing the grasp of thin objects could be a hard task because not enough contact constraints are provided by the object

# A solution to the problem

Use of environmental constraints to arrange an easier grasp and then perform the grasp



**Example**

1. the hand is placed on the object
2. the hand drags the object until it sticks out of the border of the table
3. the hand grasps the object

# How to perform a safe dragging phase

The dragging phase could damage the object due to uncontrolled contact forces between the hand and the object.

An hybrid position force control strategy allows to avoid this issue.

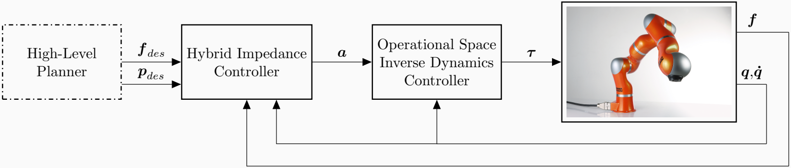
**Aim of this project**

Develop an hybrid force position control that allows to move the hand and at the same time to regulate the contact force using a force feedback signal.

# Control architecture

The hybrid force position control was obtained using the Hybrid Impedance Approach (HIC) proposed by Anderson and Spong in the 1988.

This approach requires the existence of an inverse dynamics control inner loop to cancel the nonlinearities of the manipulator




The inner loop applied to the manipulator results in a double integrator system $\ddot{\boldsymbol{x}} = \boldsymbol{a}$ where $\boldsymbol{x}$ is a Cartesian description of the end-effector and $\boldsymbol{a}$ is chosen using the HIC approach.

# Contents

The contents presented are

- the HIC approach
- the development of a HIC based outer control loop
- the inverse dynamics inner loop
- how to correctly use a force torque sensor that is required to close the outer loop
- the results obtained in simulation and in a real scenario
- notes about the software implementation
- future works

# Hybrid impedance control

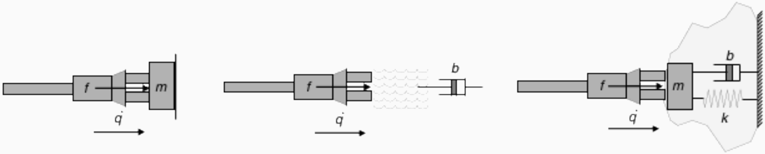
# Hybrid impedance control [Anderson, Spong, 1988]

The Hybrid Impedance Control approach features a more general concept of impedance than that used in typical PD position controllers with tuned apparent impedances of the form

$$(\ddot{\boldsymbol{x}}^{des} - \ddot{\boldsymbol{x}}) + K_{damping}(\dot{\boldsymbol{x}}^{des} - \dot{\boldsymbol{x}}) + K_{stifness}(\boldsymbol{x}^{des} - \boldsymbol{x}) = \boldsymbol{F}$$

It allows to synthesize for each DoF of the manipualator both a position or a direct force controller by matching a given "environment impedance" with the appropriate "manipulator impedance"

# HIC - Type of impedances




- for each cartesian DoF the manipulator and the environment can be described using impedances $Z_m$ and $Z_e$
- the environment is defined to be any element connected to or contacting the robot anywhere past the wrist force sensor
- $Z(\omega) = R(\omega) + jX(\omega)$
- type of impedances

  inertial iff $|Z(0)| = 0$

  resistive iff $|Z(0)| = c \in (0, \infty)$

  capacitive iff $|Z(0)| \to \infty$

# HIC - Duality principle and equivalence to circuit theory

**Duality principle**

The manipulator should be controlled to respond as the dual of the environment

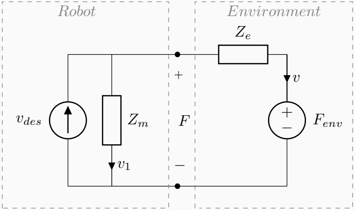
This principle is most easily described in terms of Norton and Thèvenin equivalents

- an inertial environment is represented using a Thèvenin equivalent
- a capacitive environment is represented using a Norton equivalent
- a resistive environment is represented using either a Thèvenin or a Norton equivalent

# HIC - Duality principle for position control

Manipulator impedance chosen as the dual of the environment impedance in order to obtain zero steady state error to a step input

$$v = \frac{Z_m(s)}{Z_m(s) + Z_e(s)} v_{des} - \frac{F_{env}}{Z_e + Z_m}$$




$$e_{ss}\Big|_{F_{env} \equiv 0} = \lim_{s \to 0} (v - v_{des}) = \frac{-Z_e(0)}{Z_m(0) + Z_e(0)} = 0$$

as long as $Z_m(0) \neq 0$ and $Z_e(0) = 0$

**Rule of thumb**

inertial environments are position controlled with a noninertial manipulator impedance (the actual value of $Z_e(0)$ should not be known)

# HIC - Position controlled subsystem

The electrical circuit can be seen as a control feedback scheme where $a$ is the outer loop acceleration corresponding to the desired velocity $v_{des}$

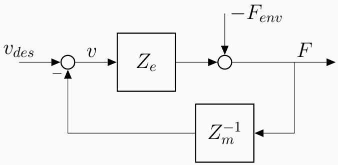
$$v_{des} = v + v_1$$

$$v_1 = \frac{F}{Z_m}$$

$$F = Z_e v + F_{env}$$

$$a = \dot{v} = \frac{\mathrm{d}}{\mathrm{d}t}\left(v_{des} - \frac{F}{Z_m}\right)$$

$$Z_m = Ms + \tilde{Z}_m$$




The acceleration can be written without derivatives

$$a = \frac{\mathrm{d}}{\mathrm{d}t}\left(v_{des} - \frac{F}{Ms + \tilde{Z}_m}\right) = \dot{v}_{des} - \frac{Fs}{Ms + \tilde{Z}_m} = \dot{v}_{des} - sv_1$$

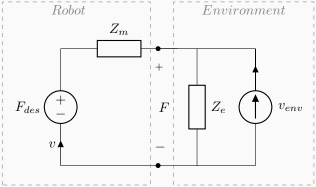
$$F = v_1(Ms + \tilde{Z}_m) \quad v_1 = \frac{F - (v_{des} - v)\tilde{Z}_m}{Ms}$$

$$a = \dot{v}_{des} - sv_1 = \dot{v}_{des} - \cancel{s}\left(\frac{F - (v_{des} - v)\tilde{Z}_m}{M\cancel{s}}\right) = \dot{v}_{des} + \frac{(v_{des} - v)\tilde{Z}_m}{M} - \frac{F}{M}$$

# HIC - Duality principle for force control

Manipulator impedance chosen as the dual of the environment impedance in order to obtain zero steady state error to a step input

$$F = \frac{Z_e(s)}{Z_m(s) + Z_e(s)} F_{des} + \frac{Z_e Z_m}{Z_m + Z_e} V_{env}$$




$$e_{ss}\Big|_{v_{env}\equiv 0} = \lim_{s\to 0}(F - F_{des}) = \frac{-Z_m(0)}{Z_m(0) + Z_e(0)} = 0$$

as long as $Z_m(0) < \infty$ and $Z_e(0) \to \infty$

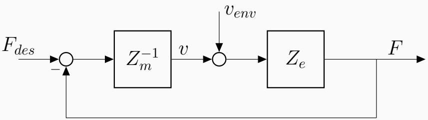
**Rule of thumb**

capacitive environments are force controlled with a noncapacitive manipulator impedance (the actual value of $Z_e(0)$ should not be known)

# HIC - Force controlled subsystem

The electrical circuit can be seen as a control feedback scheme where $a$ is the outer loop acceleration of the DoF corresponding to the desired force $F_{des}$

$$
\begin{aligned}
F &= F_{des} + Z_m v \\
v &= \frac{F - F_{des}}{Z_m} \\
F &= Z_e(v + v_{env}) \\
a = \dot{v} &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{F - F_{des}}{Z_m}\right) \\
Z_m &= Ms + \tilde{Z}_m
\end{aligned}
$$



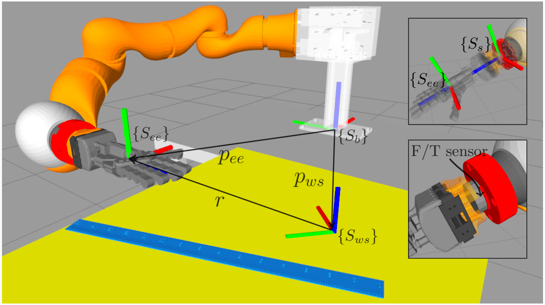

The acceleration can be written without derivatives

$$
\begin{aligned}
a &= \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{F - F_{des}}{Ms + \tilde{Z}_m}\right) = \left(\frac{s(F - F_{des})}{Ms + \tilde{Z}_m}\right) \\
vMs + v\tilde{Z}_m &= F - F_{des} \\
v &= \frac{1}{Ms}(F - F_{des} - v\tilde{Z}_m) \\
a = \dot{v} &= \frac{\cancel{s}}{M\cancel{s}}(F - F_{des}) - \frac{\cancel{s}}{M\cancel{s}}(\tilde{Z}_m v) = \frac{1}{M}(F - F_{des}) - \frac{1}{M}(\tilde{Z}_m v)
\end{aligned}
$$

# HIC based control architecture

# References frames

Before delving into an HIC based controller let us introduce some useful reference frames and notation




$b$: base

$s$: sensor

$ee$: end-effector (hand)

$ws$: work-space (table)

$$\{S_b\} = \{B; x_b, y_b, z_b\} \qquad \{S_{ee}\} = \{E; x_{ee}, y_{ee}, z_{ee}\}$$

$$\{S_s\} = \{S; x_s, y_s, z_s\} \qquad \{S_{ws}\} = \{W; x_{ws}, y_{ws}, z_{ws}\}$$

# Notation

### Space state vector definition

A natural choice for a basis in which the commanded positions and forces are expressed is *ws*

$$^{ws}\boldsymbol{x} = \begin{bmatrix} ^{ws}r_x & ^{ws}r_y & ^{ws}r_z & \psi & \theta & \phi \end{bmatrix}^T$$

$$^{ws}R_{ee} = R_{ZYZ}(\psi, \theta, \phi) = R_{ZYZ}(\boldsymbol{\Phi})$$

### Convention

Convention used for any quantity $X$ encountered, $^{b}X_{p}$

- $^{b}.$ reference frame
- $._{p}$ reference point (for wrench and Jacobian only)

## Control aims

Suppose that the second derivative of the state can be chosen arbitrarily

$$^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}_{cmd}$$

Find $\boldsymbol{a}_{cmd}$ using HIC such that

- $r_x$, $r_y$ and $\boldsymbol{\Phi}$ are rigidly controlled i.e.
  i. $\ddot{e}_x + B_x\dot{e}_x + K_x e_x = 0 \qquad e_x(t) = {}^{ws}r_{x,des}(t) - {}^{ws}r_x(t)$
  ii. $\ddot{e}_y + B_y\dot{e}_y + K_y e_y = 0 \qquad e_y(t) = {}^{ws}r_{y,des}(t) - {}^{ws}r_y(t)$
  iii. $\ddot{\boldsymbol{e}}_\Phi + B_\Phi\dot{\boldsymbol{e}}_\Phi + K_\Phi \boldsymbol{e}_\Phi = \boldsymbol{0} \qquad \boldsymbol{e}_\Phi(t) = \boldsymbol{\Phi}_{des}(t) - \boldsymbol{\Phi}(t)$
- the DoF along $z_{ws}$ is force controlled i.e.
  i. $e_z \xrightarrow[t\to\infty]{} 0 \qquad e_z(t) = {}^{ws}F_{z,des}(t) - {}^{ws}F_z(t)$

where ${}^{ws}F_z(t)$ is the force exerted by the hand to the object expressed in $ws$

# Resulting HIC based controller

**Position ($^{ws}r_x$, $^{ws}r_y$)**

- inertial environment supposed, i.e., manipulator moving a payload along given axis
- $Z_{m,p} = M_p s + \tilde{Z}_{m,p} = M_p s + B_p + \frac{K_p}{s} = s + B_p + \frac{K_p}{s}$

$$^{ws}a_{cmd,x} = \ddot{r}_{x,des} + B_x(\dot{r}_{x,des} - \dot{r}_x) + K_x(r_{x,des} - r_x) - F_x$$
$$^{ws}a_{cmd,y} = \ddot{r}_{y,des} + B_y(\dot{r}_{y,des} - \dot{r}_y) + K_y(r_{y,des} - r_y) - F_y$$

**Attitude ($\psi$, $\theta$, $\phi$)**

- inertial environment supposed, i.e., manipulator rotating a payload about given axis
- $Z_{m,a} = M_a s + \tilde{Z}_{m,a} = M_a s + B_a + \frac{K_a}{s} = s + B_a + \frac{K_a}{s}$

$$^{ws}\boldsymbol{a}_{cmd,\Phi} = \ddot{\boldsymbol{\Phi}}_{des} + B_\Phi(\dot{\boldsymbol{\Phi}}_{des} - \dot{\boldsymbol{\Phi}}) + K_\Phi(\boldsymbol{\Phi}_{des} - \boldsymbol{\Phi})$$

**Force ($^{ws}F_z$)**

- capacitive environment supposed
- $Z_{m,f} = M_f s + \tilde{Z}_{m,f} = M_f s + \tilde{B}_f = \frac{s}{K_f} + \frac{B_f}{K_f}$

$$^{ws}a_{cmd,z} = -B_f \dot{r}_z + K_f(F_{z,des} - F_z)$$

# Inner loop inverse dynamics

# Inverse dynamics controller

In order to assign the second derivative of the state $\boldsymbol{x}$ arbitrarily an inverse dynamics controller is required

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - {}^{b}J_S^{T\,b}\boldsymbol{w}_S \longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

# Inverse dynamics controller

In order to assign the second derivative of the state $\boldsymbol{x}$ arbitrarily an inverse dynamics controller is required

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - {}^{b}J_S^{T\,b}\boldsymbol{w}_S \longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

Joint Space description

$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^{b}J_S^{T}({}^{b}\gamma + {}^{b}\boldsymbol{w}_S)$

# Inverse dynamics controller

In order to assign the second derivative of the state $\boldsymbol{x}$ arbitrarily an inverse dynamics controller is required

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - {}^{b}J_S^{T}\,{}^{b}\boldsymbol{w}_S \longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

Joint Space description

$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^{b}J_S^{T}({}^{b}\gamma + {}^{b}\boldsymbol{w}_S)$

$\ddot{\boldsymbol{q}} = B^{-1}({}^{b}J_S^{T})^{b}\gamma$

# Inverse dynamics controller

In order to assign the second derivative of the state $\boldsymbol{x}$ arbitrarily an inverse dynamics controller is required

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - {}^{b}J_S^{T}\,{}^{b}\boldsymbol{w}_S \longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

Joint Space description

$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^{b}J_S^{T}({}^{b}\gamma + {}^{b}\boldsymbol{w}_S)$

$\ddot{\boldsymbol{q}} = B^{-1}({}^{b}J_S^{T})^{b}\gamma$

HIC requires operational space description

${}^{ws}\ddot{\boldsymbol{x}} = {}^{ws}J_{A,E}\ddot{\boldsymbol{q}} + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$

# Inverse dynamics controller

In order to assign the second derivative of the state $\boldsymbol{x}$ arbitrarily an inverse dynamics controller is required

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - {}^{b}J_S^{T}\,{}^{b}\boldsymbol{w}_S \longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

Joint Space description

$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^{b}J_S^{T}({}^{b}\gamma + {}^{b}\boldsymbol{w}_S)$

$\ddot{\boldsymbol{q}} = B^{-1}({}^{b}J_S^{T})^{b}\gamma$

HIC requires operational space description

${}^{ws}\ddot{\boldsymbol{x}} = {}^{ws}J_{A,E}\ddot{\boldsymbol{q}} + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$

${}^{ws}J_{A,E} = \begin{bmatrix} I & 0 \\ 0 & T^{-1}(\boldsymbol{\Phi}) \end{bmatrix} {}^{ws}J_E \quad {}^{ws}\boldsymbol{\omega} = T(\boldsymbol{\Phi})\dot{\boldsymbol{\Phi}}$

$\det(T) = -\sin(\theta) \neq 0$

${}^{ws}\dot{J}_{A,E} = \begin{bmatrix} 0 & 0 \\ 0 & -T^{-1}\dot{T}T^{-1} \end{bmatrix} {}^{ws}J_E + \begin{bmatrix} I & 0 \\ 0 & T^{-1} \end{bmatrix} {}^{ws}\dot{J}_E$

# Inverse dynamics controller

In order to assign the second derivative of the state $\boldsymbol{x}$ arbitrarily an inverse dynamics controller is required

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - {}^{b}J_S^{T}\,{}^{b}\boldsymbol{w}_S \longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

Joint Space description

$$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^{b}J_S^{T}({}^{b}\gamma + {}^{b}\boldsymbol{w}_S)$$
$$\ddot{\boldsymbol{q}} = B^{-1}({}^{b}J_S^{T})^{b}\gamma$$

HIC requires operational space description

$${}^{ws}\ddot{\boldsymbol{x}} = {}^{ws}J_{A,E}\ddot{\boldsymbol{q}} + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$
$${}^{ws}J_{A,E} = \begin{bmatrix} I & 0 \\ 0 & T^{-1}(\boldsymbol{\Phi}) \end{bmatrix} {}^{ws}J_E \quad {}^{ws}\boldsymbol{\omega} = T(\boldsymbol{\Phi})\dot{\boldsymbol{\Phi}}$$
$$\det(T) = -\sin(\theta) \neq 0$$
$${}^{ws}\dot{J}_{A,E} = \begin{bmatrix} 0 & 0 \\ 0 & -T^{-1}\dot{T}T^{-1} \end{bmatrix} {}^{ws}J_E + \begin{bmatrix} I & 0 \\ 0 & T^{-1} \end{bmatrix} {}^{ws}\dot{J}_E$$

Substitute $\ddot{\boldsymbol{q}}$ in the operational space description

$${}^{ws}\ddot{\boldsymbol{x}} = \underbrace{{}^{ws}J_{A,E}B^{-1}({}^{b}J_S^{T})}_{\Lambda_A^{-1}}{}^{b}\gamma + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$

# Inverse dynamics controller

In order to assign the second derivative of the state $\boldsymbol{x}$ arbitrarily an inverse dynamics controller is required

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - {}^{b}J_S^{T}\,{}^{b}\boldsymbol{w}_S \longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

Joint Space description

$$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^{b}J_S^{T}({}^{b}\gamma + {}^{b}\boldsymbol{w}_S)$$
$$\ddot{\boldsymbol{q}} = B^{-1}({}^{b}J_S^{T})^{b}\gamma$$

HIC requires operational space description

$${}^{ws}\ddot{\boldsymbol{x}} = {}^{ws}J_{A,E}\ddot{\boldsymbol{q}} + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$
$${}^{ws}J_{A,E} = \begin{bmatrix} I & 0 \\ 0 & T^{-1}(\boldsymbol{\Phi}) \end{bmatrix} {}^{ws}J_E \quad {}^{ws}\boldsymbol{\omega} = T(\boldsymbol{\Phi})\dot{\boldsymbol{\Phi}}$$
$$\det(T) = -\sin(\theta) \neq 0$$
$${}^{ws}\dot{J}_{A,E} = \begin{bmatrix} 0 & 0 \\ 0 & -T^{-1}\dot{T}T^{-1} \end{bmatrix} {}^{ws}J_E + \begin{bmatrix} I & 0 \\ 0 & T^{-1} \end{bmatrix} {}^{ws}\dot{J}_E$$

Substitute $\ddot{\boldsymbol{q}}$ in the operational space description

$${}^{ws}\ddot{\boldsymbol{x}} = \underbrace{{}^{ws}J_{A,E}B^{-1}({}^{b}J_S^{T})}_{\Lambda_A^{-1}}{}^{b}\gamma + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$

$$\Lambda_A{}^{ws}\ddot{\boldsymbol{x}} = \gamma + \Lambda_A{}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$

# Inverse dynamics controller

In order to assign the second derivative of the state $\boldsymbol{x}$ arbitrarily an inverse dynamics controller is required

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - {}^{b}J_S^{T\,b}\boldsymbol{w}_S \longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

Joint Space description

$$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^{b}J_S^T({}^{b}\gamma + {}^{b}\boldsymbol{w}_S)$$

$$\ddot{\boldsymbol{q}} = B^{-1}({}^{b}J_S^T)^{b}\gamma$$

HIC requires operational space description

$${}^{ws}\ddot{\boldsymbol{x}} = {}^{ws}J_{A,E}\ddot{\boldsymbol{q}} + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$

$${}^{ws}J_{A,E} = \begin{bmatrix} I & 0 \\ 0 & T^{-1}(\boldsymbol{\Phi}) \end{bmatrix} {}^{ws}J_E \quad {}^{ws}\boldsymbol{\omega} = T(\boldsymbol{\Phi})\dot{\boldsymbol{\Phi}}$$

$$\det(T) = -\sin(\theta) \neq 0$$

$${}^{ws}\dot{J}_{A,E} = \begin{bmatrix} 0 & 0 \\ 0 & -T^{-1}\dot{T}T^{-1} \end{bmatrix} {}^{ws}J_E + \begin{bmatrix} I & 0 \\ 0 & T^{-1} \end{bmatrix} {}^{ws}\dot{J}_E$$

Substitute $\ddot{\boldsymbol{q}}$ in the operational space description

$${}^{ws}\ddot{\boldsymbol{x}} = \underbrace{{}^{ws}J_{A,E}B^{-1}({}^{b}J_S^T)}_{\Lambda_A^{-1}}{}^{b}\gamma + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$

$$\Lambda_A{}^{ws}\ddot{\boldsymbol{x}} = \gamma + \Lambda_A{}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$

$$\gamma_{cmd} = \Lambda_A\boldsymbol{a} - \Lambda_A{}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$

# Inverse dynamics controller

In order to assign the second derivative of the state $\boldsymbol{x}$ arbitrarily an inverse dynamics controller is required

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} - {}^{b}J_S^{T\,b}\boldsymbol{w}_S \longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

Joint Space description

$$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^{b}J_S^{T}({}^{b}\gamma + {}^{b}\boldsymbol{w}_S)$$
$$\ddot{\boldsymbol{q}} = B^{-1}({}^{b}J_S^{T})^{b}\gamma$$

HIC requires operational space description

$${}^{ws}\ddot{\boldsymbol{x}} = {}^{ws}J_{A,E}\ddot{\boldsymbol{q}} + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$
$${}^{ws}J_{A,E} = \begin{bmatrix} I & 0 \\ 0 & T^{-1}(\boldsymbol{\Phi}) \end{bmatrix} {}^{ws}J_E \quad {}^{ws}\boldsymbol{\omega} = T(\boldsymbol{\Phi})\dot{\boldsymbol{\Phi}}$$
$$\det(T) = -\sin(\theta) \neq 0$$
$${}^{ws}\dot{J}_{A,E} = \begin{bmatrix} 0 & 0 \\ 0 & -T^{-1}\dot{T}T^{-1} \end{bmatrix} {}^{ws}J_E + \begin{bmatrix} I & 0 \\ 0 & T^{-1} \end{bmatrix} {}^{ws}\dot{J}_E$$

Substitute $\ddot{\boldsymbol{q}}$ in the operational space description

$${}^{ws}\ddot{\boldsymbol{x}} = \underbrace{{}^{ws}J_{A,E}B^{-1}({}^{b}J_S^{T})}_{\Lambda_A^{-1}}{}^{b}\gamma + {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$

$$\Lambda_A{}^{ws}\ddot{\boldsymbol{x}} = \gamma + \Lambda_A{}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$
$$\gamma_{cmd} = \Lambda_A\boldsymbol{a} - \Lambda_A{}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}}$$
$$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^{b}J_S^{T}(\Lambda_A\boldsymbol{a}_{cmd} - \Lambda_A{}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}} + {}^{b}\boldsymbol{w}_S) \Longrightarrow {}^{ws}\ddot{\boldsymbol{x}} = \boldsymbol{a}$$

# Issues with internal motions

In the case of the task described the Kuka LWR 4+ is a kinematically redundant manipulator.

Extensive simulations revealed that the uncontrolled internal motions of the robot, while not affecting the desired attitude of the hand, cause the 5-th and 7-th links to rotate cooperatively and reach their limits soon. Another issue with internal motions is that they could cause, in some situations, collisions between the 4-th link and the table which is part of the workspace of the robot.

These issues are solved using a dynamically consistent generalized inverse of the Jacobian [Khatib, 1987]

# Control of internal motions [Khatib, 1987]

The standard operational space dynamics can be written by substituting (1) in (2) resulting in (3)

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} = J^T\gamma \quad (1)$$

$$\ddot{\boldsymbol{x}} = J(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \quad (2)$$

$$\Lambda\ddot{\boldsymbol{x}} + \boldsymbol{\mu}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{p} = \gamma \quad (3)$$

$$\Lambda = (JB^{-1}J^T)^{-1}$$
$$\boldsymbol{\mu} = \bar{J}^T C\dot{\boldsymbol{q}} - \Lambda\boldsymbol{h}$$
$$\boldsymbol{p} = \bar{J}^T\boldsymbol{G}$$
$$\bar{J} = B^{-1}J^T\Lambda$$

The equation (3) can also be written in the form

$$\bar{J}^T(B\ddot{\boldsymbol{q}} + C\dot{\boldsymbol{q}} + \boldsymbol{G}) = \boldsymbol{\gamma}$$

resulting in

$$\gamma = \bar{J}^T\boldsymbol{\tau}$$

i.e. $\bar{J} = B^{-1}J^T\Lambda$ is a dynamically consistent generalized inverse of the Jacobian matrix

# Control of internal motions [Khatib, 1987]

In order to control the undesired internal motions a command torque $\boldsymbol{\tau}$ of the form

$$\boldsymbol{\tau} = J^T \gamma + (I_7 - J^T \bar{J}^T)\gamma_0$$

can be used where $\boldsymbol{\gamma}_0$ is projected in the null space of $\bar{J}^T$ hence not affecting the command wrench seen by the hand

The final control law is

$$\boldsymbol{\tau} = C\dot{\boldsymbol{q}} + \boldsymbol{G} + {}^bJ_S^T(\Lambda_A \boldsymbol{a}_{cmd} - \Lambda_A {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}} + {}^b\boldsymbol{w}_S) + (I_7 - ({}^bJ_S^T)({}^b\bar{J}_S^T))\gamma_0$$

where

$${}^b\bar{J}_S = B^{-1}({}^bJ_S^T)({}^b\Lambda_S) \quad {}^b\Lambda_S = ({}^bJ_S B^{-1}({}^bJ_S^T))^{-1}$$

# Design of the null operational wrench command $\gamma_0$

To fix the orientation of the 5-th link and to regulate the altitude between the 4-th link and the hand $\gamma_0$ is chosen

$$\gamma_0 = J_{im}^T (K_p(\boldsymbol{x}_{im,des} - \boldsymbol{x}_{im}) - K_d \dot{\boldsymbol{x}}_{im})$$

$$\boldsymbol{x}_{im} = \begin{bmatrix} {}^b x_{l4} & {}^b y_{l4} & {}^b z_{l4} & \psi_{l5} & \theta_{l5} & \phi_{l5} \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{p}_{l4}^T & \boldsymbol{\Phi}_{l5}^T \end{bmatrix}^T$$

$$\begin{bmatrix} \dot{\boldsymbol{p}}_{l4} \\ \dot{\boldsymbol{\Phi}}_{l5} \end{bmatrix} = J_{im} \dot{\boldsymbol{q}}$$

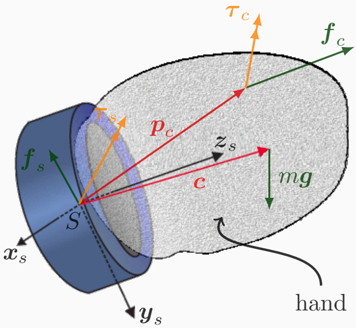
$$K_p = \mathrm{diag}(0, 0, k_{p,z}^{im}, 0, 0, k_{p,att}^{im}) \quad K_d = k_d^{im} I_6$$

$$\boldsymbol{p}_{l4} = \begin{bmatrix} * & * & {}^b p_{ee_z} + off_z \end{bmatrix}^T \quad \boldsymbol{\Phi}_{l5} = \begin{bmatrix} * & * & 0 \end{bmatrix}^T$$

# Measure of forces/torques exchanged with the object

# Force/torque sensor with hand attached

The wrench $\boldsymbol{w}_S$ exerted by the hand on the object is required to implement the controllers properly




-$\boldsymbol{f}_s$, -$\boldsymbol{\tau}_s$ := measured forces

-$\boldsymbol{f}_c$, -$\boldsymbol{\tau}_c$ := contact forces

$\boldsymbol{f}_{pl}$, $\boldsymbol{\tau}_{pl}$ := due to mounting plate [1]

$\boldsymbol{c}$ := CoM of the hand

$\boldsymbol{p}_c$ := vector from $S$ to contact point

$$\boldsymbol{w}_S = -\begin{bmatrix} \boldsymbol{f}_c \\ \boldsymbol{\tau}_c + \tilde{\boldsymbol{p}}_c \boldsymbol{f}_c \end{bmatrix}$$

[1] Not shown in the picture

# Newton-Euler equations for a rigid body attached to the sensor

The Newton-Euler equations show how to obtain $\boldsymbol{w}_S$ from $\boldsymbol{f}_s$ and $\boldsymbol{\tau}_s$

$$\boldsymbol{f}_s = -\boldsymbol{f}_{pl} - \boldsymbol{f}_c - m\boldsymbol{g} + m\boldsymbol{a}_{cm}$$

$$\boldsymbol{\tau}_s = -\boldsymbol{\tau}_{pl} - \boldsymbol{\tau}_c - \tilde{\boldsymbol{p}}_c \boldsymbol{f}_c + \tilde{\boldsymbol{g}} m\boldsymbol{c} - \tilde{\boldsymbol{a}}_{cm} m\boldsymbol{c} \quad [2]$$

[2] Angular velocites and accelerations neglected because the f/t sensor is primarily used when the hand is still or when it moves along straight lines

# Software induced offset

The signal produced by the f/t sensor is

$$\boldsymbol{f}_{meas} = -\boldsymbol{f}_s + \boldsymbol{f}_{sw,off}$$

$$\boldsymbol{\tau}_{meas} = -\boldsymbol{\tau}_s + \boldsymbol{\tau}_{sw,off}$$

Offsets $\boldsymbol{f}_{sw,off}$, $\boldsymbol{\tau}_{sw,off}$ are set when the f/t sensor is calibrated such that

$$\boldsymbol{f}_{m,0} = \boldsymbol{\tau}_{m,0} = \mathbf{0}^3$$

$$\boldsymbol{f}_{sw,off} = \boldsymbol{f}_{s,0} = -\boldsymbol{f}_{pl} - m\,{}^s\boldsymbol{g}_0$$

$$\boldsymbol{\tau}_{sw,off} = \boldsymbol{\tau}_{s,0} = -\boldsymbol{\tau}_{pl} + {}^s\tilde{\boldsymbol{g}}_0 m\boldsymbol{c}$$

The resulting signal is

$$\boldsymbol{f}_{meas} = \boldsymbol{f}_c + m\boldsymbol{g} - m\,{}^s\boldsymbol{g}_0 - m\boldsymbol{a}_{cm}$$

$$\boldsymbol{\tau}_{meas} = \boldsymbol{\tau}_c + \tilde{\boldsymbol{p}}_c\boldsymbol{f}_c - \tilde{\boldsymbol{g}}m\boldsymbol{c} + {}^s\tilde{\boldsymbol{g}}_0 m\boldsymbol{c} + \tilde{\boldsymbol{a}}_{cm}m\boldsymbol{c}$$

---

[3] The zero subscript represents the calibration condition performed when the manipulator is still, i.e. $\boldsymbol{a}_{cm} = \mathbf{0}$

# Estimation of unknowns quantities

In order to obtain $\boldsymbol{f}_c$ and $\boldsymbol{\tau}_c$ the unkowns are estimated

$$\boldsymbol{f}_{meas} = \boldsymbol{f}_c + m\boldsymbol{g} + (-m\,{}^s\boldsymbol{g}_0) - m\boldsymbol{a}_{cm}$$

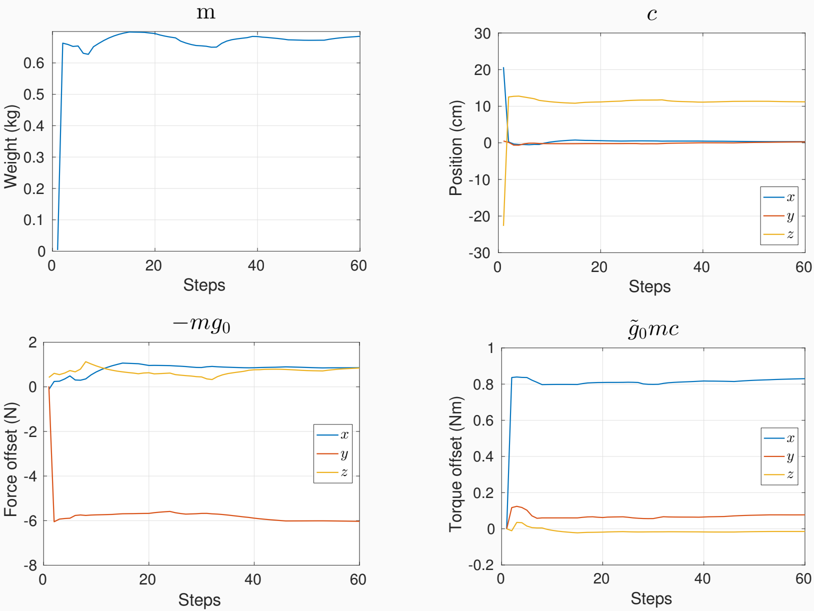
$$\boldsymbol{\tau}_{meas} = \boldsymbol{\tau}_c + \tilde{\boldsymbol{p}}_c\boldsymbol{f}_c - \tilde{\boldsymbol{g}}m\boldsymbol{c} + {}^s\tilde{\boldsymbol{g}}_0 m\boldsymbol{c} + \tilde{\boldsymbol{a}}_{cm}m\boldsymbol{c}$$

The estimation is obtained from $n$ measurements collected when the manipulator assumes a static pose, i.e. $\boldsymbol{a}_{cm} = \boldsymbol{0}$

$$\begin{bmatrix} \boldsymbol{f}_{meas} \\ \boldsymbol{\tau}_{meas} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}m + (-m\,{}^s\boldsymbol{g}_0) \\ -\tilde{\boldsymbol{g}}m\boldsymbol{c} + ({}^s\tilde{\boldsymbol{g}}_0 m\boldsymbol{c}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{g} & 0_{3\times3} & I_{3x3} & 0_{3x3} \\ \boldsymbol{0} & -\tilde{\boldsymbol{g}} & 0_{3x3} & I_{3x3} \end{bmatrix} \begin{bmatrix} m \\ m\boldsymbol{c} \\ -m\,{}^s\boldsymbol{g}_0 \\ {}^s\tilde{\boldsymbol{g}}_0 m\boldsymbol{c} \end{bmatrix} = H({}^s\boldsymbol{g}(\boldsymbol{q}))\boldsymbol{\theta} = H(\boldsymbol{q})\boldsymbol{\theta}$$

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} H(\boldsymbol{q}^1) \\ \vdots \\ H(\boldsymbol{q}^n) \end{bmatrix}^+ \begin{bmatrix} \boldsymbol{f}_m^1 \\ \boldsymbol{\tau}_m^1 \\ \vdots \\ \boldsymbol{f}_m^n \\ \boldsymbol{\tau}_m^n \end{bmatrix} = H_n^+ \begin{bmatrix} \boldsymbol{f}_m^1 \\ \boldsymbol{\tau}_m^1 \\ \vdots \\ \boldsymbol{f}_m^n \\ \boldsymbol{\tau}_m^n \end{bmatrix}$$

# Estimation of unknowns quantities - An example



# Approaching phase

# Approaching phase

Before using HIC the robot is moved in a given position above the surface of the table using a joint space inverse dynamics to avoid singularity introduced by Euler ZYZ parametrization

$$B(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} \longrightarrow \ddot{\boldsymbol{q}} = \boldsymbol{a}$$

$$\begin{aligned}
\boldsymbol{\tau} &= C(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + B(\boldsymbol{q})\boldsymbol{a}_{p2p} \\
\boldsymbol{a}_{p2p} &= \ddot{\boldsymbol{q}}_{des} + K_d(\dot{\boldsymbol{q}}_{des} - \dot{\boldsymbol{q}}) + K_p(\boldsymbol{q}_{des} - \boldsymbol{q}) \\
\boldsymbol{q}_f &= FK^{-1}(\boldsymbol{x}_{des})
\end{aligned}$$

where $\boldsymbol{q}_f$ is discarded whenever its component exceeds the joints limits

$$\begin{aligned}
&q_{des,i}(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\
&q_{des,i}(0) = q_{prev,i} \quad \dot{q}_{des,i}(0) = 0 \quad \ddot{q}_{des,i}(0) = 0 \\
&q_{des,i}(t_f) = q_{f,i} \quad \dot{q}_{des,i}(t_f) = 0 \quad \ddot{q}_{des,i}(t_f) = 0
\end{aligned}$$

# Results of simulation

# Simulation setup

- emulated Kuka FRI Joint specific impedance control mode

$$\boldsymbol{\tau}_{cmd} = K_j(\boldsymbol{q}_{FRI} - \boldsymbol{q}_{msr}) + D(d_j) + \boldsymbol{\tau}_{FRI} + \boldsymbol{f}_{dynamics}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$$

- $K_j = 0$
- $\boldsymbol{q}_{FRI} = \boldsymbol{q}_{msr}$
- $d_j = 0$
- $\boldsymbol{f}_{dynamics}$ supposed to be $\boldsymbol{G}$ evaluated using KDL library [4]
- $\boldsymbol{\tau}_{FRI}$ used as commanded torque

[4] The Kinematics and Dynamics Library (KDL) develops an application independent framework for modelling and computation of kinematic chains

# Simulation setup (continued)

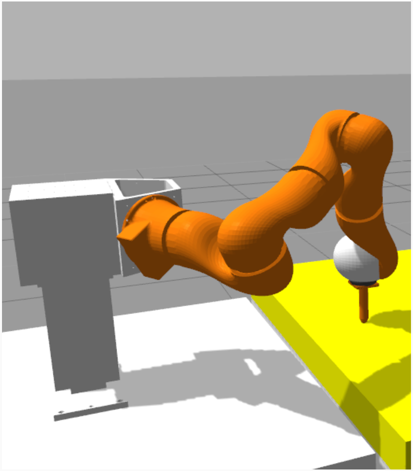
$$\boldsymbol{\tau}_{FRI} = C\dot{\boldsymbol{q}} + \cancel{\boldsymbol{G}} + B\boldsymbol{a}_{p2p}$$

$$\boldsymbol{\tau}_{FRI} = C\dot{\boldsymbol{q}} + \cancel{\boldsymbol{G}} + {}^{b}J_S^T(B_A\boldsymbol{a}_{cmd} - B_A {}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}} + {}^{b}\boldsymbol{w}_S) + (I_7 - ({}^{b}J_S^T)({}^{b}\bar{J}_S^T))\gamma_0$$

- controllers running @ 1kHz
- $G$ canceled because the gravity is compensated by the robot
- $B$ given by KDL
- $C\dot{\boldsymbol{q}}$ given by KDL
- only $\boldsymbol{q}$ is available (as in the real scenario)
  - $\dot{\boldsymbol{q}}$ estimated using an exponential smoothing $\dot{\boldsymbol{q}}_k = (1-\alpha)\dot{\boldsymbol{q}}_{k-1} + \alpha\frac{\boldsymbol{q}_k - \boldsymbol{q}_{k-1}}{t_s}$
- Jacobians are evalutated using KDL library
- $\boldsymbol{w}_F$ given by force/torque sensor sofware plugin
  - corrupted signal when the end-effector is mounted on the sensor $\longrightarrow$ massless tool
- joints friction and contact friction neglected for ease of simulation
- generic end-effector instead of the hand for ease of simulation

## Description of the simulated scenarios

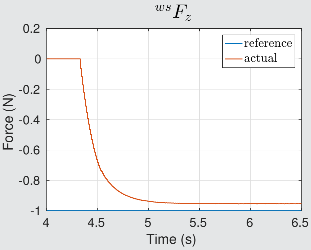
The following scenarios were simulated in the Gazebo robot simulator

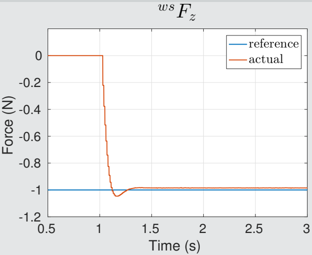


- contact phase
- force regulation
- simultaneous force regulation and dragging

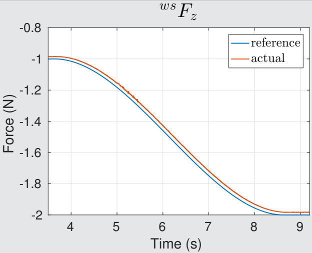
# Results - Force regulation

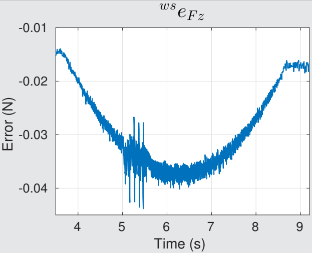
## Contact phase $K_f \in \{2, 3\}, K_d = 25$



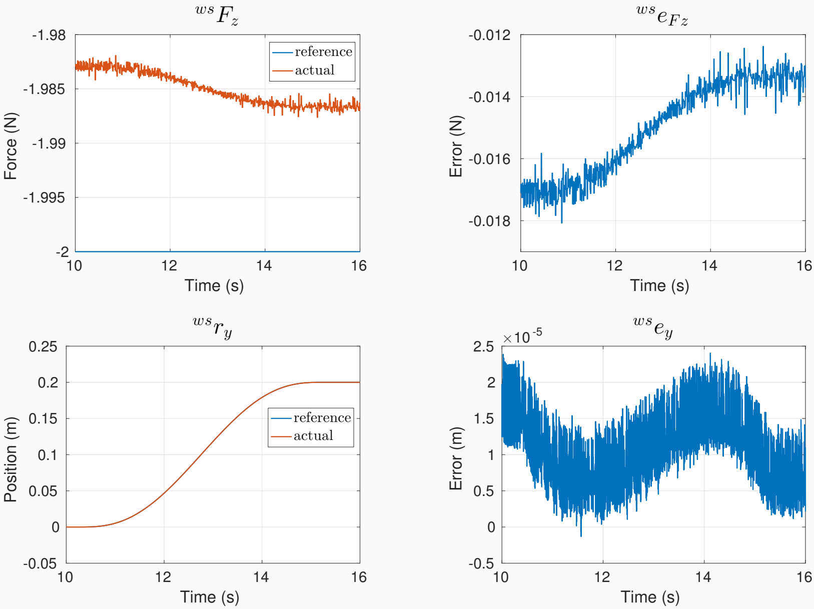




## Force setpoint $K_f = 3, B_f = 45$



# Results - Dragging and Force regulation $(K_f = 3, B_f = 45)$



# Results of experiments

# Experimental setup

- Kuka FRI Joint specific impedance control mode

$$\boldsymbol{\tau}_{cmd} = K_j(\boldsymbol{q}_{FRI} - \boldsymbol{q}_{msr}) + D(d_j) + \boldsymbol{\tau}_{FRI} + \boldsymbol{f}_{dynamics}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$$

- $K_j = 0$
- $\boldsymbol{q}_{FRI} = \boldsymbol{q}_{msr}$
- $d_j = 0$
- $\boldsymbol{f}_{dynamics}$ supposed to be $\boldsymbol{G}$
- $\boldsymbol{\tau}_{FRI}$ used as commanded torque

# Experimental setup (continued)

$$\boldsymbol{\tau}_{FRI} = C\dot{\boldsymbol{q}} + \cancel{\boldsymbol{G}} + B\boldsymbol{a}_{p2p}$$

$$\boldsymbol{\tau}_{FRI} = C\dot{\boldsymbol{q}} + \cancel{\boldsymbol{G}} + {}^{b}J_S^T(B_A\boldsymbol{a}_{cmd} - B_A{}^{ws}\dot{J}_{A,E}\dot{\boldsymbol{q}} + {}^{b}\boldsymbol{w}_S) + (I_7 - ({}^{b}J_S^T)({}^{b}\bar{J}_S^T))\gamma_0$$

- controllers running @ 300Hz
- $G$ canceled because the gravity is compensated by the robot
- $B$ given by KDL
- $C\dot{\boldsymbol{q}}$ given by KDL
- only $\boldsymbol{q}$ is available (as in the real scenario)
    - $\dot{\boldsymbol{q}}$ estimated using an exponential smoothing $\dot{\boldsymbol{q}}_k = (1-\alpha)\dot{\boldsymbol{q}}_{k-1} + \alpha\frac{\boldsymbol{q}_k - \boldsymbol{q}_{k-1}}{t_s}$
- Jacobians are evalutated using KDL library
- $\boldsymbol{w}_S$ given by force/torque sensor

# Experiment n.1 - Force regulation

# Experiment n.2 - Force regulation

# Experiment n.3 - Dragging with force regulation and grasp

# Software implementation

## Some notes about implementation

- ROS based software
- already existing (Centro Piaggio) KUKA LWR4+ software stack (model, FRI, F/T sensor interface, ROS controllers)
- built upon KinematicChainControllerBase (KDL facilities)
- extends KinematicChainControllerBase by providing an Operational Space Inverse Dynamics *abstract* controller
    - arbitrary commanded acceleration in operational space
    - state (with derivatives) ready to use and projected in *ws*
    - jacobian extended to take into account the length of the tool
    - null projection available
- Hybrid Impedance Controller based on Inverse Dynamics Controller

# Future works

## Combine HIC with a High-Level planner

A natural extension of this work would be to combine the HIC controller with a high-level planner that provide the reference forces and positions to perform complex tasks where the interaction with the environment is required.

# Appendix: Kinestatic Filtering

# How to write $a_{cmd}$ using a selection matrix $S$

A selection matrix $S$ may be used to separate the force-controlled and position-controlled *reciprocal* subspaces

$$
{}^{ws}\boldsymbol{a}_{cmd} = S \begin{bmatrix} a_{cmd,x} \\ a_{cmd,y} \\ * \\ \boldsymbol{a}_{cmd,\Phi} \end{bmatrix} + (I - S) \begin{bmatrix} * \\ * \\ a_{cmd,z} \\ * \\ * \\ * \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}
$$

# A different interpretation of the selection matrix

Position controlled and force controlled subspaces can be seen in terms of *natural* and *artificial* constraints:

- natural: directions along which the end effector can not move and exert forces
- artificial: directions along which the end effector can move and exert forces

Artificial constraints directions belong to the span of appropriate matrices

# An example

Consider admissible twists and wrenches projected in *ws*

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad a = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \begin{matrix} {}^{ws}\boldsymbol{\xi}_{adm} \in \mathcal{R}(B) \\ {}^{ws}\boldsymbol{w}_{adm} \in \mathcal{R}(a) \end{matrix}$$

The most general motion is that of a translation ($x$-$y$ direction of *ws*) and/or a rotation about some axis that pass through a point $P$, e.g. the center of the palm of the hand.

Force can only be exerted along the $z$ direction of *ws*. No torques are allowed.

# An example (continued)

In order to filter out undesired twists and wrenches the following projection matrices are used (Kinestatic filter)

$$P_B = B(B^T B)^{-1} B^T \qquad P_a = a(a^T a)^{-1} a^T$$

$$P_B \boldsymbol{e}_i = \boldsymbol{e}_i \quad i \in \{1,2,4,5,6\} \qquad P_a \boldsymbol{e}_i = \mathbf{0} \quad i \in \{1,2,4,5,6\}$$

$$P_B \boldsymbol{e}_3 = \mathbf{0} \qquad P_a \boldsymbol{e}_3 = \boldsymbol{e}_3$$

The projectors lead to the already seen selection matrices $S = P_B$ and $I - S = P_a$

# A more general example

In general twists/wrenches can be commanded with respect to a different reference point $E'$ for example $EE' = \begin{bmatrix} \bar{x} & 0 & 0 \end{bmatrix}^T$

$$
{}^{ws}\boldsymbol{\xi}' = K(\bar{x})({}^{ws}\boldsymbol{\xi}) \qquad {}^{ws}\boldsymbol{w}' = G(\bar{x})({}^{ws}\boldsymbol{w})
$$

$$
B' = KB = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & x \\ 0 & 0 & 0 & -x & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad a' = Ga = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ x \\ 0 \end{bmatrix}
$$

However using the standard projector gives for example

$$
P'_B K \boldsymbol{e}_3 = \begin{bmatrix} 0 & 0 & \frac{x^2}{x^2+1} & 0 & -\frac{x}{x^2+1} & 0 \end{bmatrix}^T \qquad P'_a G \boldsymbol{e}_5 = \begin{bmatrix} 0 & 0 & \frac{x}{x^2+1} & 0 & \frac{x^2}{x^2+1} & 0 \end{bmatrix}^T
$$

# An invariant filter [Marescotti, Bonivento and Melchiorri, 1990]

An invariant filter behaves well even when the reference point changes

**$P''_B = K P_B K^{-1}$**

$$
\begin{aligned}
\text{If } & P_B({}^{ws}\boldsymbol{\xi}) = {}^{ws}\boldsymbol{\xi} \\
\text{then } & P''_B({}^{ws}\boldsymbol{\xi}') = P''_B K({}^{ws}\boldsymbol{\xi}) \\
& = K P_B K^{-1} K({}^{ws}\boldsymbol{\xi}) \\
& = K P_B({}^{ws}\boldsymbol{\xi}) \\
& = K({}^{ws}\boldsymbol{\xi}) = {}^{ws}\boldsymbol{\xi}'
\end{aligned}
$$

$$
\begin{aligned}
\text{If } & P_B({}^{ws}\boldsymbol{\xi}) = \mathbf{0} \\
\text{then } & P''_B({}^{ws}\boldsymbol{\xi}') = K P_B({}^{ws}\boldsymbol{\xi}) \\
& = \mathbf{0}
\end{aligned}
$$

**$P''_a = G P_a G^{-1}$**

$$
\begin{aligned}
\text{If } & P_a({}^{ws}\boldsymbol{w}) = {}^{ws}\boldsymbol{w} \\
\text{then } & P''_a({}^{ws}\boldsymbol{w}') = P''_a G({}^{ws}\boldsymbol{w}) \\
& = G P_a G^{-1} G({}^{ws}\boldsymbol{w}) \\
& = G P_a({}^{ws}\boldsymbol{w}) \\
& = G({}^{ws}\boldsymbol{w}) = {}^{ws}\boldsymbol{w}'
\end{aligned}
$$

$$
\begin{aligned}
\text{If } & P_a({}^{ws}\boldsymbol{w}) = \mathbf{0} \\
\text{then } & P''_a({}^{ws}\boldsymbol{w}') = G P_a({}^{ws}\boldsymbol{w}) \\
& = \mathbf{0}
\end{aligned}
$$

# Kinestatic filtering

Since the kinestatic filtering is applied to twists and wrenches it is not possible to apply it directly to the commanded accelerations described above.

Anyway, as observed in [Marescotti, Bonivento and Melchiorri, 1990], there is no necessity to use the alternative projectors $P''_B$ and $P''_a$ in the control algorithm since twists and wrenches expressed with respect to another reference point can always be transformed back to their original form using the inverse transformations $K^{-1}$ and $G^{-1}$.

In that case the projection matrices always reduce to the selection matrices $S$ and $I - S$ that are also suitable to filter undesired commanded accelerations.