IMSD 2016

Modeling of flexible link robots for end-effector trajectory tracking tasks

Arthur Lismonde[1], Valentin Sonneville[2] and Olivier Brüls[1]

[1]University of Liège, Belgium
Department of Aerospace and Mechanical engineering

[2]University of Maryland, USA
Aerospace engineering

Context

Current trend in robotics:
- Optimize energy consumption.
- Improve **mass to pay-load ratio**.
- Improve safety of robots.

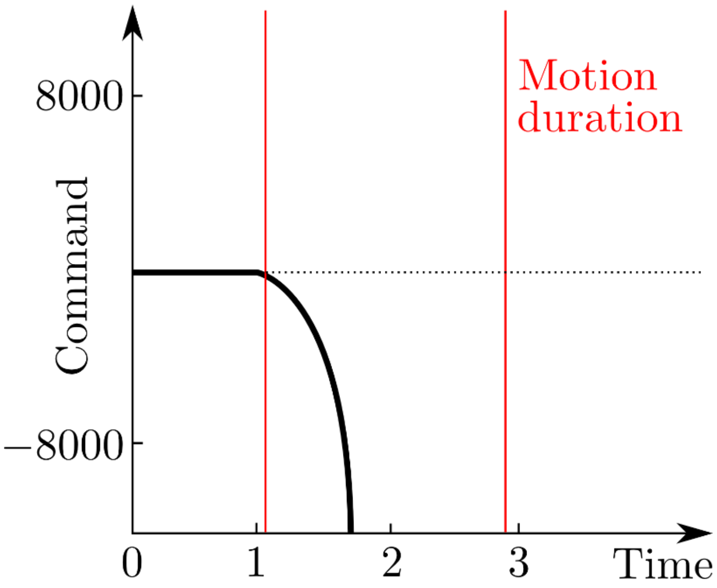
Design of **lightweight and compliant robots** that could have elastic deformation issues.

⇒ **Elastic deformation and vibrations** can be taken care of **using** the control system: **feedback** and **feed-forward** commands.

Context

- Computation of a **feed-forward action** that takes into account elastic deformations.
- For trajectory tracking tasks, the **inverse dynamics** of such flexible multibody systems (MBS) has **to be solved**.

- But **flexible systems** might have **unstable internal dynamics** (non-minimum phase).
- Time integration methods lead to **unbounded solutions**.

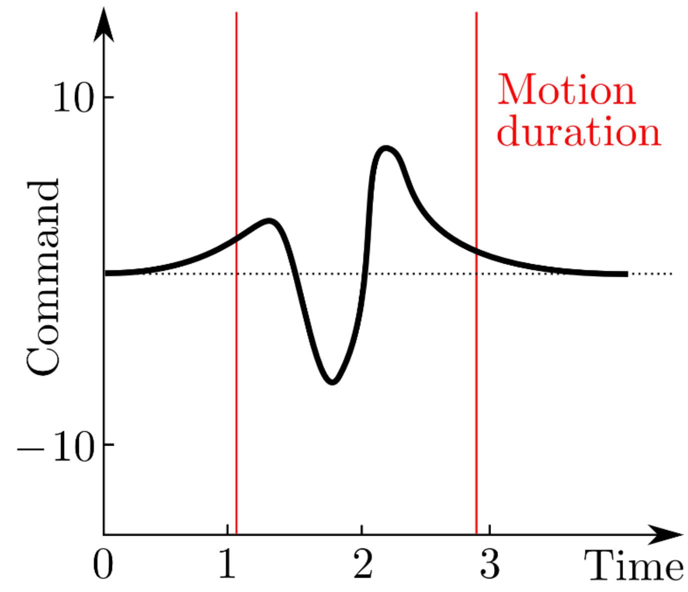


Context

- **Particular methods** to solve the inverse dynamics of flexible MBS:

  1. **Stable inversion** method: Boundary value problem, [Seifried 2013], [Devasia et al 1996].
  2. **Optimal control** method: Minimization of objective function, [Bastos et al 2013] for **2D systems**.

- Both lead to similar solution that are **non-causal** but **bounded**.



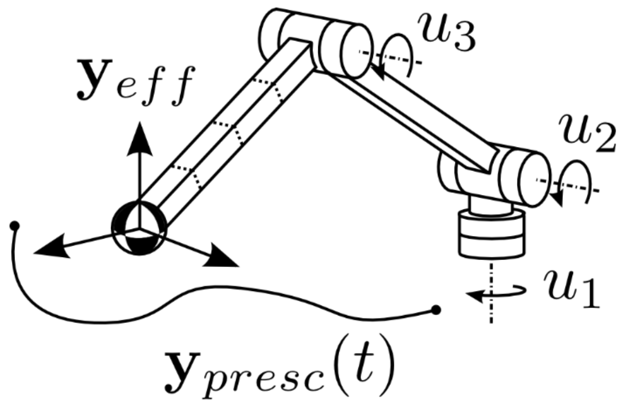
Objective and originality

- Solve the **inverse dynamics** of **3D flexible MBS** for **trajectory tracking** tasks:
  1. **Model** using finite elements with **SE(3) formalism** to avoid direct parameterization of 3D equation of motion.
  2. **Optimal control formulation** to avoid the definition of suitable boundary conditions (as for the stable inversion).

- The **method** is developed to deal with **general flexible MBS**: serial or parallel systems, localized or distributed flexibility, 1-6 controls,…

Formulation

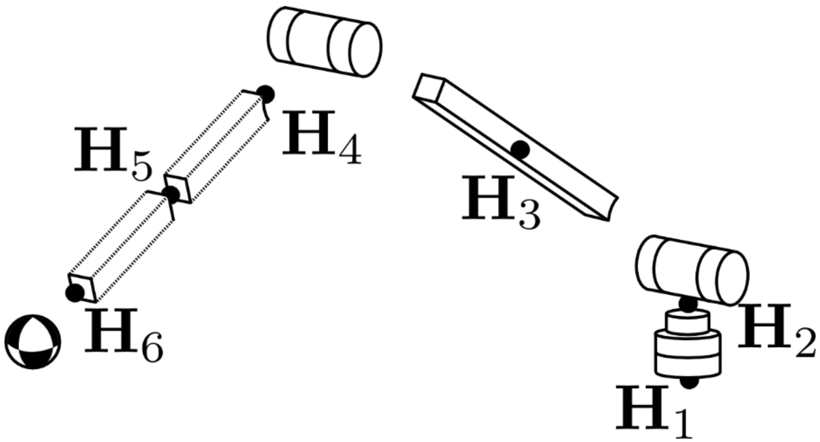
- **To illustrate**, we first consider the following **3D serial manipulator**:
  1. Robot with 3 dof.
  2. Made up of a rigid and a flexible link.
  3. Point mass end-effector that follows a prescribed trajectory over time.



Formulation

Finite element formalism on SE(3): [Sonneville et al 2014, 2015]

- Rigid, flexible elements and kinematic joints.
- Spatial discretization of flexible bodies (e.g., beam finite elements).
- Kinematics described using position & orientation of the N nodes.



$$\mathbf{H}_I = \begin{pmatrix} \mathbf{R}_I & \mathbf{p}_I \\ \mathbf{0} & 1 \end{pmatrix} \in SE(3)$$

with $\begin{cases} \mathbf{R}_I \in SO(3) \\ \mathbf{p}_I \in \mathbb{R}^3 \end{cases}$

Configuration variable: $\mathbf{H} = \mathrm{diag}(\mathbf{H}_1, \ldots, \mathbf{H}_N)$

Formulation

Optimal control problem as a NLP problem:

- **Objective** function ***J***: **strain energy** and **controls** of the system.

$$\min_{\mathbf{H}} J = \min_{\mathbf{H}} \frac{1}{2T} \int_{t_i}^{t_f} [\boldsymbol{\epsilon}^T(\mathbf{H})\mathbf{K}\boldsymbol{\epsilon}(\mathbf{H}) + \mathbf{u}^T\mathbf{G}\mathbf{u}]dt$$

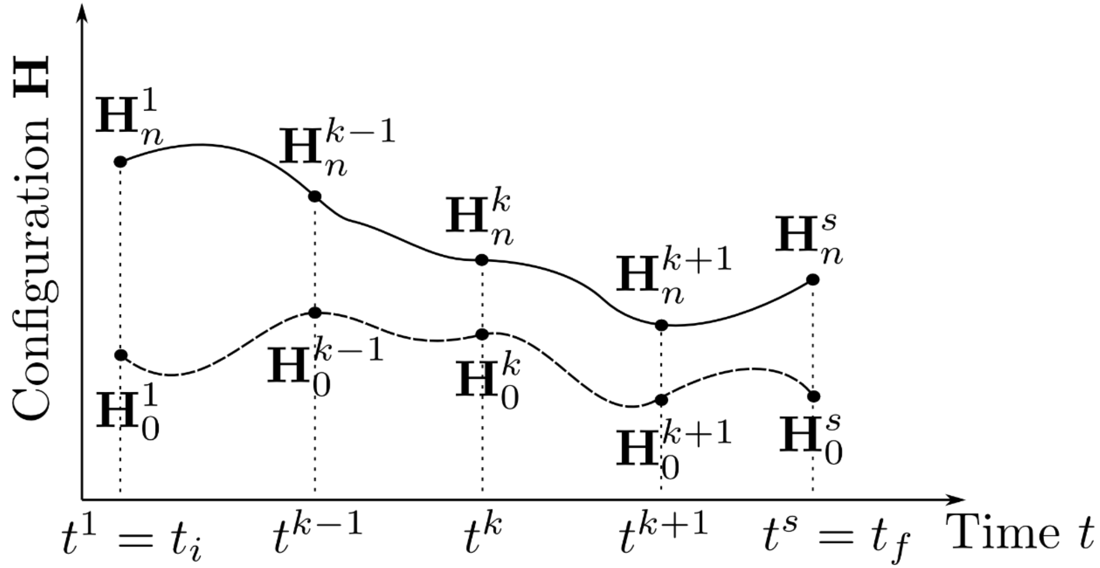
- **Constraints** of the NLP: equation of inverse dynamics of the MBS.

$\dot{\mathbf{H}}_I = \mathbf{H}_I\tilde{\mathbf{v}}_I$	Compatibility equation
$\mathbf{M}\dot{\mathbf{v}} + \mathbf{g}(\mathbf{H}, \mathbf{v}) + \mathbf{B}^T\boldsymbol{\lambda} = \mathbf{A}\mathbf{u}$	Equation of motion
$\boldsymbol{\Phi}(\mathbf{H}) = \mathbf{0}$	Kinematic constraints
$\mathbf{y}_{eff}(\mathbf{H}) - \mathbf{y}_{presc}(t) = \mathbf{0}$	"Servo" constraints

Method

Use of the **direct transcription method**:

- **Discretization** of all states into "**s**" time steps (of size "h").
- Optimization variables at **each time steps** "$\boldsymbol{k}$" ($k = 1, \dots, s$).
- Optimum after "$\boldsymbol{n}$" iterations.



Method

Resulting **discrete form** of the NLP problem:

- Objective function J:

$$\min_{\mathbf{H}^1,\dots,\mathbf{H}^s} J = \min_{\mathbf{H}^1,\dots,\mathbf{H}^s} \frac{1}{2T}\sum_{k=1}^{s}[\boldsymbol{\epsilon}^{k,T}(\mathbf{H}^k)\mathbf{K}\boldsymbol{\epsilon}^k(\mathbf{H}^k) + \mathbf{u}^{k,T}\mathbf{G}\mathbf{u}^k]h$$

- Constraints of the NLP:

$$\begin{aligned}
\dot{\mathbf{H}}_I^k - \mathbf{H}_I^k\tilde{\mathbf{v}}_I^k &= \mathbf{0}\\
\mathbf{M}^k\dot{\mathbf{v}}^k + \mathbf{g}(\mathbf{H}^k,\mathbf{v}^k) + \mathbf{B}^{k,T}\boldsymbol{\lambda}^k - \mathbf{A}\mathbf{u}^k &= \mathbf{0}\\
\boldsymbol{\Phi}(\mathbf{H}^k) &= \mathbf{0}\\
\mathbf{y}_{eff}(\mathbf{H}^k) - \mathbf{y}_{presc}(t^k) &= \mathbf{0}
\end{aligned}$$

Method

Need for **additional time constraints** between time steps:

- Standard methods (e.g., **Euler-implicit** type) are **not acceptable**:

$$\mathbf{H}_I^{k+1} = \mathbf{H}_I^k + h\dot{\mathbf{H}}_I^{k+1} \notin SE(3)$$

- **On $SE(3)$**, exponential mapping can be used:

$$\mathbf{H}_I^{k+1} = \mathbf{H}_I^k \exp_{SE(3)}(\widetilde{\Delta\mathbf{Q}_I^k}) \in SE(3)$$

where $\Delta\boldsymbol{Q}^k = (\Delta\boldsymbol{Q}_1^k, \dots, \Delta\boldsymbol{Q}_N^k)$ is the **change in configuration between two consecutive times k and $k+1$**.

Method

- Possible set of optimization variables would be

$$(\mathbf{H}^1, \mathbf{v}^1, \dot{\mathbf{v}}^1, \boldsymbol{\lambda}^1, \mathbf{u}^1, \ldots, \mathbf{H}^s, \mathbf{v}^s, \dot{\mathbf{v}}^s, \boldsymbol{\lambda}^s, \mathbf{u}^s)$$

with $\boldsymbol{H} \in SE(3)$, which **can not be dealt with** by classical NLP solvers.

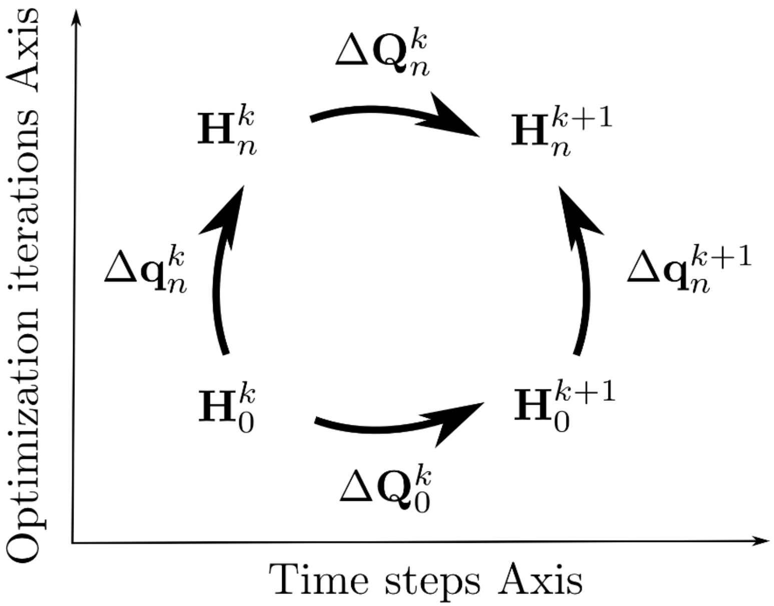
- Alternative: use of a **new vector** $\Delta\boldsymbol{q} = (\Delta\boldsymbol{q}_1, \ldots, \Delta\boldsymbol{q}_N)$ which is the **change in configuration between the initial guess states and the optimized states**, through the exponential mapping

$$\mathbf{H}^k_{I,n} = \mathbf{H}^k_{I,0} \exp_{SE(3)}(\widetilde{\Delta\mathbf{q}^k_{I,n}})$$

- **At the end**, the optimization variables are

$$\mathbf{x} = (\Delta\mathbf{q}^1, \mathbf{v}^1, \dot{\mathbf{v}}^1, \boldsymbol{\lambda}^1, \mathbf{u}^1, \ldots, \Delta\mathbf{q}^s, \mathbf{v}^s, \dot{\mathbf{v}}^s, \boldsymbol{\lambda}^s, \mathbf{u}^s)$$

Method



- ***Important remark:***

Difference between "**time**" related mapping

$$\mathbf{H}_I^{k+1} = \mathbf{H}_I^k \exp_{SE(3)}(\widetilde{\Delta \mathbf{Q}_I^k})$$

and "**iteration**" related mapping

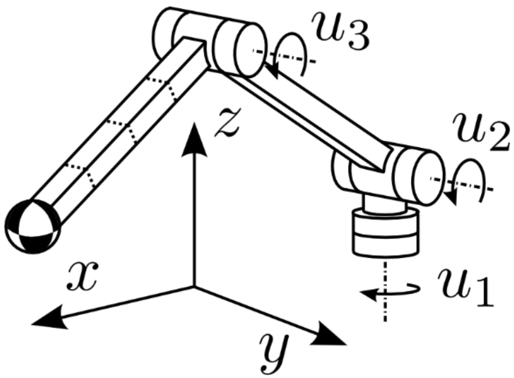
$$\mathbf{H}_{I,n}^k = \mathbf{H}_{I,0}^k \exp_{SE(3)}(\widetilde{\Delta \mathbf{q}_{I,n}^k})$$

Computation

- NLP problem is very **large but sparse**.
- Tests using NLP solvers such as KNITRO, IPOPT and **FMINCON**.
- Use of the "**interior point**" algorithm with large scale and sparse options and **analytical gradients are provided**.

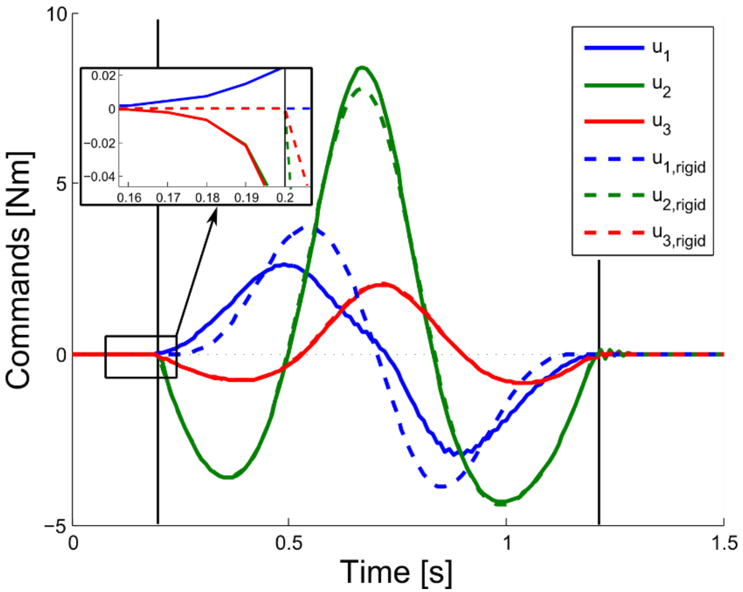
Results – Serial robot

- Serial robot with **3 dof**.
- Made up of 2 links:
  1. **Rigid link**:
     Alu, 1 x 0,02 x 0,02 m.
  2. **Flexible link** (4 beams):
     Alu, 1 x 0,005 x 0,005 m.
- **Point mass** at the end-effector (0,1 kg).
- Trajectory: **half-circle** with 0,5 m radius in the ***yz* plan**, to be completed in **1 s**.
- Analysis: 1^{st} unstable pole at 8 Hz.

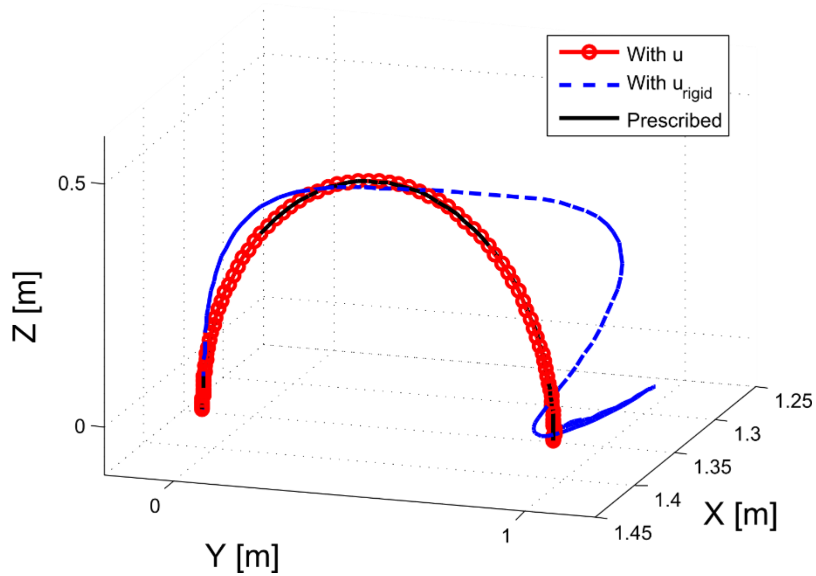


Results – Serial robot

s = 300 (h = 0,005s, 70k var.), 20 min. and RMS error from 5,2% to 0,1%

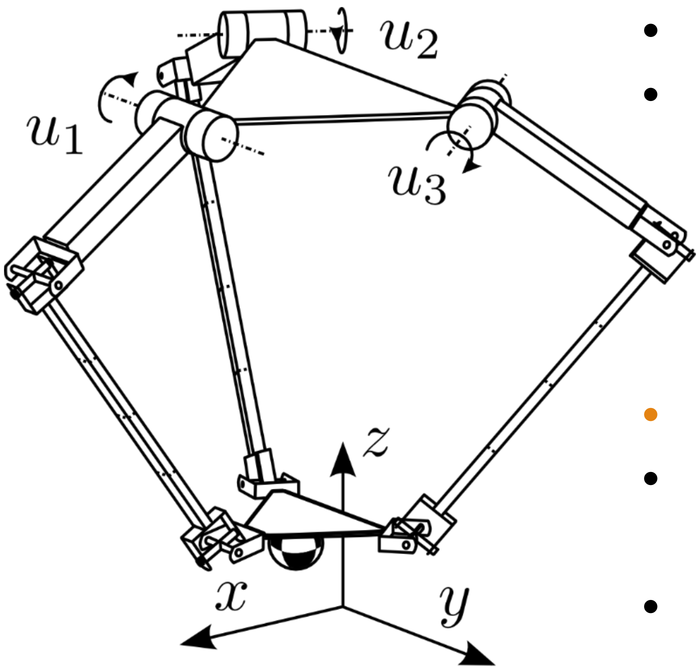


Commands before and after optimization



Actual trajectories with both commands

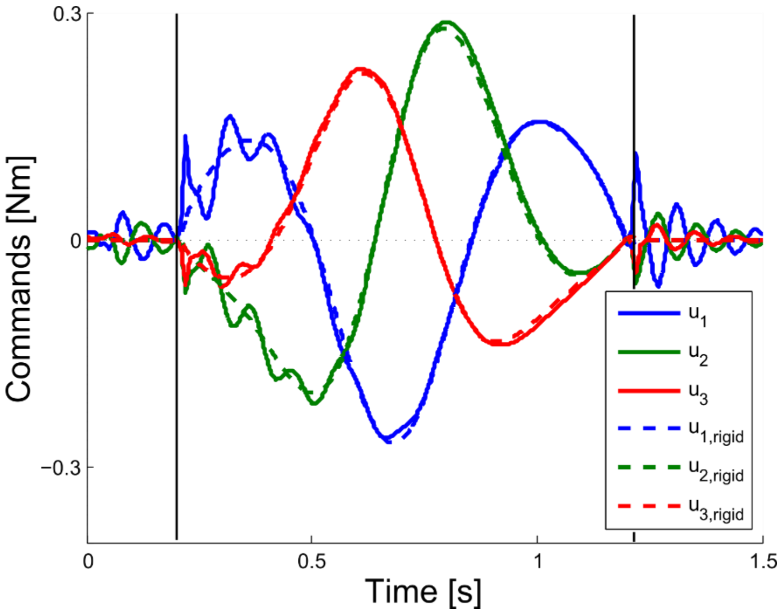
Results – Parallel robot



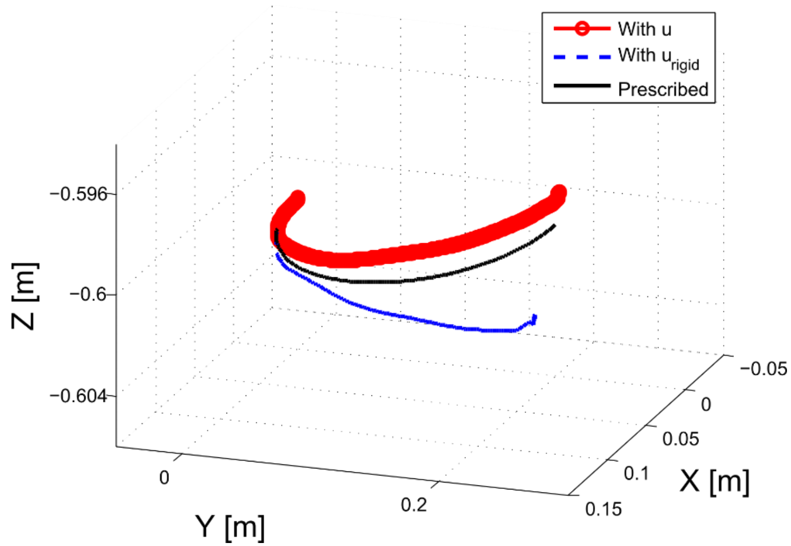
- Parallel robot with **3 dof**.
- Made up of 2 links:
  1. **Rigid links** (3):
     Alu, 0,25 x 0,02 x 0,02 m.
  2. **Flexible links** (3 x 4 beams):
     Alu, 0,51 x 0,005 x 0,005 m.
- **Point mass** at the end-effector (0,1 kg).
- Trajectory: **half-circle** with 0,1 m radius in the ***xy* plan**, to be completed in **1 s**.
- Analysis: 1^{st} unstable pole at 14 Hz.

Results – Parallel robot

s = 150 (h = 0,01s, 110k var.), 40 min. and RMS error from 2% to 0,7%



Commands before and after optimization



Actual trajectories with both commands

Summary

- **In this work:**
  - Use of a $SE(3)$ **formalism** to reduce non-linearity in the equations of **3D flexible MBS** problem.
  - Formulation of the inverse dynamics problem as an **optimal control problem**.
  - New **vectorial variable** is introduced to solve the optimization with **classical NLP tools**.
  - Method successful for **3D flexible serial** and **parallel** systems.

- **On going work and perspectives:**
  - Feed-forward solution on **robotic testbed** (adding feedback).
  - Consider compliant joints.
  - Consider contact problems with end-effector.

Thank you for your attention.

IMSD 2016

Modeling of flexible link robots for end-effector trajectory tracking tasks

Arthur Lismonde[1], Valentin Sonneville[2] and Olivier Brüls[1]

[1]University of Liège, Belgium
Department of Aerospace and Mechanical engineering

[2]University of Maryland, USA
Aerospace engineering