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

Arthur Lismonde\textsuperscript{1}, Valentin Sonneville\textsuperscript{2} and Olivier Brüls\textsuperscript{1}

\textsuperscript{1}University of Liège, Belgium
Department of Aerospace and Mechanical engineering

\textsuperscript{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.

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

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

Configuration variable: $H = \text{diag}(H_1, ..., H_N)$
Formulation

**Optimal control problem** as a NLP problem:
- **Objective** function $J$: strain energy and controls of the system.

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

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

\[
\dot{\mathbf{H}}_I = \mathbf{H}_I \ddot{\mathbf{v}}_I \quad \text{Compatibility equation}
\]

\[
\mathbf{M} \dot{\mathbf{v}} + \mathbf{g}(\mathbf{H}, \mathbf{v}) + \mathbf{B}^T \lambda = \mathbf{A} \mathbf{u} \quad \text{Equation of motion}
\]

\[
\Phi(\mathbf{H}) = 0 \quad \text{Kinematic constraints}
\]

\[
\mathbf{y}_{\text{eff}}(\mathbf{H}) - \mathbf{y}_{\text{presc}}(t) = 0 \quad \text{“Servo” constraints}
\]
Method

Use of the direct transcription method:
• **Discretization** of all states into “s” time steps (of size “h”).
• Optimization variables at each time steps “k” (k = 1, ..., s).
• Optimum after “n” iterations.
Method

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

- **Objective function** $J$:

$$
\min_{H^1, \ldots, H^s} J = \min_{H^1, \ldots, H^s} \frac{1}{2T} \sum_{k=1}^{s} [\epsilon^{k,T}(H^k)K\epsilon^k(H^k) + u^{k,T}Gu^k]\eta
$$

- **Constraints of the NLP**:

\[
\begin{align*}
\dot{H}^k_I - H^k_I \tilde{v}_I^k &= 0 \\
M^k \ddot{v}^k + g(H^k, v^k) + B^k,T \lambda^k - Au^k &= 0 \\
\Phi(H^k) &= 0 \\
y_{eff}(H^k) - y_{presc}(t^k) &= 0
\end{align*}
\]
Method

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

- Standard methods (e.g., Euler-implicit type) are **not acceptable**:
  \[ H_{I}^{k+1} = H_{I}^{k} + h\dot{H}_{I}^{k+1} \notin SE(3) \]

- On \( SE(3) \), exponential mapping can be used:
  \[ H_{I}^{k+1} = H_{I}^{k} \exp_{SE(3)}(\widetilde{\Delta Q}_{I}^{k}) \in SE(3) \]

where \( \Delta Q^{k} = (\Delta Q_{1}^{k}, ..., \Delta Q_{N}^{k}) \) is the **change in configuration between two consecutive times** \( k \) and \( k + 1 \).
Method

• Possible set of optimization variables would be
  \[(H^1, v^1, \dot{v}^1, \lambda^1, u^1, ..., H^s, v^s, \dot{v}^s, \lambda^s, u^s)\]
  with \(H \in SE(3)\), which can not be dealt with by classical NLP solvers.

• Alternative: use of a new vector \(\Delta q = (\Delta q_1, ..., \Delta q_N)\) which is the change in configuration between the initial guess states and the optimized states, through the exponential mapping

\[H^k_{I,n} = H^k_{I,0} \exp_{SE(3)}(\Delta q^k_{I,n})\]

• At the end, the optimization variables are

\[x = (\Delta q^1, v^1, \dot{v}^1, \lambda^1, u^1, ..., \Delta q^s, v^s, \dot{v}^s, \lambda^s, u^s)\]
Method

- **Important remark:**

  Difference between “time” related mapping

  \[ H_{I}^{k+1} = H_{I}^{k} \exp_{SE(3)}(\Delta Q_{I}^{k}) \]
  
  and “iteration” related mapping

  \[ H_{I,n}^{k} = H_{I,0}^{k} \exp_{SE(3)}(\Delta q_{I,n}^{k}) \]
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\textsuperscript{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: **1st unstable pole** at 14 Hz.
Results – Parallel robot

\[ s = 150 \ (h = 0.01s, \ 110k \ var.), \ 40 \ min. \ and \ RMS \ error \ from \ 2\% \ to \ 0.7\% \]

![Graph showing commands before and after optimization](image1)

![Graph showing actual trajectories with both commands](image2)

Commands before and after optimization

Actual trajectories with both commands
Summary

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

• **On going work and perspectives:**
  • Feed-forward solution on \textit{robotic testbed} (adding feedback).
  • Consider compliant joints.
  • Consider contact problems with end-effector.
Thank you for your attention.
Modeling of flexible link robots for end-effector trajectory tracking tasks

Arthur Lismonde\textsuperscript{1}, Valentin Sonneville\textsuperscript{2} and Olivier Brüls\textsuperscript{1}

\textsuperscript{1}University of Liège, Belgium
Department of Aerospace and Mechanical engineering

\textsuperscript{2}University of Maryland, USA
Aerospace engineering