Deformation Modelling in Interreg 4A project IRFO - Intelligent Robots for Handling of Flexible Objects

by Andreas R. Fugl

Ph.D. student @
Maersk McKinney Moller Institute and the The Mads Clausen Institute
University of Southern Denmark

25. November 2011

INTERREG4A
SYDDANMARK-SCHLESWIG-K.E.R.N.

European Union
European Regional
Development Fund
Investing in your future

Outline

Outline

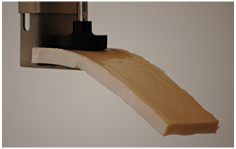
1. The need for Modelling
2. Modelling in context of IRFO system
3. Behaviour of Flexible Objects
4. Accomplished
5. Future plans

The need for Modelling

Rigid: Idealization, neglecting deformation

- The assumption of traditional robotics
- Keeps mathematical models simple.
- No such thing as a rigid object!
- An physical approximation for stiff

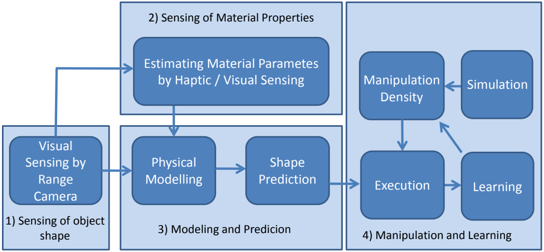
Importance of Modelling



When is something elastic?

- Always! - but rigid may be a very good approximation
- Relevant parameters
  - Scale
  - Forces
  - Length
  - Stiffness
  - Weight

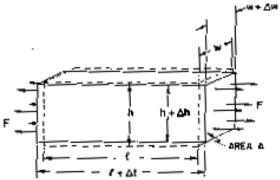
Modelling in the System



Modelling and Prediction

- Receives object shape data from stage 1) and material properties from stage 2)
- Predicts the shape and sends data to the Manipulation/Learning stage

Hooke's Law



Relating stress and strain

- $\frac{F}{A} = E \times \frac{\Delta l}{l}$
- Stress - force per unit area $\frac{F}{A}$
- Young's modulus E
- Strain - stretch per unit length $\frac{\Delta l}{l}$

Elasticity - A light introduction

What's going to happen

- We're making Hooke's Law $Stress = (Youngs\ Modulus) \times (Strain)$ work on a 3D elastic body for all sorts of crazy deformations, then we combine it with Newton's 2nd Law and end up with equations of motion
- In place of the simple stress (force) in Hooke's Law we're going to introduce the stress tensor, and in place of the simple strain (deformation) we're introducing the strain tensor

Elasticity - A light introduction

What's going to happen

- We're making Hooke's Law
  $Stress = (Youngs\ Modulus) \times (Strain)$ work on a 3D elastic body for all sorts of crazy deformations, then we combine it with Newton's 2nd Law and end up with equations of motion
- In place of the simple stress (force) in Hooke's Law we're going to introduce the stress tensor, and in place of the simple strain (deformation) we're introducing the strain tensor

The Stiffness/elasticity tensor

Generalizing Hooke's Law

- *Stresses are proportional to the strains*
- We must linearly relate each component of the stress tensor σ_{kl} to each component of the strain tensor u_{ik}

$$\sigma_{ij} = \sum_{k,l} C_{ijkl} u_{kl}$$

- where C_{ijkl} is the stiffness tensor
- Multi-dimensional analog to spring constant ($F = -kx$)

The Equations of Motion

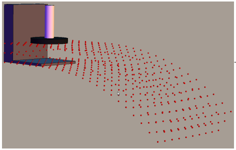
The equations of motion for an elastic material

- We skipped a few steps, where we revisited Newton's 2nd Law and our vector calculus
- For the important special case of a *homogeneous, isotropic* material we end up with

$$\rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} = (\lambda + \mu)\nabla(\nabla \cdot \boldsymbol{u}) + \mu \nabla^2 \boldsymbol{u} \tag{3}$$

- where λ and μ are the Lamé coefficients
- This PDE is the Navier-Cauchy equation

3D numerical solution to Navier-Cauchy PDE

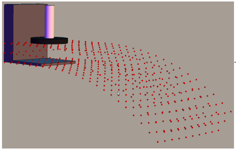


$$
\begin{aligned}
f_x = (\lambda + \mu) \Bigg[& \frac{u_x(i+1,j,k) - 2u_x(i,j,k) + u_x(i-1,j,k)}{{h_x}^2} \\
& + \frac{u_y(i+1,j+1,k) - u_y(i+1,j-1,k) - u_y(i-1,j+1,k) + u_y(i-1,j-1,k)}{4h_x h_y} \\
& + \frac{u_z(i+1,j,k+1) - u_z(i+1,j,k-1) - u_z(i-1,j,k+1) + u_z(i-1,j,k-1)}{4h_x h_z} \Bigg] \\
+ \mu \Bigg[& \frac{u_x(i+1,j,k) - 2u_x(i,j,k) + u_x(i-1,j,k)}{{h_x}^2} \\
& + \frac{u_x(i,j+1,k) - 2u_x(i,j,k) + u_x(i,j-1,k)}{{h_y}^2} \\
& + \frac{u_x(i,j,k+1) - 2u_x(i,j,k) + u_x(i,j,k-1)}{{h_z}^2} \Bigg]
\end{aligned}
$$

Finite Difference Simulation software developed

- *An Outline for an intelligent System performing Peg-in-Hole Actions with flexible Objects, ICIRA 2011*
- + Supports relevant robotics grasping situations
- + Full 3D deformation output
- - Computationally expensive (seconds per frame), but large room for optimization, GPUs, etc.
- -Domain problems!

3D numerical solution to Navier-Cauchy PDE

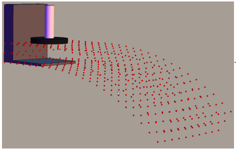


$$
\begin{aligned}
f_x = (\lambda+\mu)\Bigg[&\frac{u_x(i+1,j,k)-2u_x(i,j,k)+u_x(i-1,j,k)}{{h_x}^2} \\
&+\frac{u_y(i+1,j+1,k)-u_y(i+1,j-1,k)-u_y(i-1,j+1,k)+u_y(i-1,j-1,k)}{4h_xh_y} \\
&+\frac{u_z(i+1,j,k+1)-u_z(i+1,j,k-1)-u_z(i-1,j,k+1)+u_z(i-1,j,k-1)}{4h_xh_z}\Bigg] \\
+\mu\Bigg[&\frac{u_x(i+1,j,k)-2u_x(i,j,k)+u_x(i-1,j,k)}{{h_x}^2} \\
&+\frac{u_x(i,j+1,k)-2u_x(i,j,k)+u_x(i,j-1,k)}{{h_y}^2} \\
&+\frac{u_x(i,j,k+1)-2u_x(i,j,k)+u_x(i,j,k-1)}{{h_z}^2}\Bigg]
\end{aligned}
$$

Finite Difference Simulation software developed

- *An Outline for an intelligent System performing Peg-in-Hole Actions with flexible Objects, ICIRA 2011*
- \+ Supports relevant robotics grasping situations
- \+ Full 3D deformation output
- \- Computationally expensive (seconds per frame), but large room for optimization, GPUs, etc.
- -Domain problems!

3D numerical solution to Navier-Cauchy PDE

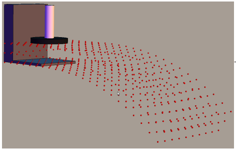


$$
\begin{aligned}
f_x = (\lambda+\mu)\Bigg[&\frac{u_x(i+1,j,k)-2u_x(i,j,k)+u_x(i-1,j,k)}{{h_x}^2}\\
&+\frac{u_y(i+1,j+1,k)-u_y(i+1,j-1,k)-u_y(i-1,j+1,k)+u_y(i-1,j-1,k)}{4h_xh_y}\\
&+\frac{u_z(i+1,j,k+1)-u_z(i+1,j,k-1)-u_z(i-1,j,k+1)+u_z(i-1,j,k-1)}{4h_xh_z}\Bigg]\\
&+\mu\Bigg[\frac{u_x(i+1,j,k)-2u_x(i,j,k)+u_x(i-1,j,k)}{{h_x}^2}\\
&+\frac{u_x(i,j+1,k)-2u_x(i,j,k)+u_x(i,j-1,k)}{{h_y}^2}\\
&+\frac{u_x(i,j,k+1)-2u_x(i,j,k)+u_x(i,j,k-1)}{{h_z}^2}\Bigg]
\end{aligned}
$$

Finite Difference Simulation software developed

- *An Outline for an intelligent System performing Peg-in-Hole Actions with flexible Objects, ICIRA 2011*
- \+ Supports relevant robotics grasping situations
- \+ Full 3D deformation output
- \- Computationally expensive (seconds per frame), but large room for optimization, GPUs, etc.
- -Domain problems!

3D numerical instability

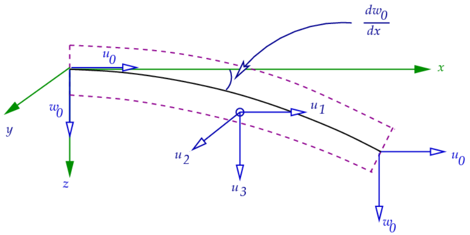


$$f_x = (\lambda + \mu)\left[\frac{u_x(i+1,j,k) - 2u_x(i,j,k) + u_x(i-1,j,k)}{{h_x}^2}\right.$$
$$+\frac{u_y(i+1,j+1,k) - u_y(i+1,j-1,k) - u_y(i-1,j+1,k) + u_y(i-1,j-1,k)}{4h_x h_y}$$
$$\left.+\frac{u_z(i+1,j,k+1) - u_z(i+1,j,k-1) - u_z(i-1,j,k+1) + u_z(i-1,j,k-1)}{4h_x h_z}\right]$$
$$+\mu\left[\frac{u_x(i+1,j,k) - 2u_x(i,j,k) + u_x(i-1,j,k)}{{h_x}^2}\right.$$
$$+\frac{u_x(i,j+1,k) - 2u_x(i,j,k) + u_x(i,j-1,k)}{{h_y}^2}$$
$$\left.+\frac{u_x(i,j,k+1) - 2u_x(i,j,k) + u_x(i,j,k-1)}{{h_z}^2}\right]$$

Domain

- Experiences: Soft materials + thin object = large strains/rotations: Numerical instability
- Stable up to deformation angles of around 30^o - Not sufficient for IRFO!
- Root cause: Linearizing simplification in theory (strain tensor)

Beam models



Beam models

- Bending deformation only
- Stretching in x-direction neglected
- Cross-section info (y and z) through second moment of inertia I.

Linear beam model

$$EI\,\frac{d^4w}{dx^4} = q(x) \tag{4}$$

Small deformation

- *Learning Peg-In-Hole Actions with flexible Objects, SSIR 2012*
- Same linear strain tensor as Navier-Cauchy PDE, but has analytical solutions which are inherently stable
- Geometric post-processing can compensate for these errors.

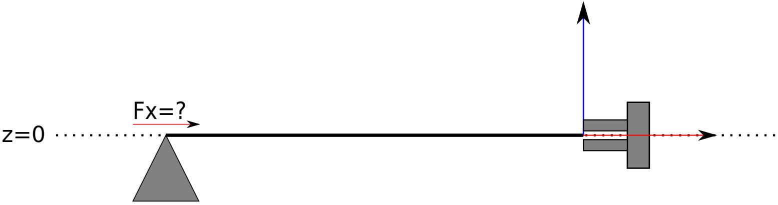
Non-linear beam model

$$EI\,\frac{d^4w}{dx^4} - \frac{3}{2}\,EA\left(\frac{dw}{dx}\right)^2\left(\frac{d^2w}{dx^2}\right) = q(x) \qquad (5)$$

Large-deformation

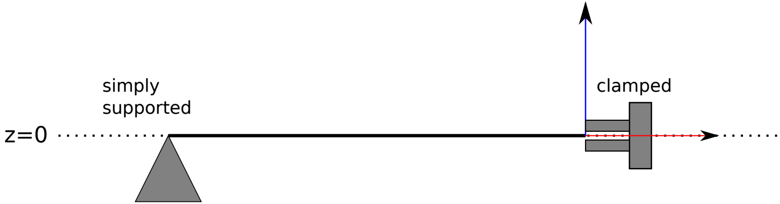
- Here shown for constant E, I and A
- Extra non-linear term (second order effects included)
- Numerical solution methods the only viable option

Laying Down



- The problem:
  - Laying down/Placement operation
  - Determine the unknown elastic force $F_x(x = 0, t)$ due to bending, at the table contact point.
  - Point is fixed due to static friction, but for some critical force F_{crit} it will begin to slide.

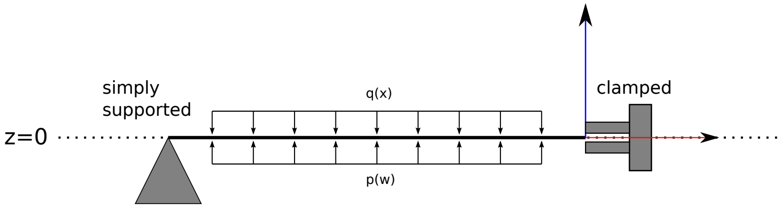
Laying Down



- Assumptions:
  - End at $x = 0$ is fixed on the table, but free to rotate
  - End at $x = L$ is grasped by the robot gripper, with a known position and rotation

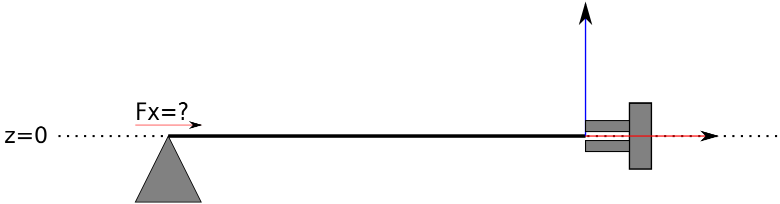
- Boundary conditions:
  - x=0: Simply supported, $w(0, t) = 0$, $w''(0, t) = 0$
  - x=L: Clamped: $w(L, t) = f(t)$, $w'(L, t) = g(t)$

Laying Down



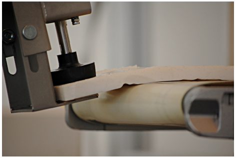
- Penalize table contact ($w < 0$):
  - $p(w) = F_p H(0 > w > -\varepsilon)$
  - $H(w)$ is the Heaviside step function
  - F_p is the penalty force
- Include as an extra source term:
  - E.g. $EI\frac{d^4w}{dx^4} + \ldots = q(x) + p(w)$
  - If not already, equation will become non-linear

Laying Down



- Having solved for the deformation $w(x, t)$ by the constitutive laws of elasticity, strain and geometric relations we obtain $F_x(x = 0, t)$.
- Should $F_x(x = 0, t)$ exceed some critical limit F_{crit}, the object will overcome static friction and begin to slide

Summary



- Grasping
  - *An Outline for an intelligent System performing Peg-in-Hole Actions with flexible Objects, ICIRA 2011*
- Peg-in-Hole
  - *Learning Peg-In-Hole Actions with flexible Objects, SSIR 2012*
- Placement/Laying down operations (Modelling and Learning)
  - Ongoing work, as presented
- Vision integration and estimation of parameters
  - Ongoing work

Thanks for your attention