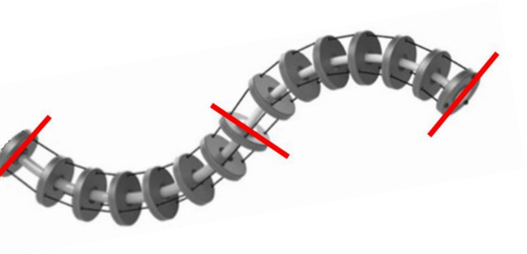
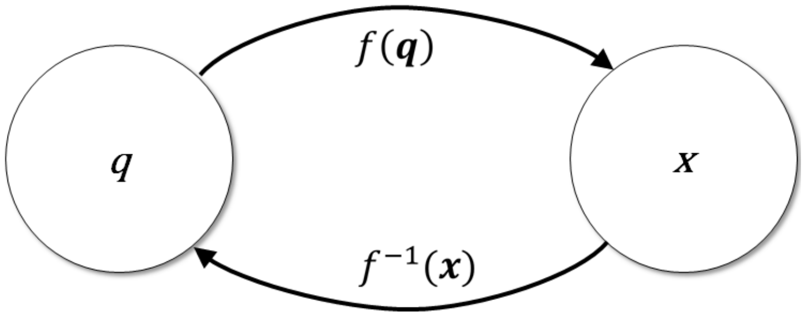
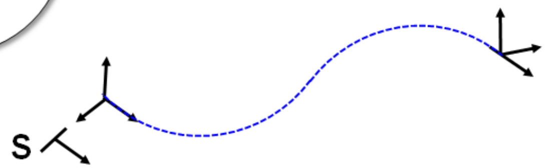
Laboratory for Continuum Robotics

Leibniz Universität Hannover







Geometric Modeling for a Tendon Actuated CR Prototype

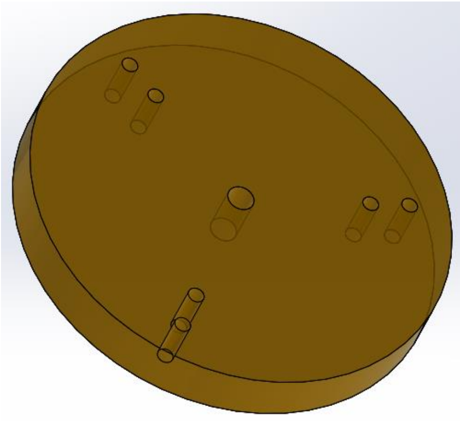
TUTORIAL
CONTINUUM ROBOTICS

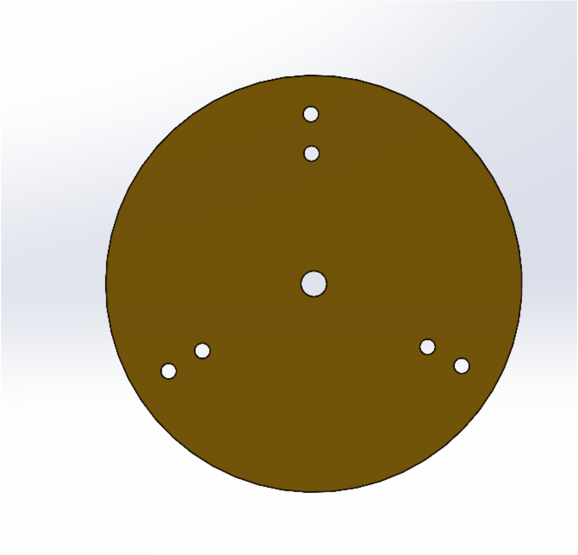
Dipl.-Ing.
Josephine Granna

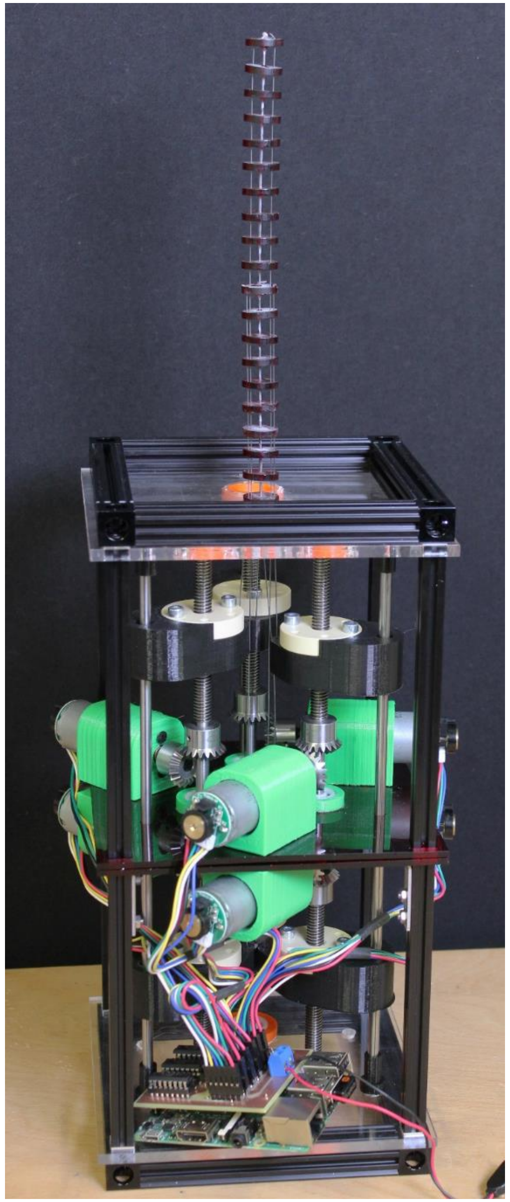
November 24, 2016

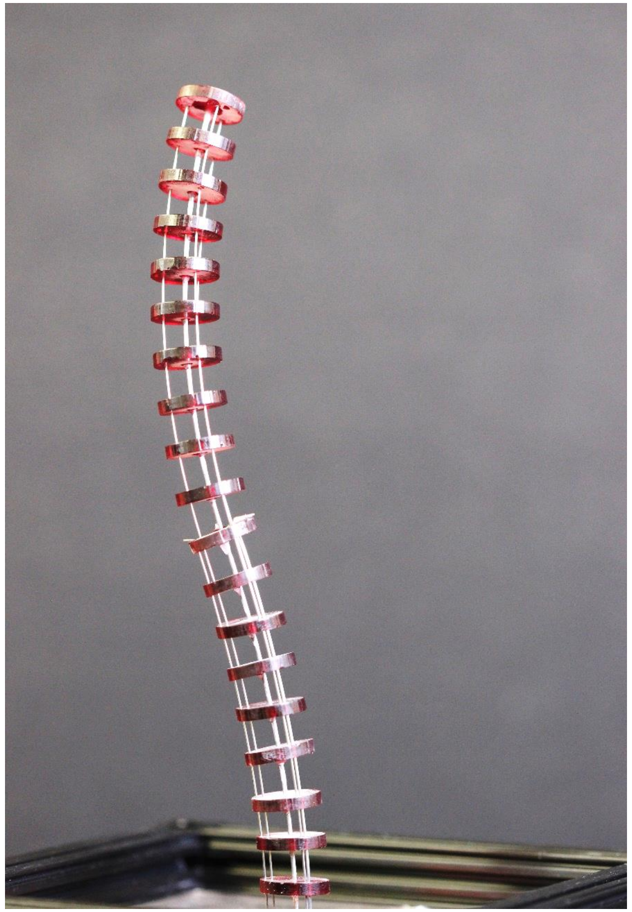
Prototype

- 2 segments
- 3 tendons per segment
- 6 motors to control each tendon





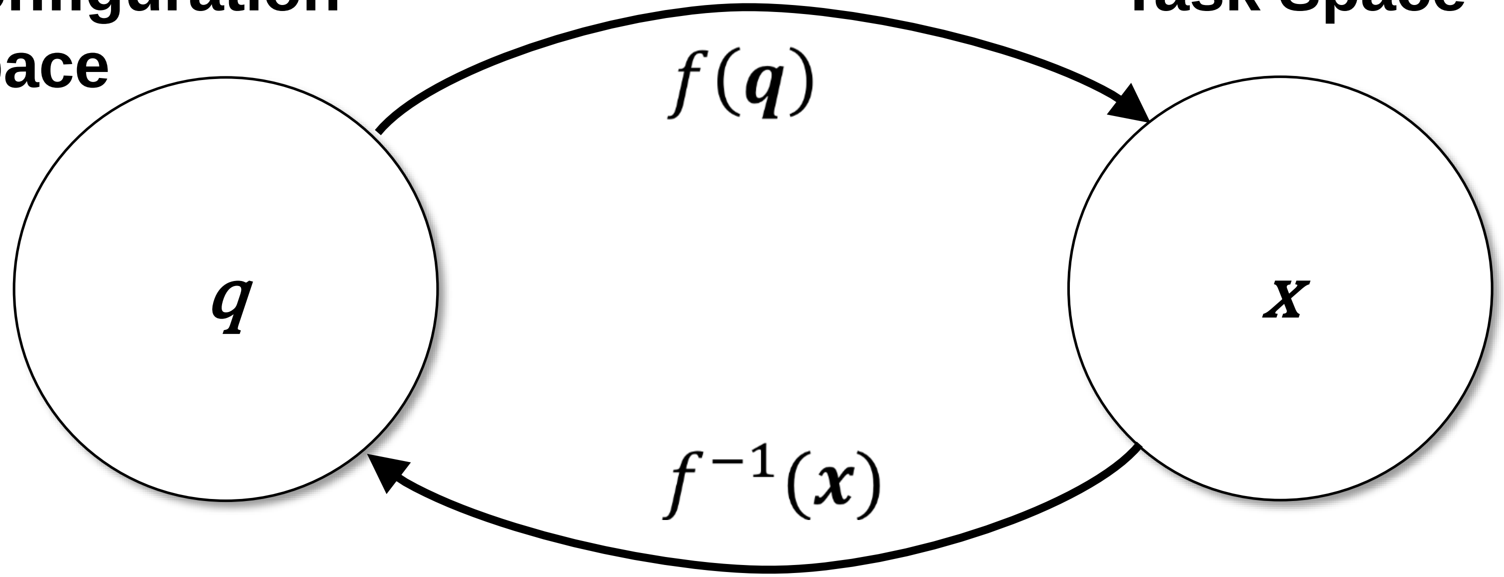




Direct and Inverse Kinematics for a Tendon Actuated Continuum Robot

Configuration Space

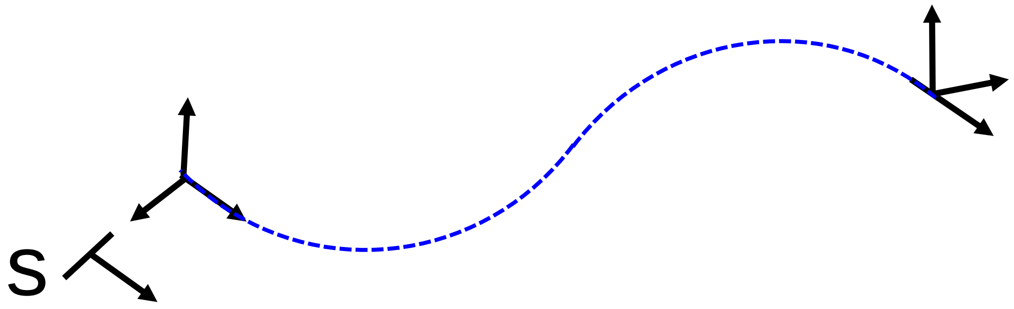
- Length of tendons/rods



Task Space

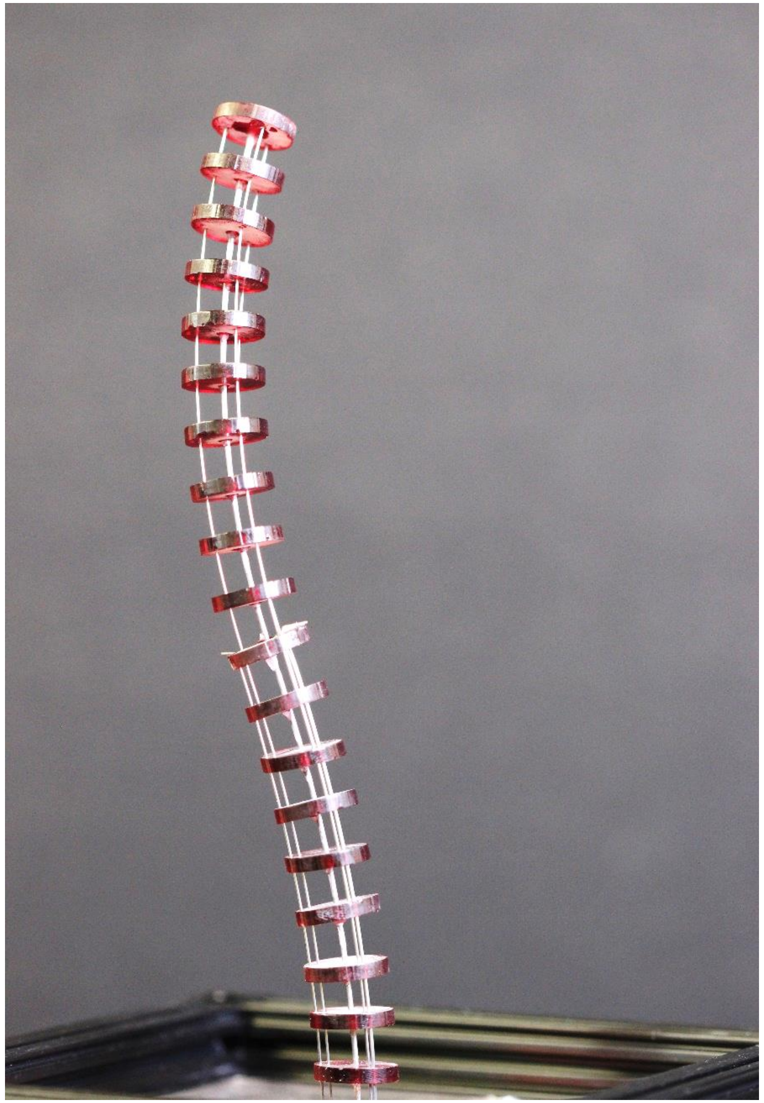
- Position
- Orientation
- Space Curve

$$q = [\Delta\ell_{11}, \Delta\ell_{12}, \Delta\ell_{13}, \Delta\ell_{21}, \Delta\ell_{22}, \Delta\ell_{23}]$$

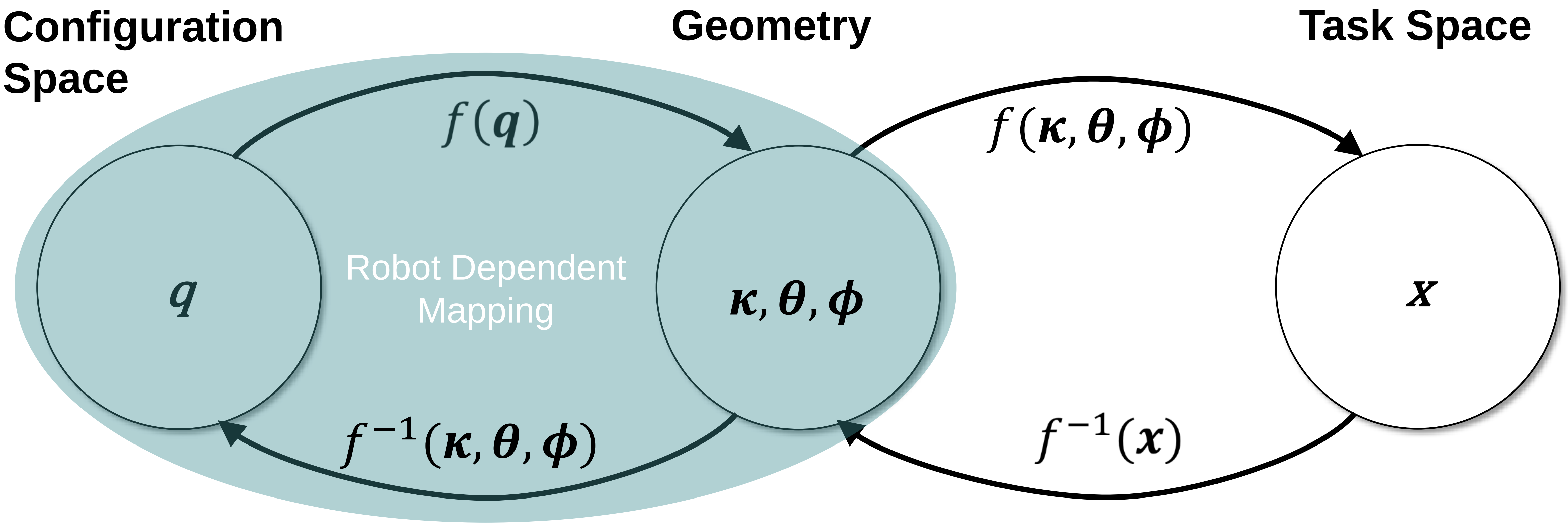


Geometric Modeling Assumptions

- Simplified linear beam deflection
  - Circular bending shape
- Neglected influencing factors
  - Gravitational energy
  - Friction
  - Weight of disks
  - External forces
- Fixed distances
  - between first and secondary backbones
  - between disks



Robot Dependent Mapping

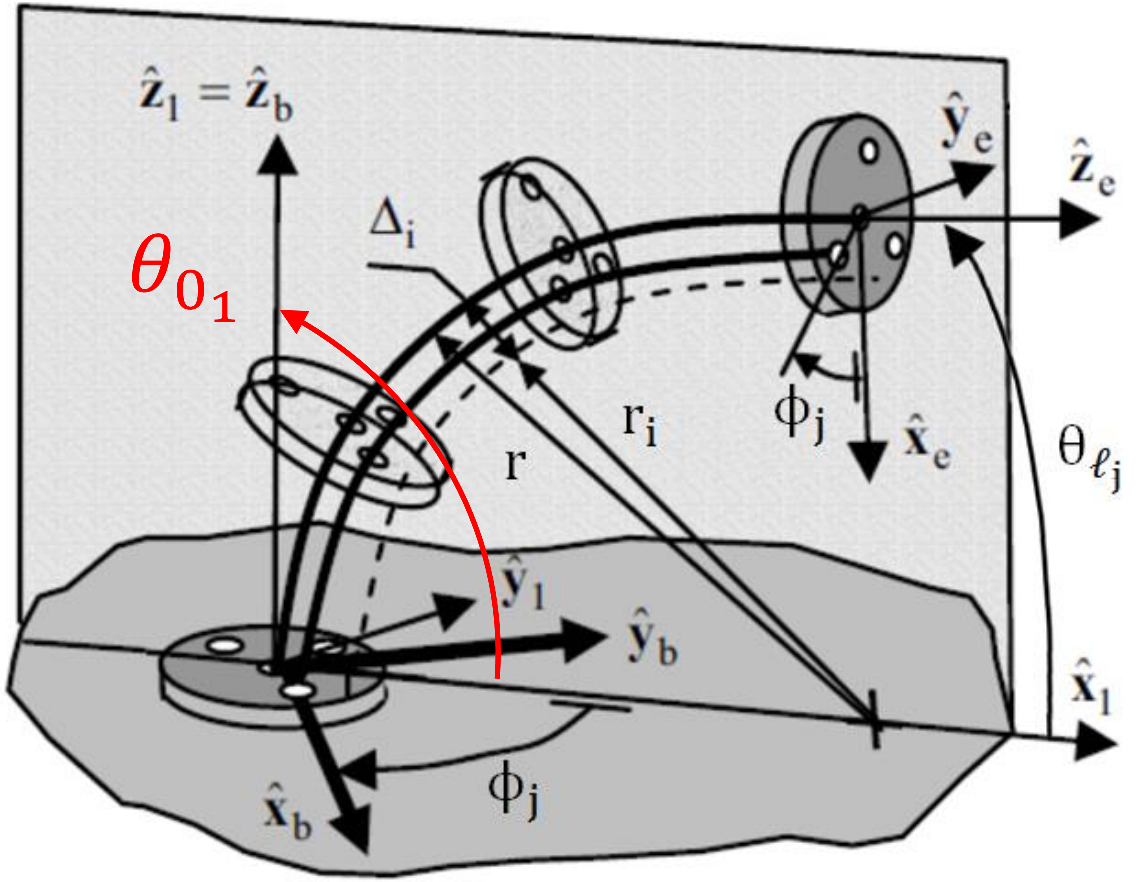


- Determine geometrical parameters first:
  - Curvature κ
  - Rotation and bending angle ϕ, θ

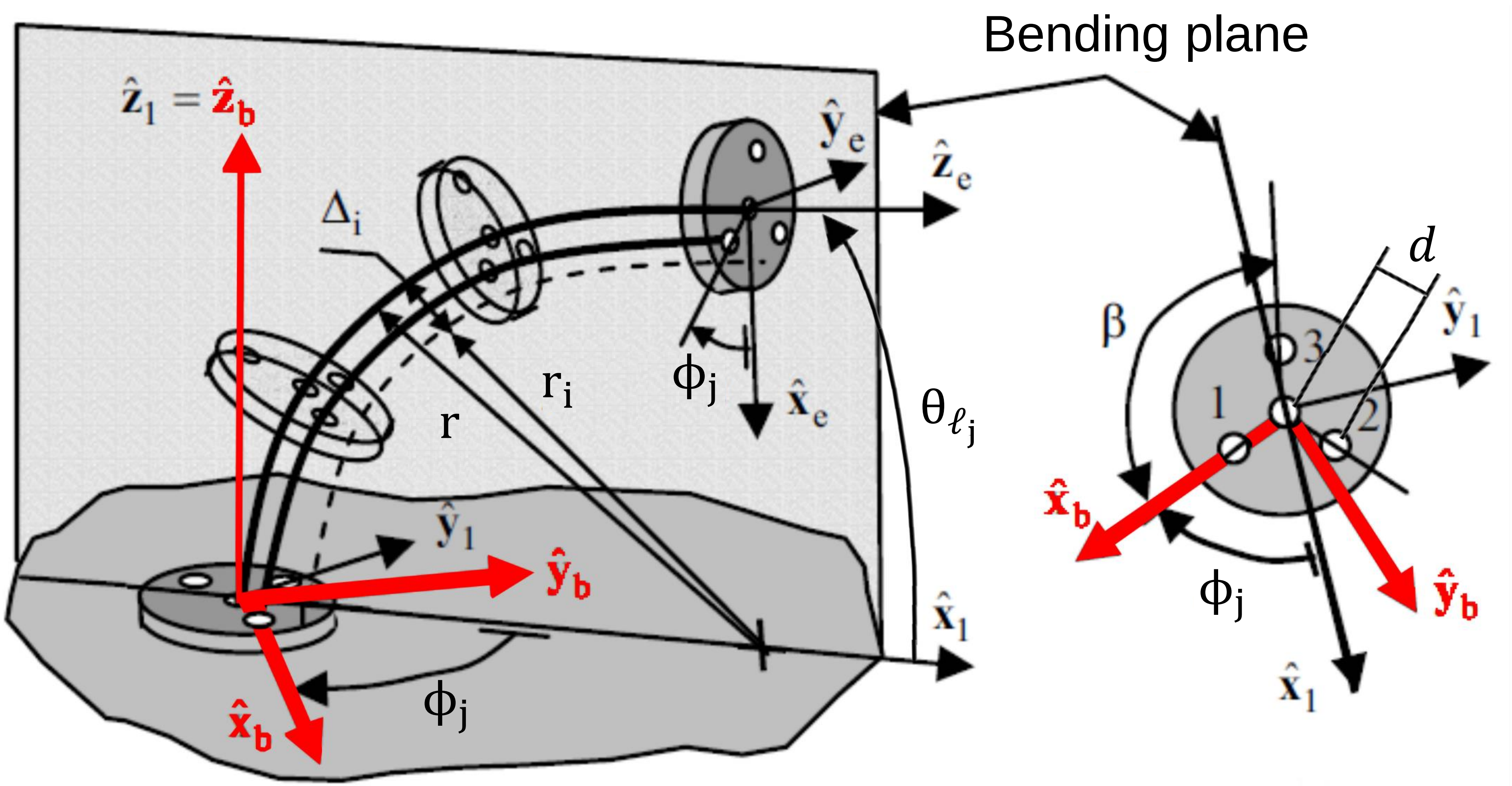
Geometrical Parameters

Single Segment

- Three tendons
- Indices
  - j: segment
  - i: secondary backbone
- Angle of the tangent backbone in the bending plane:
  - $\theta_{0_1} = \theta_{0_2} = \ldots = \pi/2$

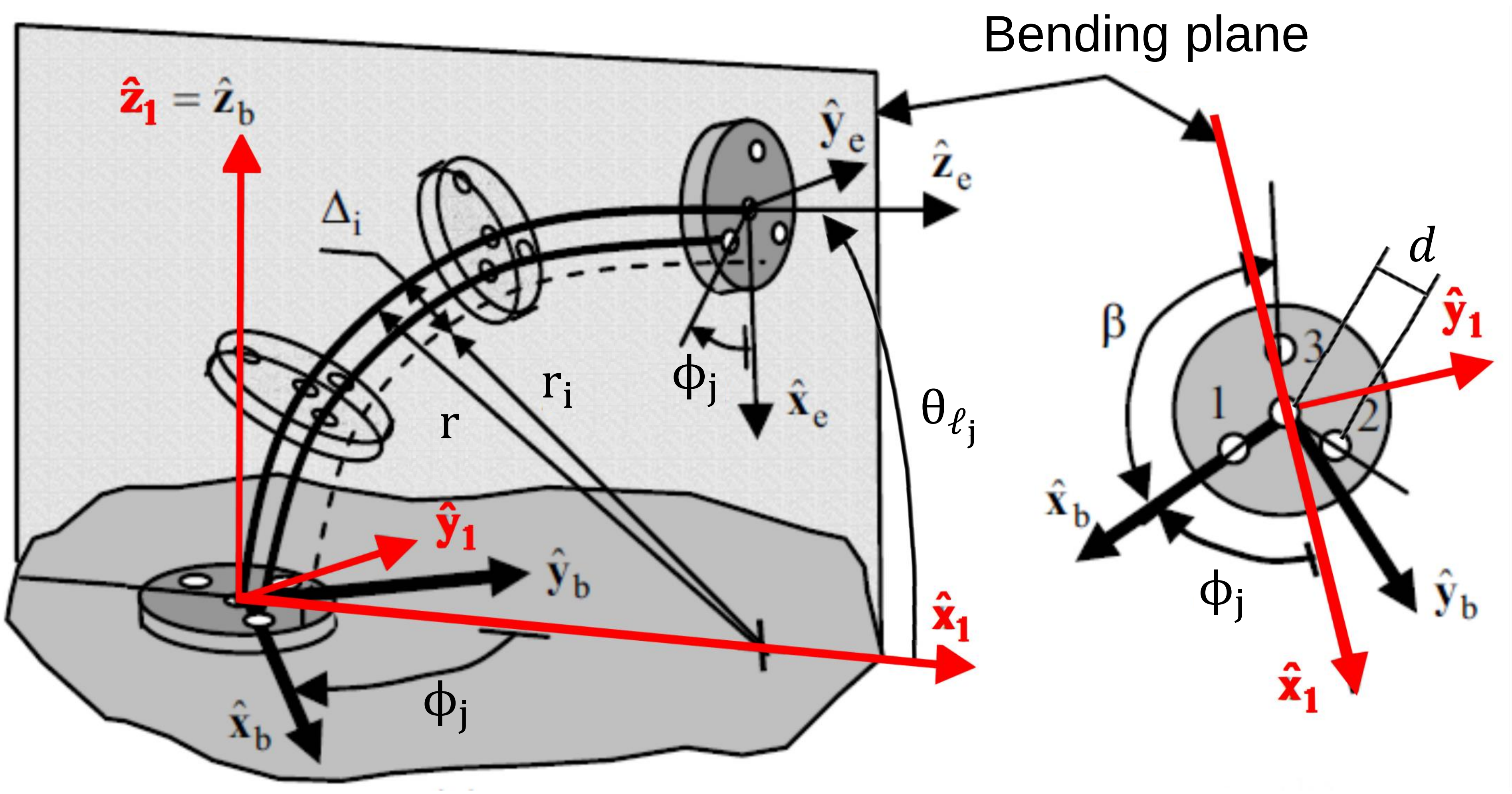


Base Disk Coordinate System

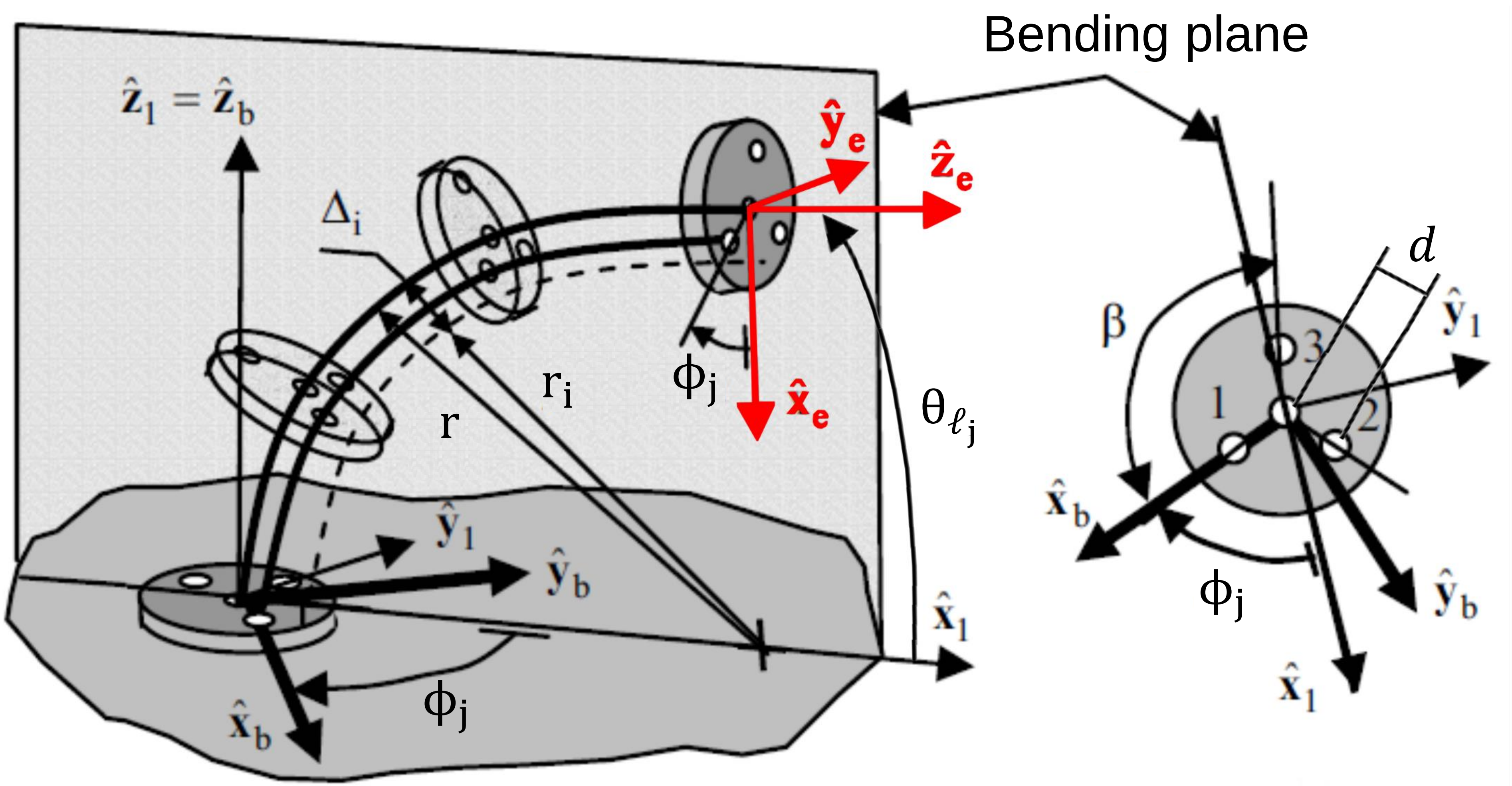


- x_b-axis points from the center to the first secondary backbone

Bending Plane Coordinate System



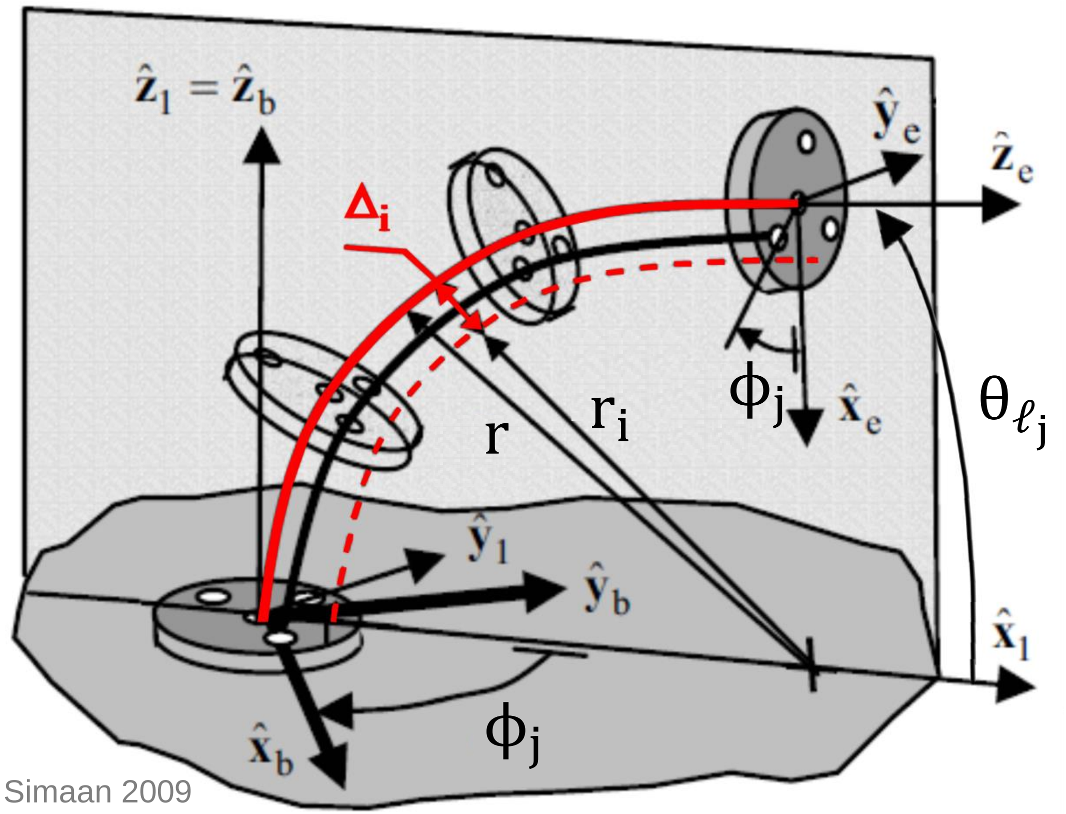
End Disk Coordinate System



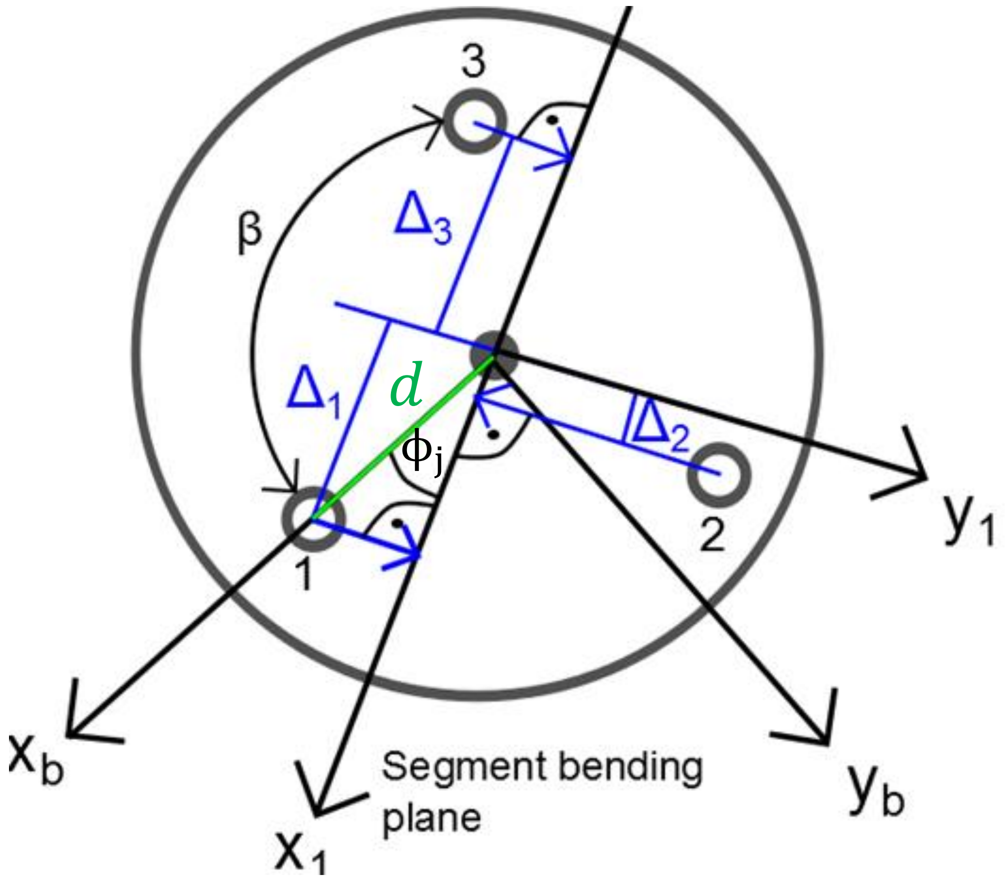
- Position and orientation of end disk coordinate system?

Secondary Backbones

- Secondary backbone is offset by Δ_i from the primary backbone



Simaan 2009



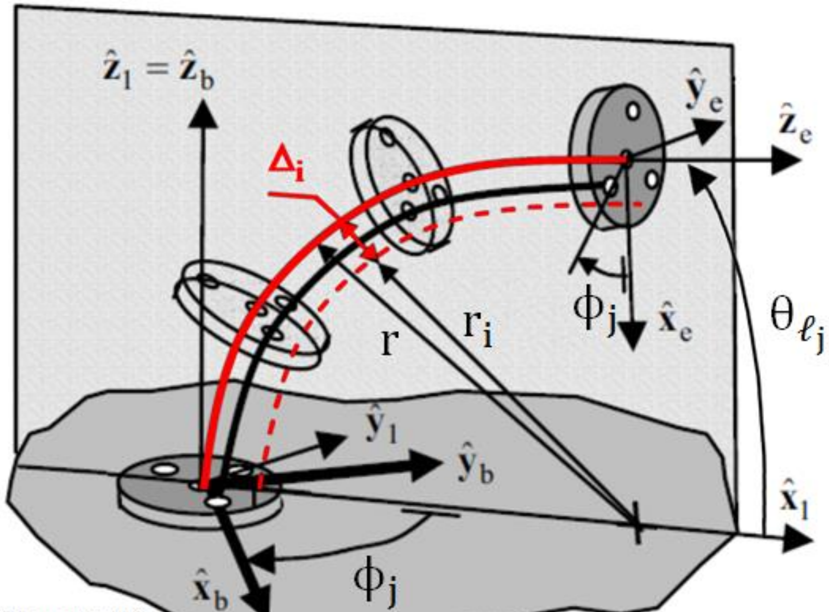
Offset Δ_i

- Secondary backbone is offset by Δ_i from the primary backbone

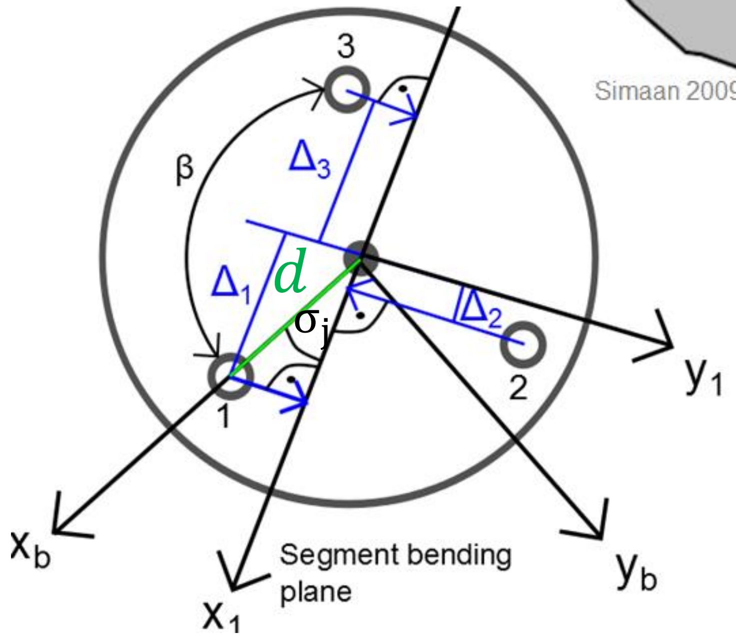
$$\Delta_i = d \cdot \cos(\sigma_j) \qquad \sigma_1 = \phi_j$$

$$\Delta_i = d \cdot \cos(\phi_j + (i-1)\,\beta)$$

$$\beta = \frac{2\pi}{3}$$



Simaan 2009

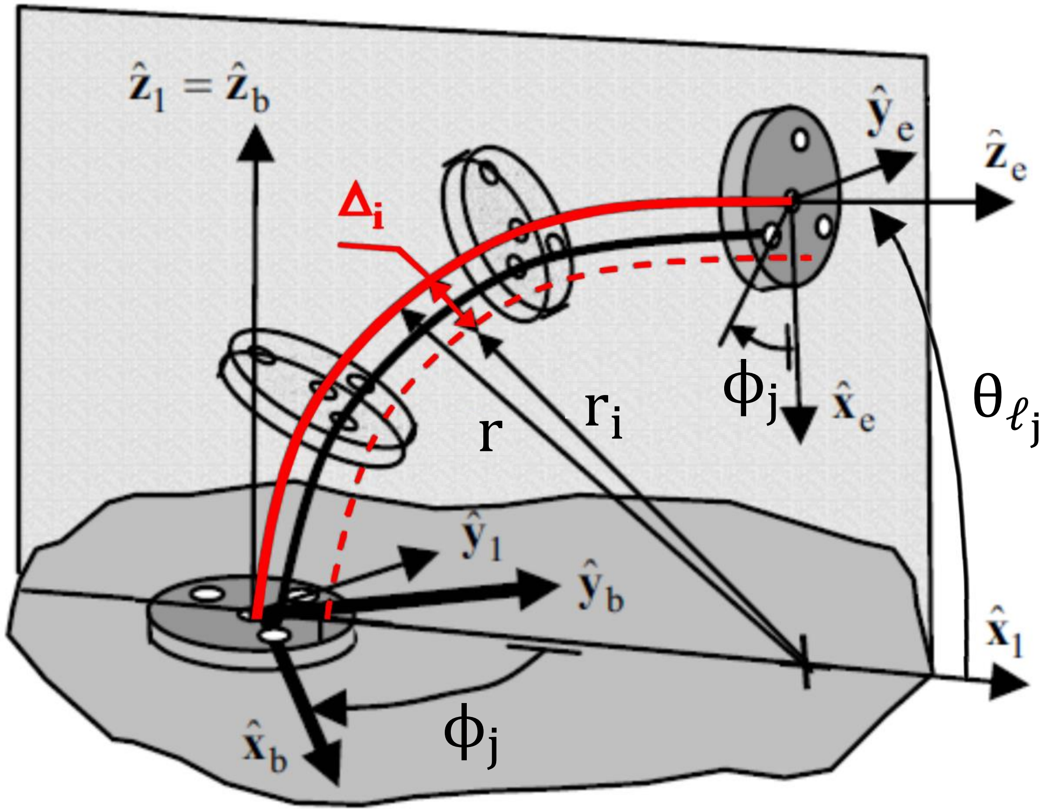


Radius of Curvature

- Radius of curvature of primary backbone and secondary backbine are related according to:

$$r = r_i + \Delta_i$$

$$\frac{1}{\kappa} = \frac{1}{\kappa_i} + \Delta_i$$



Simaan 2009

Bending Angle θ_{ℓ_j}

- s: arc length
- For one segment:

$$\ell_{s_i} = \int ds_i = \int ds_i - ds + ds$$

$$= \int_{\alpha_0}^{\alpha_\ell} (r_i - r)\, d\alpha + \ell$$

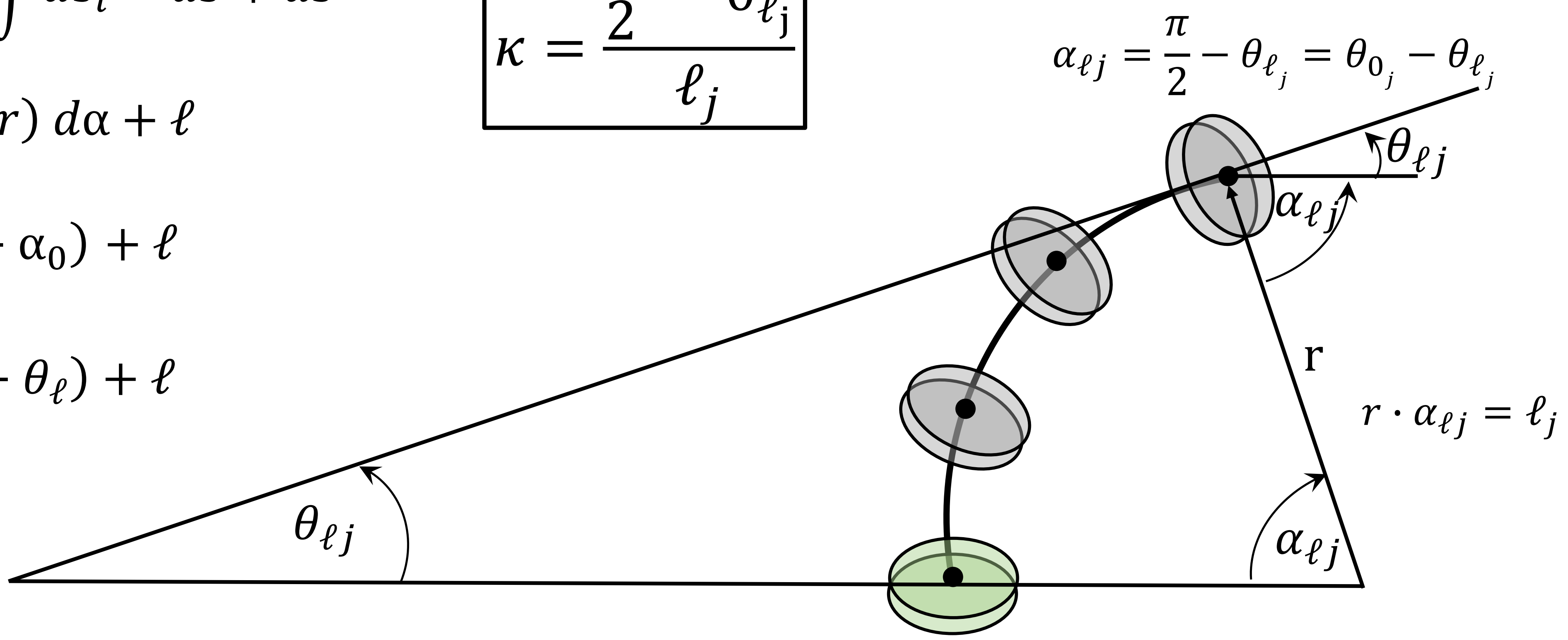
$$= -\Delta_i\, (\alpha_\ell - \alpha_0) + \ell$$

$$= -\Delta_i\, (\theta_0 - \theta_\ell) + \ell$$

$\rightarrow \theta_\ell = \theta_0 + \frac{\ell_{s_i} - \ell}{\Delta_i}$

$$\boxed{\theta_{\ell_j} = \theta_{0_j} + \frac{\Delta\ell_{ji}}{\Delta_{ji}}}$$

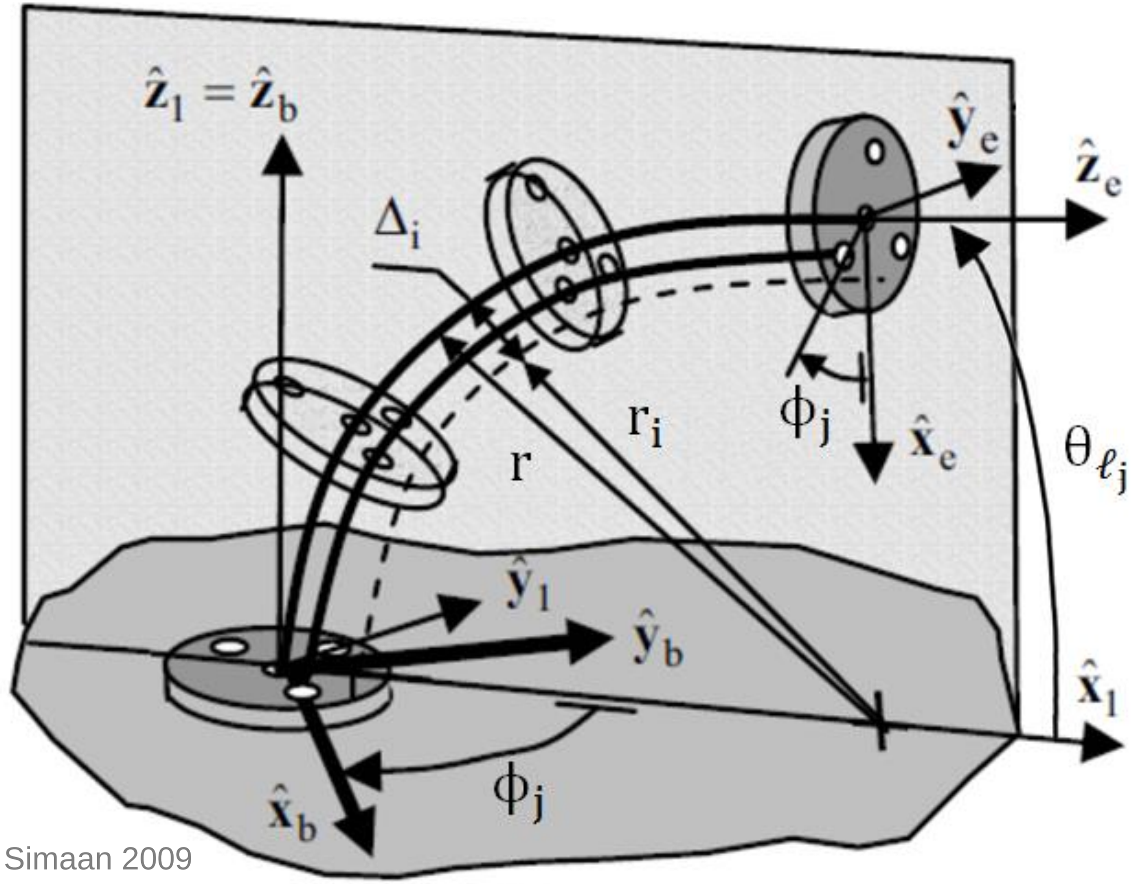
$$\boxed{\kappa = \frac{\frac{\pi}{2} - \theta_{\ell_j}}{\ell_j}}$$



Kinematic Compatibility Condition

- θ_{ℓ_j} is identical, if e.g. tendon 1 and 2 are being actuated

$$\theta_{0_j} + \frac{\Delta\ell_{11}}{\Delta 1} = \theta_{0_j} + \frac{\Delta\ell_{12}}{\Delta 2}$$



Simaan 2009

Exercise

- Use the kinematic compatibility condition to derive an equation for the rotation angle ϕ

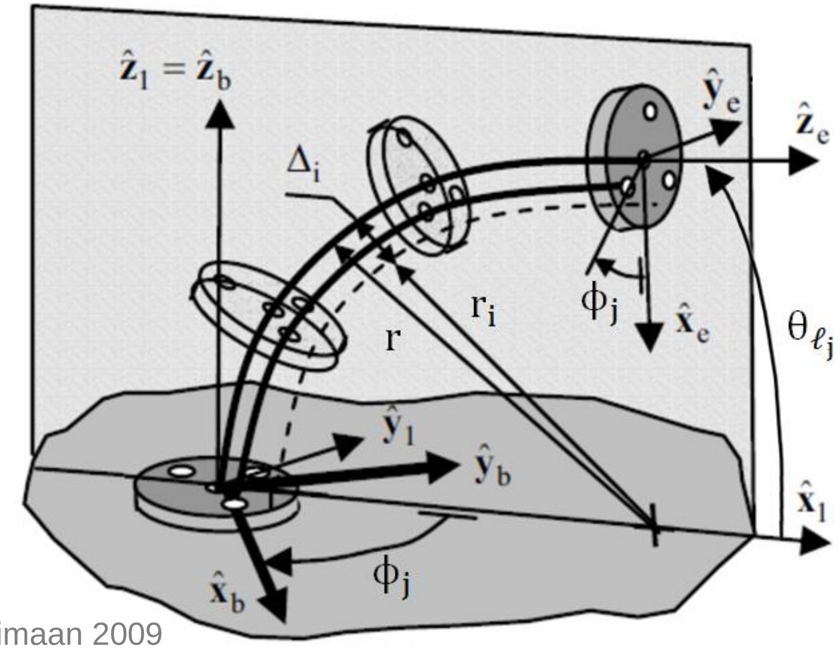
Bending angle $\theta_{\ell_j} = \theta_{0_j} + \frac{\Delta\ell_{ji}}{\Delta_i}$

Kinematic compatibility condition $\theta_{0_j} + \frac{\Delta\ell_{11}}{\Delta 1} = \theta_{0_j} + \frac{\Delta\ell_{12}}{\Delta 2}$

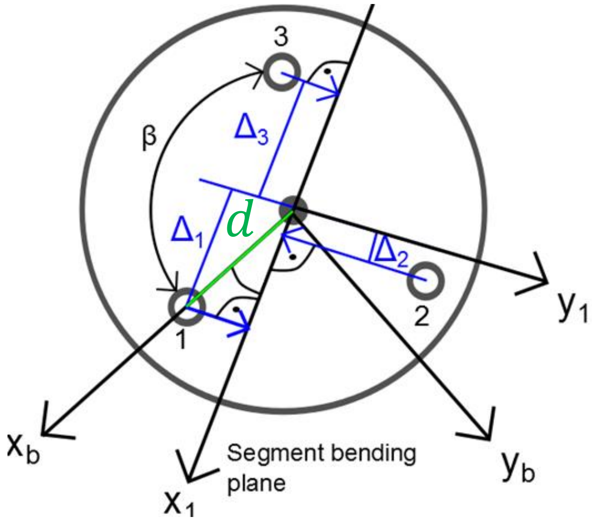
Offset $\Delta_i = d \cdot \cos(\phi_j + (i-1)\,\beta)$

$\cos(x+y)$
$= \cos(x)\cos(y)$
$- \sin(x)\sin(y)$

$\phi = ?$



Simaan 2009



Rotation Angle ϕ_j

Kinematic compatibility condition $\theta_{0_1} + \frac{\Delta\ell_{11}}{\Delta 1} = \theta_{0_1} + \frac{\Delta\ell_{12}}{\Delta 2}$

Offset $\Delta_i = d \cdot \cos(\phi_j + (i-1)\,\beta)$

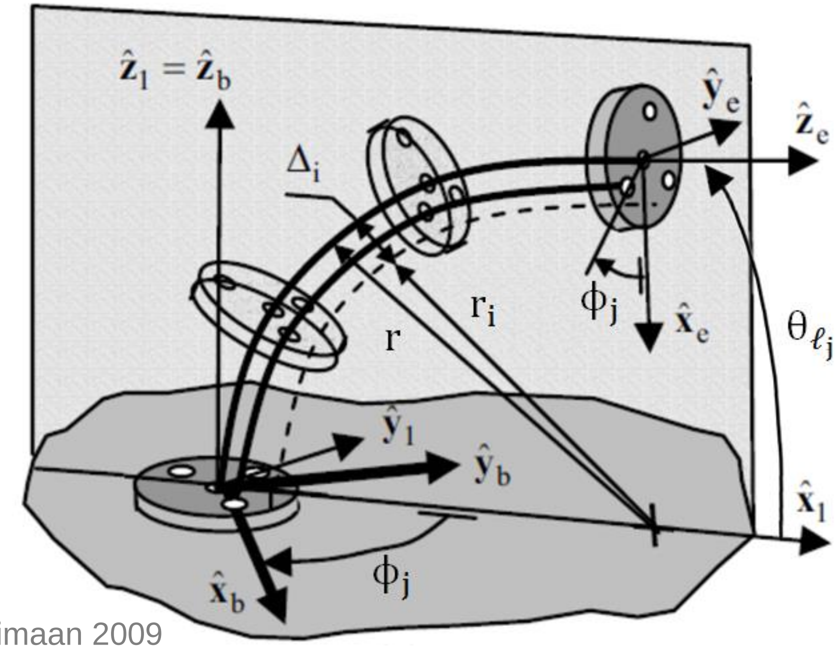
$$\theta_{0_j} + \frac{\Delta\ell_{j1}}{\Delta 1} = \theta_{0_j} + \frac{\Delta\ell_{j2}}{\Delta 2}$$

$$\frac{\Delta\ell_{j1}}{d\cos(\phi_1)} = \frac{\Delta\ell_{j2}}{d\cos(\phi_1+\beta)}$$

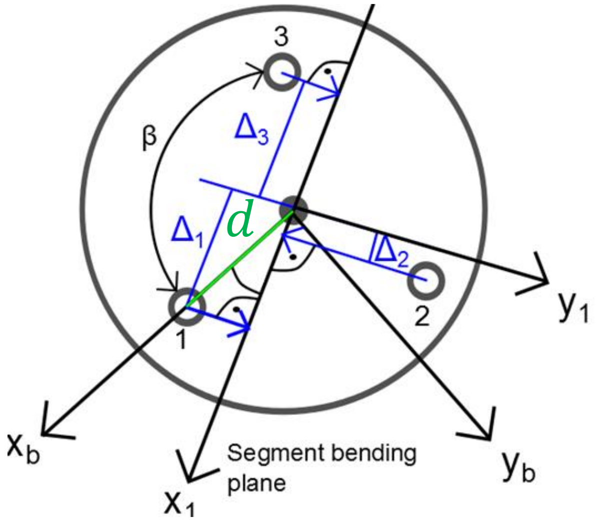
$$\frac{\Delta\ell_{j1}}{\cos(\phi_1)} = \frac{\Delta\ell_{j2}}{\cos(\phi_1)\cos(\beta)-\sin(\phi_1)\sin(\beta)}$$

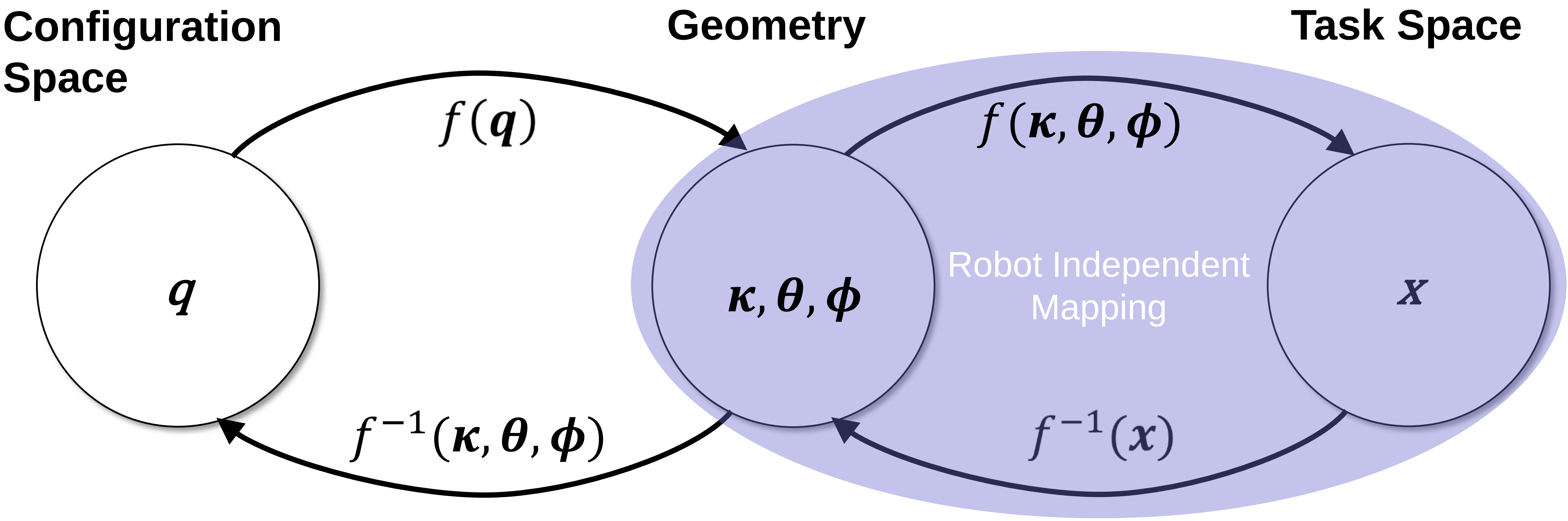
$$\left(\Delta\ell_{j1}\cos(\beta) - \Delta\ell_{j2}\right)\cos(\phi_1) = \Delta\ell_{j1}\sin(\beta)\sin(\phi_1)$$

$$\frac{\sin(\phi_j)}{\cos(\phi_j)} = \frac{\Delta\ell_{j1}\cos(\beta) - \Delta\ell_{j2}}{\Delta\ell_{j1}\sin(\beta)} \qquad \boxed{\phi_j = atan2\,\frac{\Delta\ell_{j1}\cos(\beta) - \Delta\ell_{j2}}{\Delta\ell_{j1}\sin(\beta)}}$$



Simaan 2009



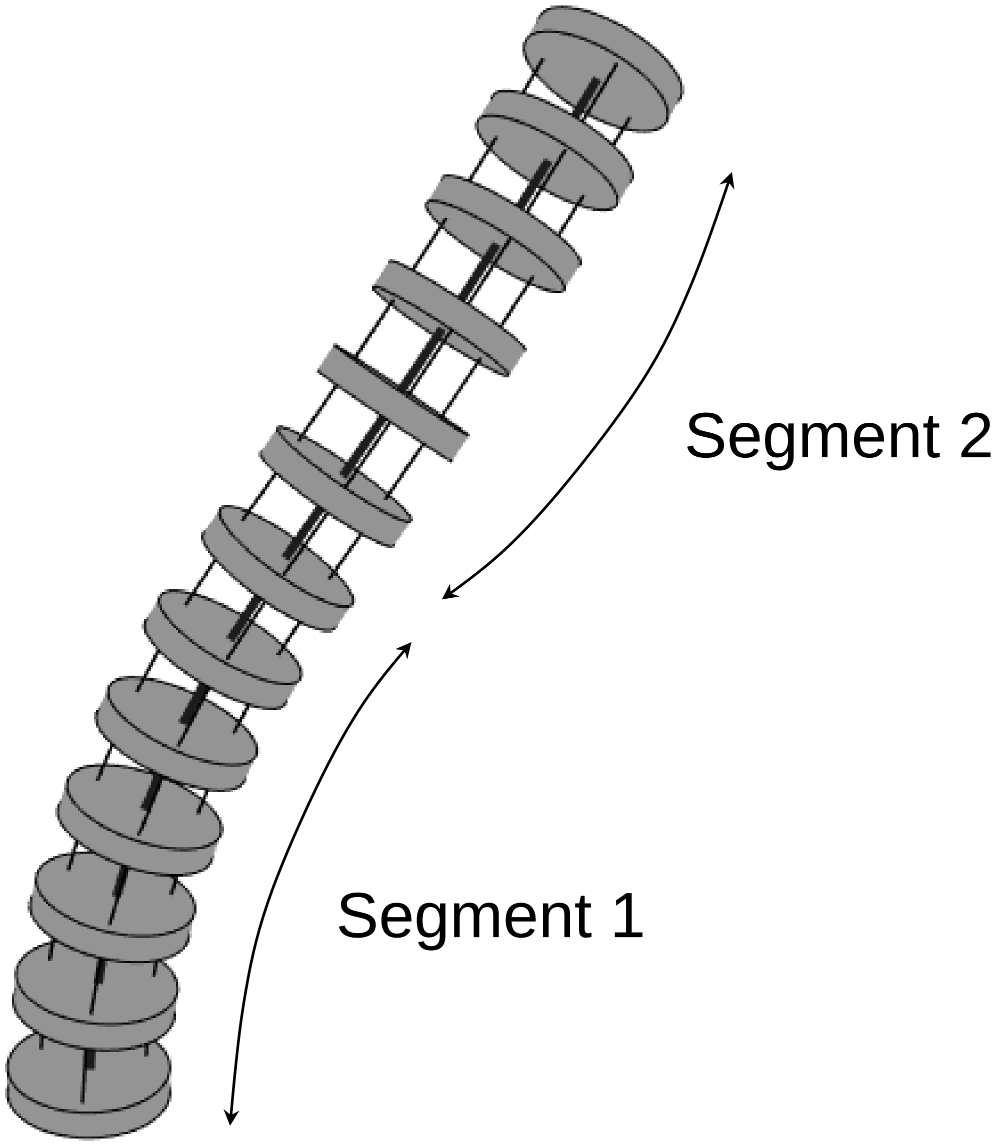


- Geometrical parameters
  - Curvature κ
  - Rotation and bending angle ϕ, θ

- Position
- Orientation
- Space Curve

3D Space Curve

Frames along Backbone



- Generate frames at each disk‘s centerpoint and determine their position and orientation
- Output of your model is space curve
- Robot independent mapping:
  - Transformation matrices with bending and rotation angle
  - Frenet-Serret Frames

Implement the Geometric Model

- runGeometricModeling.m
  - Define robot parameters
  - Run implemented model
  - Input of configurational parameters

- GeometricModel.m
  - Determine position and orientation of robot‘s space curve

- drawRobot.m
  - Visualize robot in 3D space

- Document your code!

runGeometricModeling.m

- Script file

```
% 2 segment tendon actuated continuum robot
TACR.ndisks = [10;10];                    % number of disks per segment
TACR.diskRadius = [8;8];             % disk radius
TACR.diskHeight = 3;                 % heigth of the disks
TACR.diskPitchRadius = [6.5;5];        % pitch circle radius of disks
TACR.segmentLength = [92;102];      % segment length = length of first backbone per segment

q = [4,-2,0;2,2,0];                  % actuation parameters (delta l per tendon);
                                     % Remember: only 2 tendons can be
                                     % retracted at once, the 3rd tendon has
                                     % to extend

% compute robot's space curve
robotShape = GeometricModel(TACR,q);

% visualize the robot
drawRobot(robotShape,TACR);
```

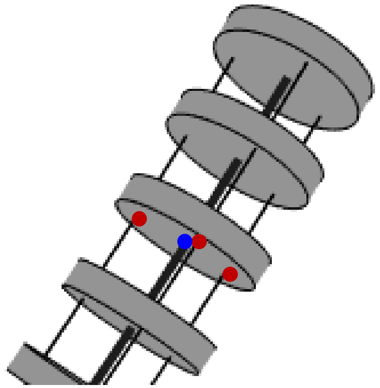
GeometricModel.m

- Function file
- Input: struct TACR, q: [j x 3] configurational parameters
- Output: robotShape

```matlab
function[robotShape] = GeometricModel(TACR,q)
%% GeometricModel.m
% This function computes the space curve of a tendon actuated continuum
% robot with 2 segments and utilizes a geometric forward kinematics model
%
% Copyright: 2016 Leibniz Universität Hannover, All rights reserved
%
% Email: continuumrobotics@lkr.uni-hannover.de
%
% Version: 1
% Date: 11/16/2016
%
% input: struct TACR, q:[jx3] configurational parameters
%
% output: robotShape.diskPoints: n rows for n disks, 12 columns for coordinates
%        (x,y,z) for central backbone, points tendon 1, points tendon 2,
%         points tendon 3
%         robotShape.diskRotation: n rows for n disks, columns represent
%         rotation matrices (3x3): columns 1-3 are matrix elements (1,1; 1,2; 1,3),
%         columns 4-6 are matrix elements (2,1; 2,2; 2,3),
%         columns 7-9 are matrix elements (3,1; 3,2; 3,3),

%% implement the geometric model here


robotShape.diskPoints=[];
robotShape.diskRotation=[];

end
```

drawRobot.m

- Function file
- Input: robotShape
- Output: []



→ Points measured at bottom of disk

robotShape.diskPoints

	Central Backbone			Tendon 1			Tendon 2			Tendon 3		
Disks	x	y	z	x	y	z	x	y	z	x	y	z
1												
2												
3												

3rd row, 3rd column of matrix

robotShape.diskRotation

Disks	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
1									
2									
3									

- Rotation matrices for each backbone point with respect to base frame

Find your team member today

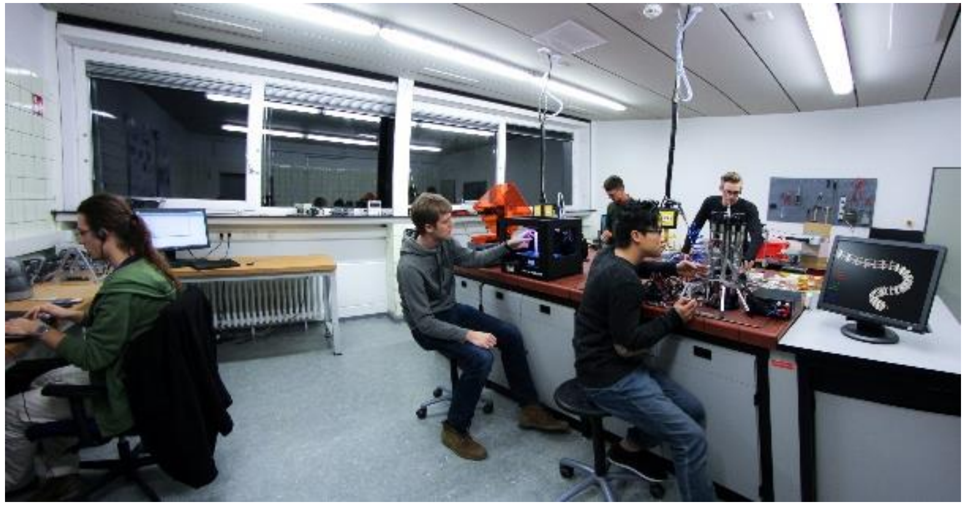
- 2 students per team

- Send Email until 11/25/16 with:
  - your name & matriculation number
  - your team member‘s name & matriculation number

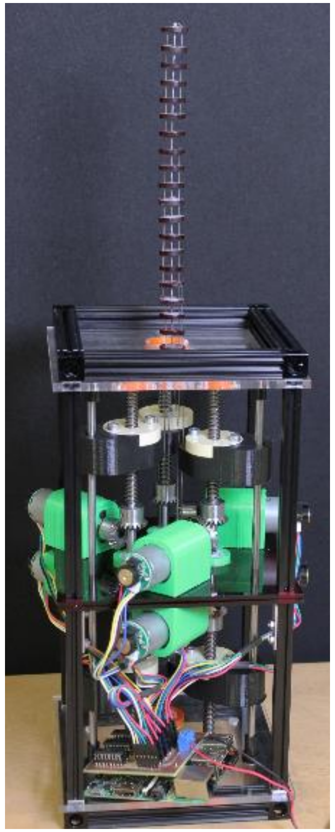
continuumrobotics@lkr.uni-hannover.de

Appointment for Experimental Evaluation

- Make appointment for experimental evaluation in our lab during office hours



© www.3d-microscribe.com



- Office hours: Monday 4pm-5pm, Wednesday 3pm-4pm