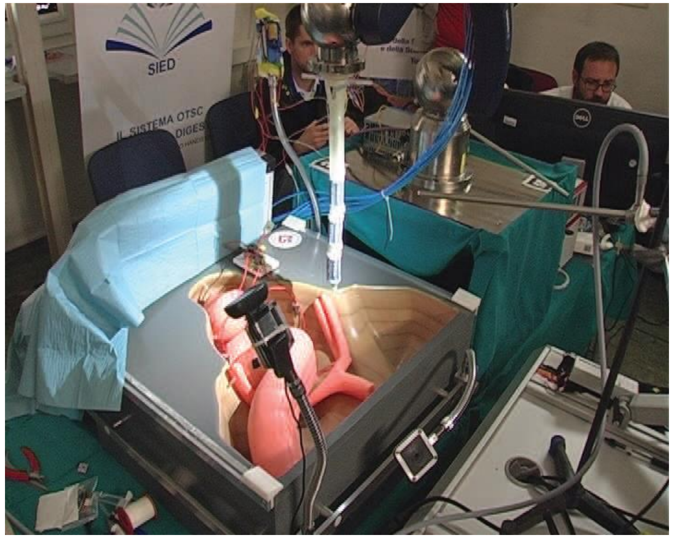
Modeling and Control of Surgical Soft-Continuum Manipulators



Dr. Steeve Mbakop
Eng. Mechanical Design
M.sc Robotics and Automation
PhD Robotics

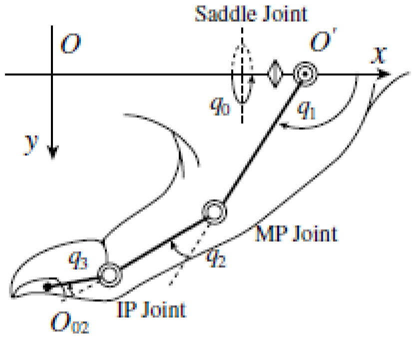
Academic year 2023/ 2024

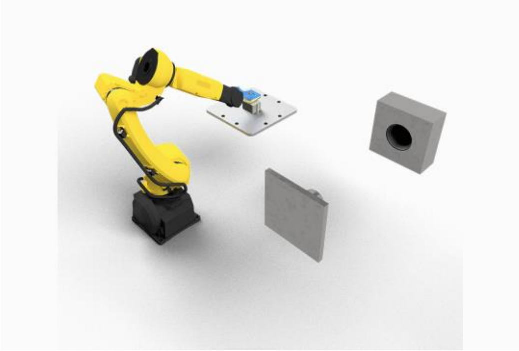
« Diviser chacune des difficultés que j'examinerais, en autant de parcelles qu'il se pourrait et qu'il serait requis pour les mieux résoudre »
Réné Descartes, Discours de la méthode

Soft Robotics State-of-arts

Summary

Robotics and the bio-inspiration



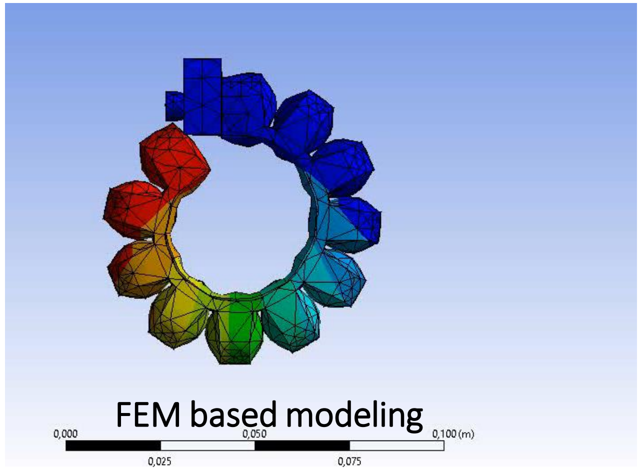


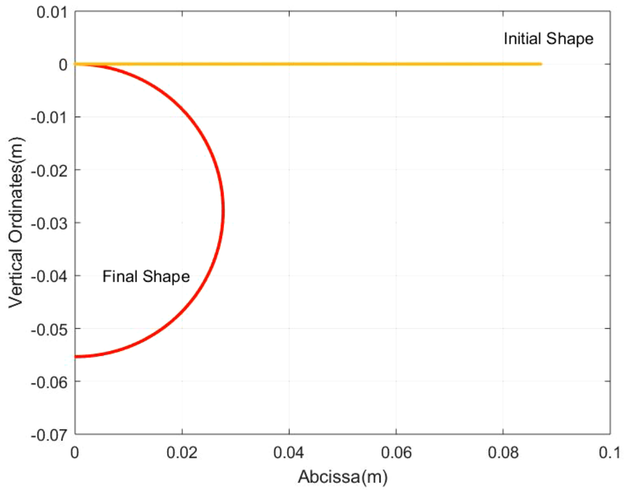
1. Overview

Quantitative models (model-based approaches)

Elastic methods
- Finite Element Methods(**FEM**)
- Cosserat-rod based modeling
- Euler-Bernouilli Kinematics (**EB**)

Geometry based modeling
- Constant Curvature (**CC**) ,
- Piece-wiese Constant Curvature (**PCC**)
- Curve based technique

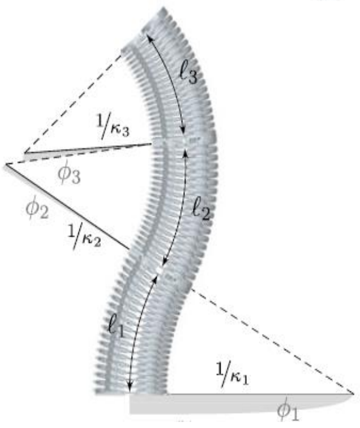

FEM based modeling

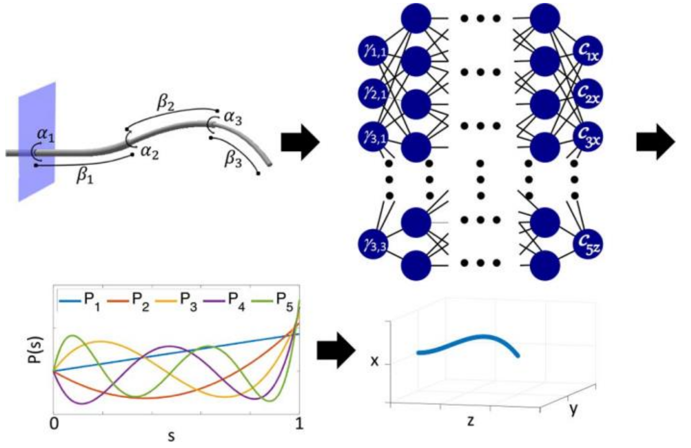


Curve based modeling

Qualitative approaches (Model-free approaches)

AI based modeling
- Neural Network (**NN**),
- Machine Learning (**ML**)
- Deep Learning (**DL**)
- Fuzzy Logic (**FL**)


PCC based modeling


Neural Network (NN) based modeling

CRIStAL Centre de Recherche en Informatique, Signal et Automatique de Lille

centralelille ÉCOLE CENTRALE DE LILLE

Geometry based modeling: PCC approach

Kinematics is about the movement of bodies regardless of the forces/torques that cause the movement.

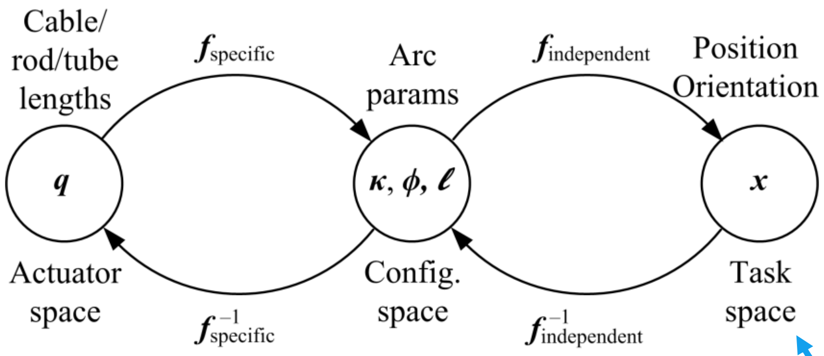
Objectives

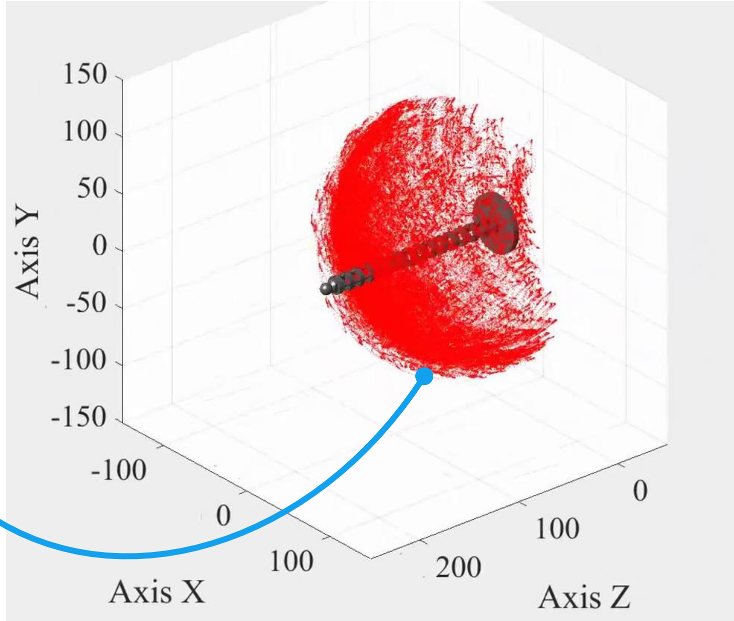
- Geometric representation of the robot (Inverse kinematic modeling or IKM)
- Position of the end-Effector (Forward Kinematic modeling or FKM)

Forward Kinematic Modeling (FKM)

FKM
$$x = f_{ind}(\kappa, \phi, l)$$

FKM
$$\kappa, \phi, l = f_s(q)$$





Inverse Kinematic Modeling (IKM)

PCC approach using Denavit Hartenberg formalism

- Denavit–Hartenberg representation

•z_{n-1} is driven by the joint between the body C_{n-1} and C_n
•x_n is driven by the common normal to z_{n-1} and $z_n (x_n = z_{n-1} \wedge z_n)$
•$d = d_n$, is the Offset about z_{n-1} beetween x_{n-1} and x_n

•$q = q_n$, is the angle about z_{n-1} between x_{n-1} and x_n
•$a = a_n$, is the Offset with respect to x_n between z_{n-1} and z_n
•$\alpha = \alpha_n$, is the angle about x_n between z_{n-1} and z_n

$$R_{n-1,n} \qquad T_{n-1,n}$$

$$T_{n-1,n} = \begin{pmatrix} \cos(\vartheta) & -\cos(\alpha)\sin(\vartheta) & \sin(\alpha)\sin(\vartheta) & r\cos(\vartheta) \\ \sin(\vartheta) & \cos(\alpha)\cos(\vartheta) & -\sin(\alpha)\cos(\vartheta) & r\sin(\vartheta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

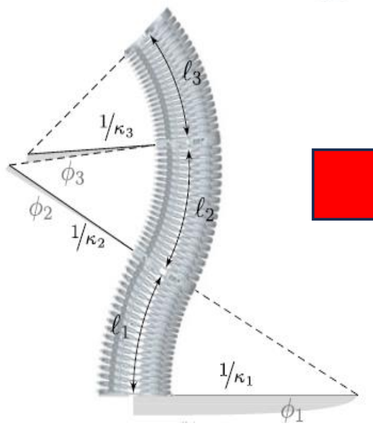
The pose of the body $\boldsymbol{n}$ with respect to $\boldsymbol{n-1}$ may be represented by the matrix $T_{n-1,n}$

The Upper Left $\mathbf{3 \times 3}$ submatrix represents the relative orientation of the Two bodies

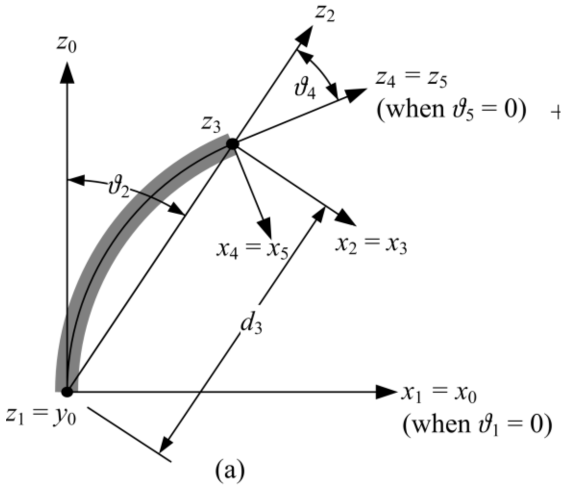
$$R_{n-1,n} = \begin{pmatrix} \cos(\vartheta) & -\cos(\alpha)\sin(\vartheta) & \sin(\alpha)\sin(\vartheta) \\ \sin(\theta) & \cos(\alpha)\cos(\vartheta) & -\sin(\alpha)\cos(\vartheta) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

The Upper Left $\mathbf{3 \times 1}$ submatrix represents the relative position of the Two bodies

$$T_{n-1,n} = \begin{pmatrix} \boldsymbol{r\cos(\vartheta)} \\ \boldsymbol{r\sin(\vartheta)} \\ \boldsymbol{d} \end{pmatrix}$$

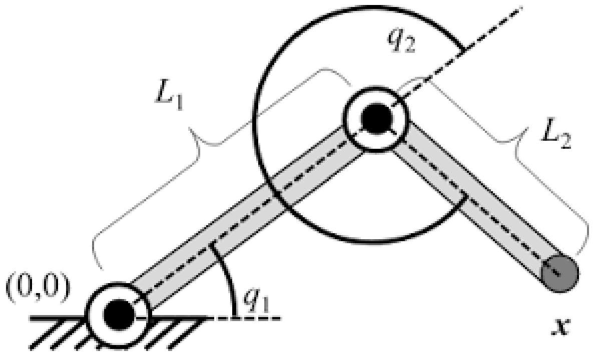


PCC based modeling



Denavit Hartenberg formalism

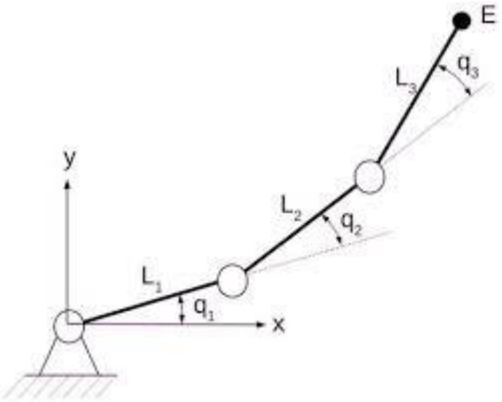
Exercice 1



Practical Exercices:

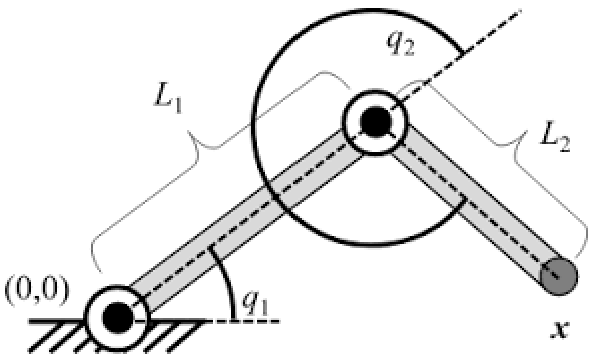
- Give the **FKM** by means of : Vector modeling approach
- Give the **IKM** by means of : Vector modeling approach

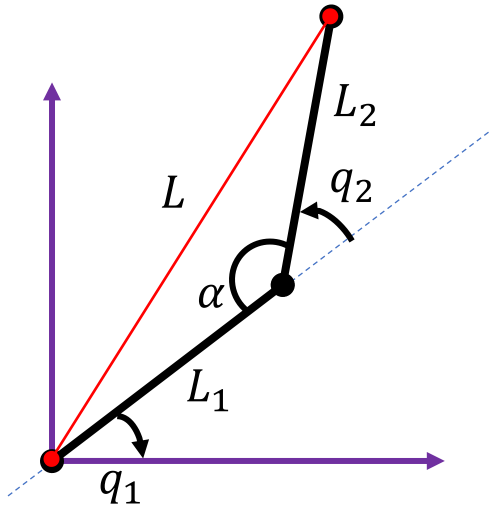
Exercice 2



Denavit Hartenberg formalism

Solution to Exercice 1





Forward Geometric Modeling (Modèle Géométrique Directe)

$$(x_1, \dots, x_n) = f(q_1, q_2, q_3 \dots q_n)$$

$$\begin{cases} x = L_1 \cos(q_1) + L_2 \cos(q_1 + q_2) \\ y = L_1 \sin(q_1) + L_2 \sin(q_1 + q_2) \end{cases}$$

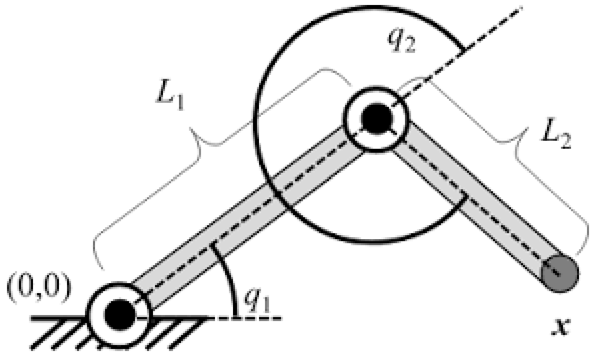
Inverse Geometric Modeling (Modèle Géométrique Inverse) $\quad q_1, q_2, q_3 \dots q_n = f^{-1}(x_1, \dots, x_n)$

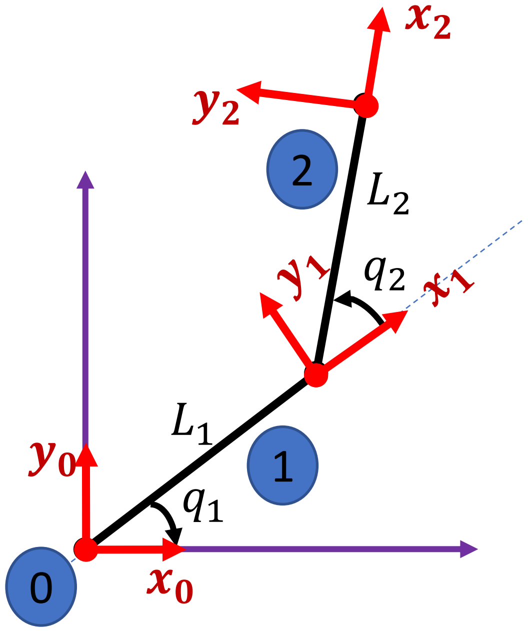
$$\begin{cases} L^2 = x^2 + y^2 \\ L^2 = L_1^2 + L_2^2 - 2L_1L_2\cos(\alpha) \end{cases} \Rightarrow \begin{cases} L^2 = x^2 + y^2 \\ L^2 = L_1^2 + L_2^2 + 2L_1L_2\cos(q_2) \end{cases} \Rightarrow q_2 = \mp \cos^{-1}\left(\frac{x^2 + y^2 - (L_1^2 + L_2^2)}{2L_1L_2}\right)$$

$$\begin{cases} (L_1 + L_2\, cos(q_2))\cos(q_1) - L_2\, sin(q_2)\, sin(q_1) = x \\ L_1 \sin(q_2)\, cos(q_1) + (L_1 + L_2\, cos(q_2))\sin(q_1) = y \end{cases} \Rightarrow q_1 = \tan^{-1}\left(\frac{y(L_1 + L_2\, cos(q_2)) - xL_2\sin(q_2)}{y(L_1 + L_2\, cos(q_2)) + xL_2\sin(q_2)}\right)$$

Denavit Hartenberg formalism

Solution to Exercice 1





Forward Geometric Modeling (Modèle Géométrique Directe)

$$T_{0,1} = \begin{pmatrix} \cos(\boldsymbol{q_1}) & -\sin(\boldsymbol{q_1}) & 0 & L_1\cos(\boldsymbol{q_1}) \\ \sin(\boldsymbol{q_1}) & \cos(\boldsymbol{q_1}) & 0 & L_1\sin(\boldsymbol{q_1}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{1,2} = \begin{pmatrix} \cos(\boldsymbol{q_2}) & -\sin(\boldsymbol{q_2}) & 0 & L_2\cos(\boldsymbol{q_2}) \\ \sin(\boldsymbol{q_2}) & \cos(\boldsymbol{q_2}) & 0 & L_2\sin(\boldsymbol{q_2}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

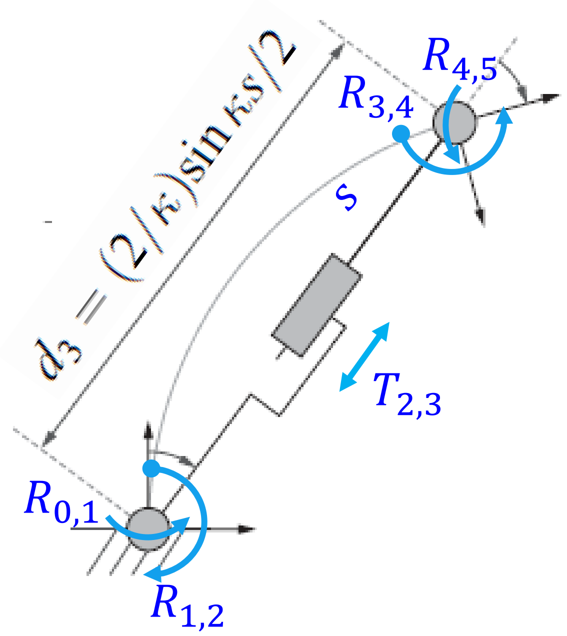
$$T_{0,2} = T_{0,1}T_{1,2} = \begin{pmatrix} \cos(\boldsymbol{q_1}+\boldsymbol{q_2}) & -\sin(\boldsymbol{q_1}+\boldsymbol{q_2}) & 0 & L_1\cos(\boldsymbol{q_1}) + L_2\cos(\boldsymbol{q_1}+\boldsymbol{q_2}) \\ \sin(\boldsymbol{q_1}+\boldsymbol{q_2}) & \cos(\boldsymbol{q_1}+\boldsymbol{q_2}) & 0 & L_2\sin(\boldsymbol{q_2}) + L_2\sin(\boldsymbol{q_1}+\boldsymbol{q_2}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

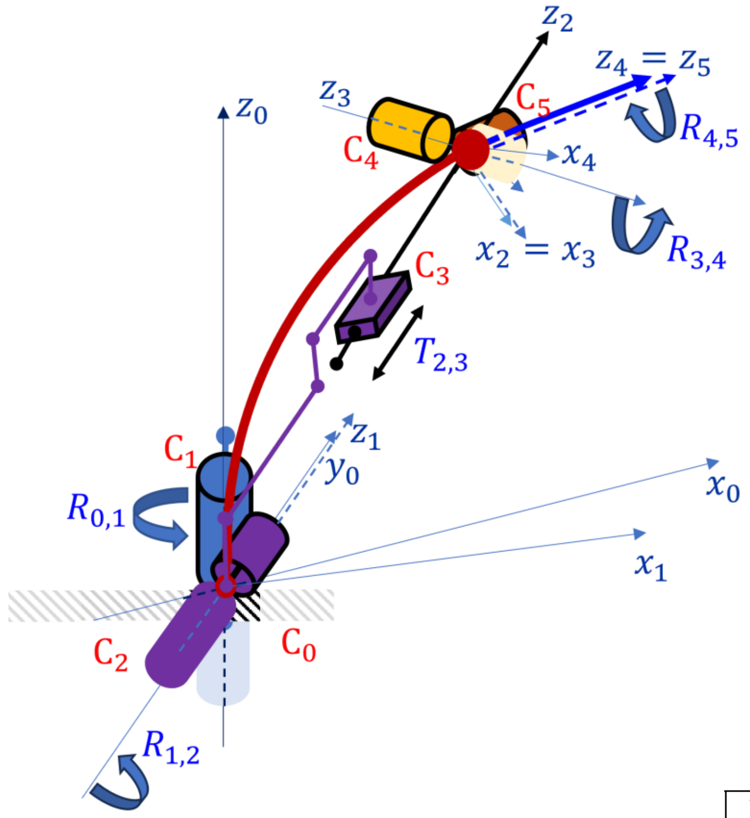
Denavit–Hartenberg representation

Elements	$\boldsymbol{q}$(rad)	$\boldsymbol{d}(m)$	$\boldsymbol{r}(m)$	$\boldsymbol{\alpha}$(rad)
1	$\boldsymbol{q_1}$	0	L_1	0
2	$\boldsymbol{q_2}$	**0**	L_2	0

$$R_{0,2} = \begin{pmatrix} \cos(\boldsymbol{q_1}+\boldsymbol{q_2}) & -\sin(\boldsymbol{q_1}+\boldsymbol{q_2}) & 0 \\ \sin(\boldsymbol{q_1}+\boldsymbol{q_2}) & \cos(\boldsymbol{q_1}+\boldsymbol{q_2}) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_{0,2} = \begin{pmatrix} L_1\cos(\boldsymbol{q_1}) + L_2\cos(\boldsymbol{q_1}+\boldsymbol{q_2}) \\ L_2\sin(\boldsymbol{q_2}) + L_2\sin(\boldsymbol{q_1}+\boldsymbol{q_2}) \\ 0 \end{pmatrix}$$





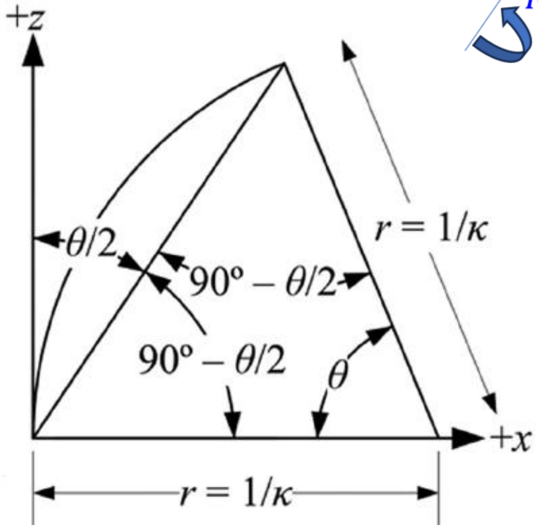
Shape configuration variable (κ, ϕ, l)

Length of the portion: $l = s$

Curvature: κ

$$\theta = \kappa s$$

$$d_3 = 2r\sin\frac{\theta}{2}$$

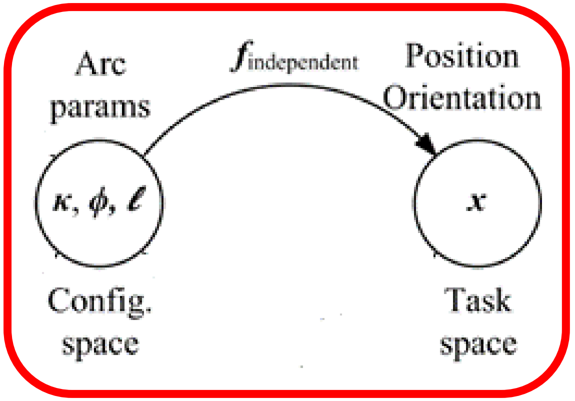


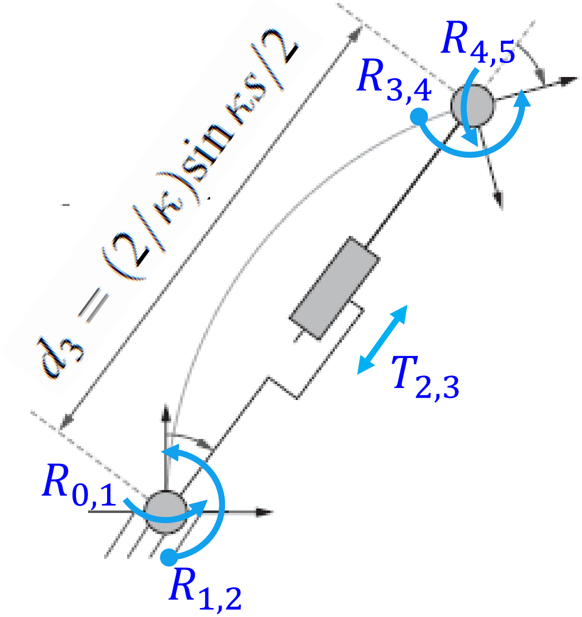
Denavit Hartenberg parameter

Link	ϑ	d	a	α
1	$\vartheta_1 = \phi$	0	0	$-\pi/2$
2	$\vartheta_2 = \kappa s/2$	0	0	$\pi/2$
3	0	$d_3 = (2/\kappa)\sin\kappa s/2$	0	$-\pi/2$
4	$\vartheta_4 = \kappa s/2$	0	0	$\pi/2$
5	$\vartheta_5 = -\phi$	0	0	0

PCC approach using Denavit Hartenberg formalism

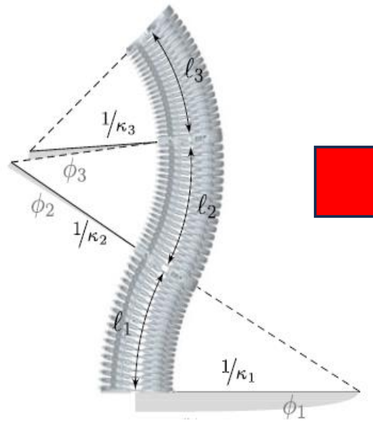
FKM



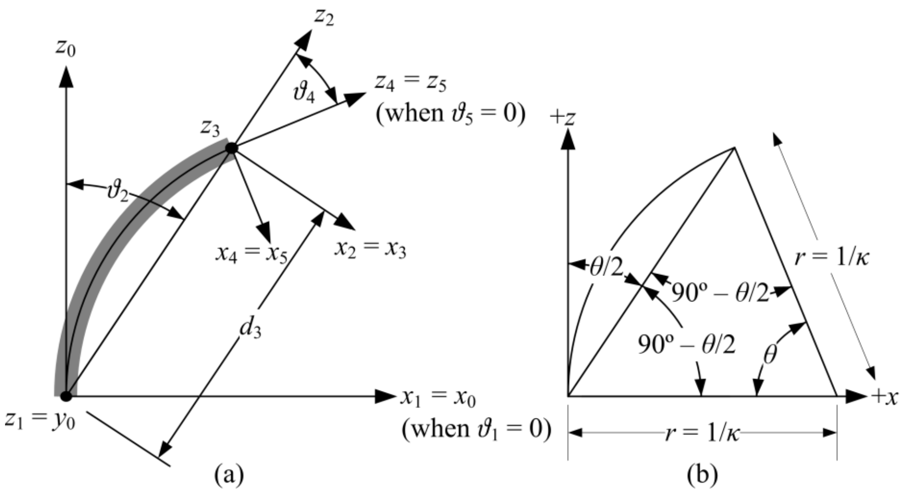


Denavit Hartenberg parameter

Link	ϑ	d	a	α
1	$\vartheta_1 = \phi$	0	0	$-\pi/2$
2	$\vartheta_2 = \kappa s/2$	0	0	$\pi/2$
3	0	$d_3 = (2/\kappa)\sin\kappa s/2$	0	$-\pi/2$
4	$\vartheta_4 = \kappa s/2$	0	0	$\pi/2$
5	$\vartheta_5 = -\phi$	0	0	0



PCC based modeling



$$\boldsymbol{R} \in \boldsymbol{SO_3} \qquad \boldsymbol{p} \in \mathbb{R}^3$$

$$T = \begin{bmatrix} \cos\kappa s\cos\phi & -\sin\phi & \sin\kappa s\cos\phi & \frac{\cos\phi(1-\cos\kappa s)}{\kappa} \\ \cos\kappa s\sin\phi & \cos\phi & \sin\kappa s\sin\phi & \frac{\sin\phi(1-\cos\kappa s)}{\kappa} \\ -\sin\kappa s & 0 & \cos\kappa s & \frac{\sin\kappa s}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1

Application case of the FKM

Denavit Hartenberg parameter

Link	ϑ	d	a	α
1	$\vartheta_1 = \phi$	0	0	$-\pi/2$
2	$\vartheta_2 = \kappa s/2$	0	0	$\pi/2$
3	0	$d_3 = (2/\kappa)\sin\kappa s/2$	0	$-\pi/2$
4	$\vartheta_4 = \kappa s/2$	0	0	$\pi/2$
5	$\vartheta_5 = -\phi$	0	0	0

$$T_{n-1,n} = T_{n-1}^{n} = \begin{pmatrix} \cos(\vartheta) & -\cos(\alpha)\sin(\vartheta) & \sin(\alpha)\sin(\vartheta) & r\cos(\vartheta) \\ \sin(\vartheta) & \cos(\alpha)\cos(\vartheta) & -\sin(\alpha)\cos(\vartheta) & r\sin(\vartheta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$R_{n-1,n}$ $\quad$ $\boldsymbol{T_{n-1,n}}$

$$T_0^1 = \begin{bmatrix} \cos\phi & 0 & -\sin\phi & 0 \\ \sin\phi & 0 & +\cos\phi & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_1^2 = \begin{bmatrix} \cos\left(\frac{\kappa l}{2}\right) & 0 & \sin\left(\frac{\kappa l}{2}\right) & 0 \\ \sin\left(\frac{\kappa l}{2}\right) & 0 & -\cos\left(\frac{\kappa l}{2}\right) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

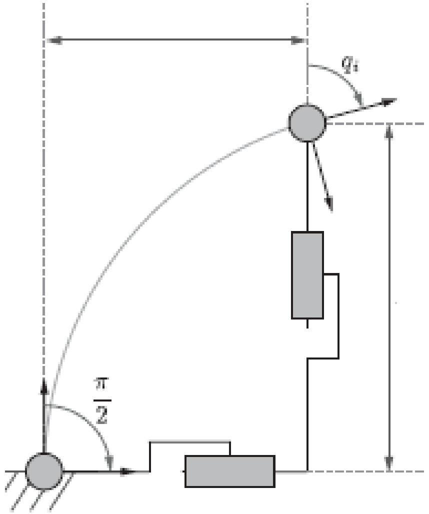
$$T_4^5 = \begin{bmatrix} \cos\phi & \sin\phi & 0 & 0 \\ -\sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad T_3^4 = \begin{bmatrix} \cos\left(\frac{\kappa l}{2}\right) & 0 & \sin\left(\frac{\kappa l}{2}\right) & 0 \\ \sin\left(\frac{\kappa l}{2}\right) & 0 & -\cos\left(\frac{\kappa l}{2}\right) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{T_0^2 = T_0^1 T_1^2} = \begin{bmatrix} \cos\phi\cdot\cos\left(\frac{\kappa l}{2}\right) & -\sin\phi & \cos\phi\cdot\sin\left(\frac{\kappa l}{2}\right) & 0 \\ \cos\left(\frac{\kappa l}{2}\right)\cdot\sin\phi & \cos\phi & \sin\phi\cdot\sin\left(\frac{\kappa l}{2}\right) & 0 \\ 0 & 0 & \cos\left(\frac{\kappa l}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

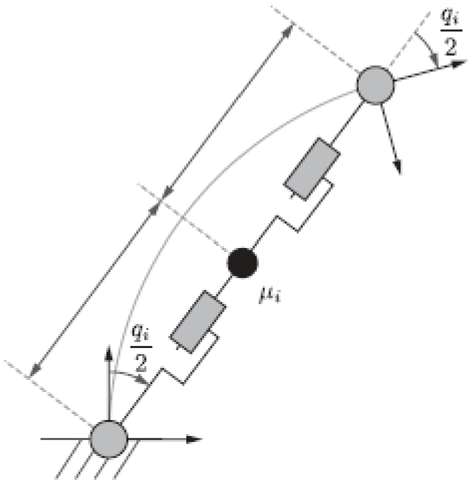
$$\boldsymbol{T_0^3} = \begin{bmatrix} \cos\phi\cdot\cos\left(\frac{\kappa l}{2}\right) & -\cos\phi\cdot\sin\left(\frac{\kappa l}{2}\right) & \sin\phi & -\cos\phi\cdot\sin\left(\frac{\kappa l}{2}\right)\cdot d_3 \\ \cos\left(\frac{\kappa l}{2}\right)\cdot\sin\phi & -\sin\phi\cdot\sin\left(\frac{\kappa l}{2}\right) & -\cos\phi & -\sin\phi\cdot\sin\left(\frac{\kappa l}{2}\right)\cdot d_3 \\ 0 & -\cos\left(\frac{\kappa l}{2}\right) & 0 & \cos\left(\frac{\kappa l}{2}\right)\cdot d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{T = T_0^4} = \begin{bmatrix} \cos\kappa l\cos\phi & -\sin\phi & \sin\kappa l\cos\phi & \frac{\cos\phi(1-\cos\kappa l)}{\kappa} \\ \cos\kappa l\sin\phi & \cos\phi & \sin\kappa l\sin\phi & \frac{\sin\phi(1-\cos\kappa l)}{\kappa} \\ -\sin\kappa l & 0 & \cos\kappa l & \frac{\sin\kappa l}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

PCC approach using Denavit Hartenberg formalism



A

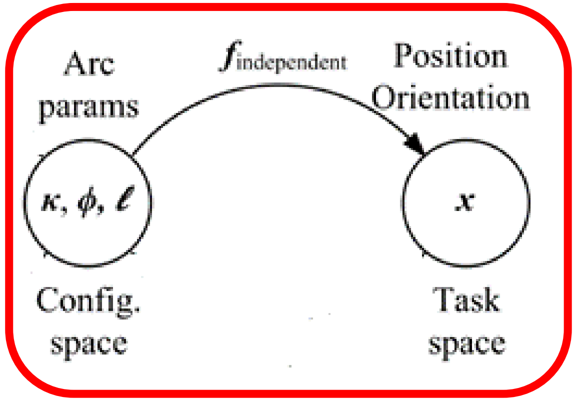


B

1. Retrieve the DH parameter
2. Compute the transformation matrix

PCC approach using Arc geometry method

FKM



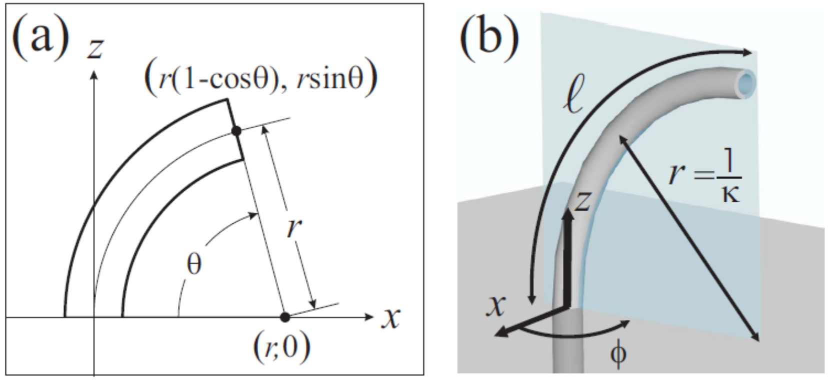
In fig(a), the bending motion occurs in the **xy** plan, with a rotation about the $\boldsymbol{y}$-axis, $R_y(\theta)$

The portion tip p is defined as:

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{r(1-\cos\theta)} \\ \boldsymbol{\sin\theta} \\ \boldsymbol{0} \end{bmatrix}$$

In fig(b), a rotation motion $R_z(\phi)$ relative to the $\boldsymbol{z}$-axis migth induce a 3D motion. This transformation reads:

$$\boldsymbol{T_z} = \begin{bmatrix} \boldsymbol{R_z(\phi)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}$$



Therefore, the total transformation of the section is consistent with:

$$T = \begin{bmatrix} R_z(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_y(\theta) & \boldsymbol{p} \\ 0 & 1 \end{bmatrix} \Rightarrow T = \begin{bmatrix} \cos\kappa s \cos\phi & -\sin\phi & \sin\kappa s \cos\phi & \frac{\cos\phi(1-\cos\kappa s)}{\kappa} \\ \cos\kappa s \sin\phi & \cos\phi & \sin\kappa s \sin\phi & \frac{\sin\phi(1-\cos\kappa s)}{\kappa} \\ -\sin\kappa s & 0 & \cos\kappa s & \frac{\sin\kappa s}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Application case of shape reconstruction of a needle

Denavit Hartenberg parameter

Link	ϑ	d	a	α
1	$\vartheta_1 = \phi$	0	0	$-\pi/2$
2	$\vartheta_2 = \kappa s/2$	0	0	$\pi/2$
3	0	$d_3 = (2/\kappa)\sin\kappa s/2$	0	$-\pi/2$
4	$\vartheta_4 = \kappa s/2$	0	0	$\pi/2$
5	$\vartheta_5 = -\phi$	0	0	0

$\boldsymbol{R} \in \boldsymbol{SO}_3$ $\quad$ $\boldsymbol{p} \in \mathbb{R}^3$

$$T = \begin{bmatrix} \cos\kappa s \cos\phi & -\sin\phi & \sin\kappa s \cos\phi & \dfrac{\cos\phi(1-\cos\kappa s)}{\kappa} \\ \cos\kappa s \sin\phi & \cos\phi & \sin\kappa s \sin\phi & \dfrac{\sin\phi(1-\cos\kappa s)}{\kappa} \\ -\sin\kappa s & 0 & \cos\kappa s & \dfrac{\sin\kappa s}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1. Create a document RoBio
2. Create a new matlab script (.m)
3. Save this as DH_BME.m in RoBio

DH_BME.m

```
function [T] = DH_BME(phi,kappa,s)

T= [cos(phi)*cos(kappa*s), -sin(phi), sin(kappa*s)*cos(phi), cos(phi)*(1-cos(kappa*s))/kappa;
    sin(phi)*cos(kappa*s), cos(phi), sin(kappa*s)*sin(phi), sin(phi)*(1-cos(kappa*s))/kappa;
    -sin(kappa*s), 0, cos(kappa*s), (sin(kappa*s))/kappa;
    0, 0, 0, 1 ];
end
```

Application case of shape reconstruction of a one section needle

beginning

FKM_BME.m

```
function [Shape]= FKM_BME(phi,theta,L)

% Finger geometry properties (in USI)
%L1 = 80e-3;                %[m] length of the needle
%Theta;                     %[deg]
%phi;                       %[deg]

kappa=(theta*pi)/(180*L);   %[m^-1]
phi=phi*pi/180;             %[rad]

T=[ ];                      %Empty Transformation matrix initialization
n=400;                      %number of point for discretization
p=[ ];                      % Empty shape coordinates

    %%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:n
   s= (i-1)*L/(n-1);              % Curvilinear coordinates on the needle
   T(:,:,i)= DH_BME(phi,kappa,s); % Transformation matrix computation
   p(i,:)=T(1:3,4,i);             % Extraction of the Positions
end

Shape=p;                          % saving each position p into Shape
```

4 Create a new matlab script (.m)

3 Save this as FKM_BME.m in RoBio

following

```
figure(1);
ax = gca;
xlabel('x (m)'); ylabel('y(m)'); zlabel('z(m)');

title('Shape Kinematics Reconstruction: PCC approach')
hold on,
plot3(p(:,1),p(:,2),p(:,3),'.');
grid on,
ax.View = [-60 30];
end
```

Application case of shape reconstruction of a two section needle

1 beginning

FKM2_BME.m

1 Create a new matlab script (.m)

```
function [Shape]= FKM_BME(phi,theta,L)

% Finger geometry properties (in USI)
%L= [80e-3 30e-3]          %[m]   length of the 1st and 2nd needle portion
%theta= [theta1 theta2]    %[deg]  theta of the 1st and 2nd needle portion
%phi   = [phi1 phi2]       %[deg]    phi of the 1st and 2nd needle portion

kappa=(theta*pi)./(180*L);    %[m^-1]
phi=phi*pi/180;               %[rad]

T01=[ ]; T02=[ ]; T12=[ ];    %Empty Transformation matrix initialization
n=400;                        %number of point for discretization
p1=[ ]; p2=[ ];               % Empty shape coordinates

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:n
   s1= (i-1)*L(1)/(n-1);
   T01(:,:,i)= DH_BME(phi(1),kappa(1),s1);
   p1(i,:)=T01 (1:3,4,i);                  % Extraction of the Positions
end
```

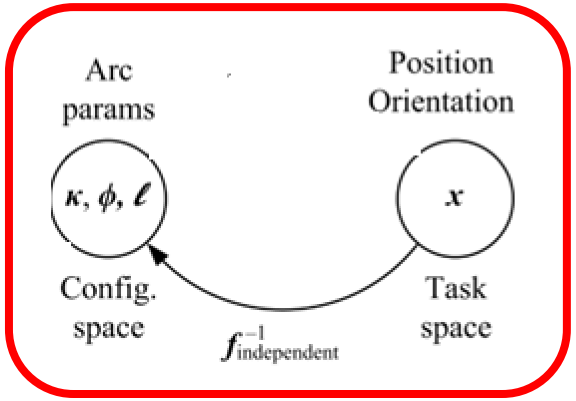
1 Following

```
for j=1:n
   s2= (i-1)*L(2)/(n-1);
   T12(:,:,j)= DH_BME(phi(2),kappa(2),s2);
   T02(:,:,j)= T01(:,:,n)* T12(:,:,j);
   p2(j,:)=T02 (1:3,4,j);
end
Shape=[p1 p2];

figure(2);
ax = gca;
xlabel(‘x (m)’); ylabel(‘y(m)’); zlabel(‘z(m)')

title('Shape Kinematics Reconstruction: PCC approach’)
hold on,
plot3(p1(:,1),p1(:,2),p1(:,3),'.’);
plot3(p2(:,1),p2(:,2),p2(:,3),'.’);
grid on,
ax.View = [-60 30];
end
```

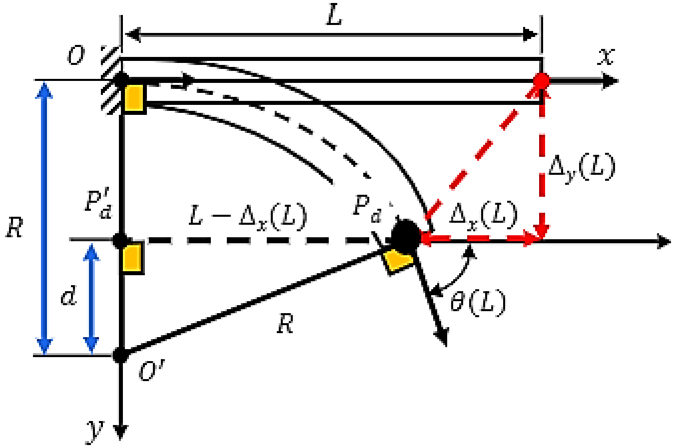
2 Save this as FKM2_BME.m in RoBio

PCC approach using DH method

IKM



Case of a one segment planar robot



$$\boldsymbol{R = d + \Delta_y(L)}$$

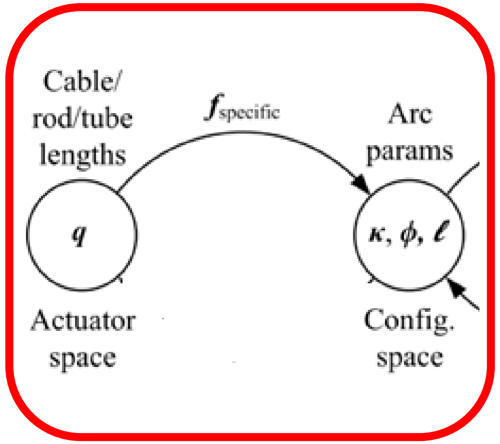
$$\boldsymbol{R = d + \Delta_y(L)}$$

$$\begin{cases} R - d = y_d \\ R^2 - d^2 = x_d^2 \end{cases} \Rightarrow R + d = \frac{x_d^2}{y_d} \Rightarrow \begin{cases} R = \frac{1}{2}\frac{y_d^2 + x_d^2}{y_d} \\ d = \frac{1}{2}\frac{y_d^2 + x_d^2}{y_d} - y_d \end{cases} \mathbf{c} \Rightarrow \begin{cases} \theta = \cos^{-1}\left(\frac{d}{R}\right)\mathbf{c} \\ L = R\theta \end{cases}$$

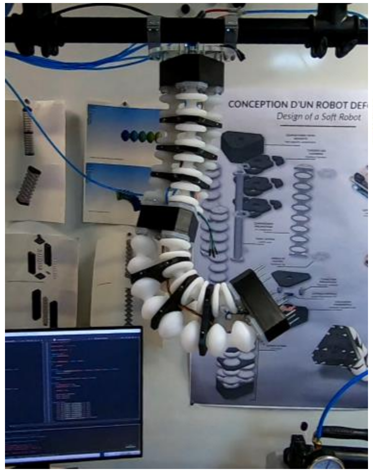
CRIStAL Centre de Recherche en Informatique, Signal et Automatique de Lille

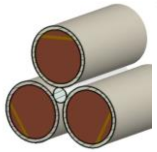
centralelille ÉCOLE CENTRALE DE LILLE

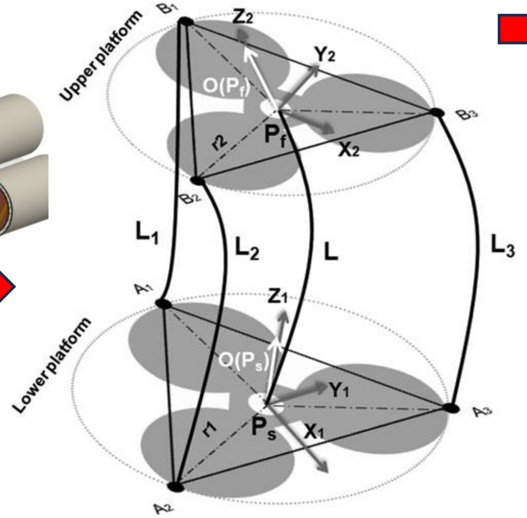
FKM

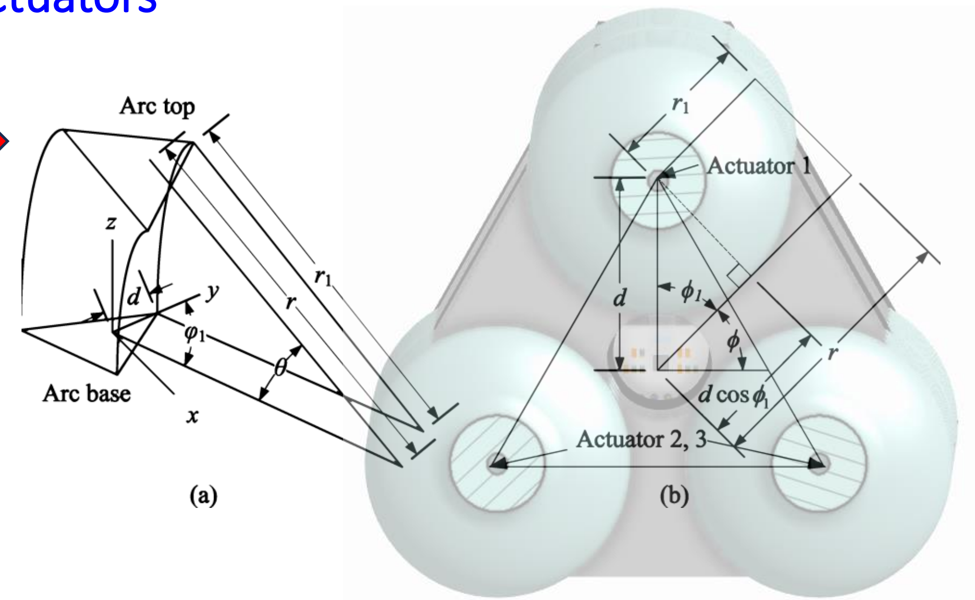


Case of Continuously bending Actuators









$$r_i = r - d\cos\phi_i$$

$$l = l_i - \theta d\cos\phi_i$$

2

ϕ The bending plane of the soft robot

$$\phi_1 = 90° - \phi$$
$$\phi_2 = 210° - \phi$$
$$\phi_3 = 330° - \phi$$
$$l = r\theta$$
$$l_i = r_i\theta$$

3

$$l(q) = \frac{l_1 + l_2 + l_3}{3}$$ 4

$$\phi(q) = \tan^{-1}\left(\frac{\sqrt{3}(l_2 + l_3 - 2l_1)}{3(l_2 - l_3)}\right)$$ 5

$$\kappa(q) = \frac{2\sqrt{({l^2}_1 + {l^2}_2 + {l^2}_3 - l_1 l_2 - l_1 l_3 - l_2 l_3)}}{d(l_1 + l_2 + l_3)}$$ 6

PCC approach using Denavit Hartenberg formalism

Parameters	Values {i=1, i=2, i=3}	Description
d	$\{0.101\}$	Radial cable distance of the portion
L_i	$\{0.016 \quad 0.006 \quad 0.006\}$	Lenght of the cable i
i is the cable		

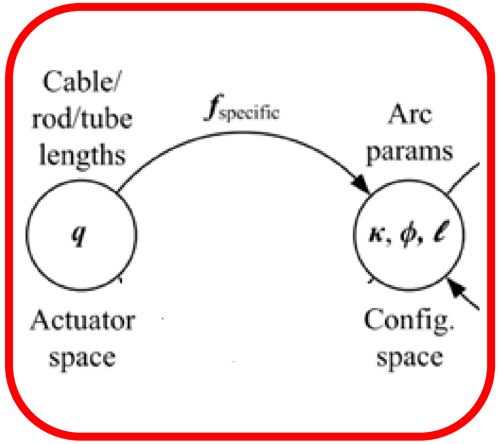
1 For one section, compute the shape config. Variables

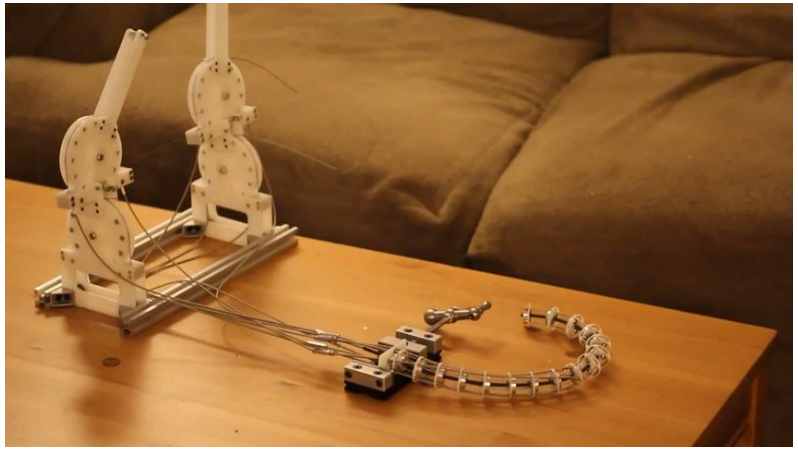
2 Compute the related shape configuration in the task configugation

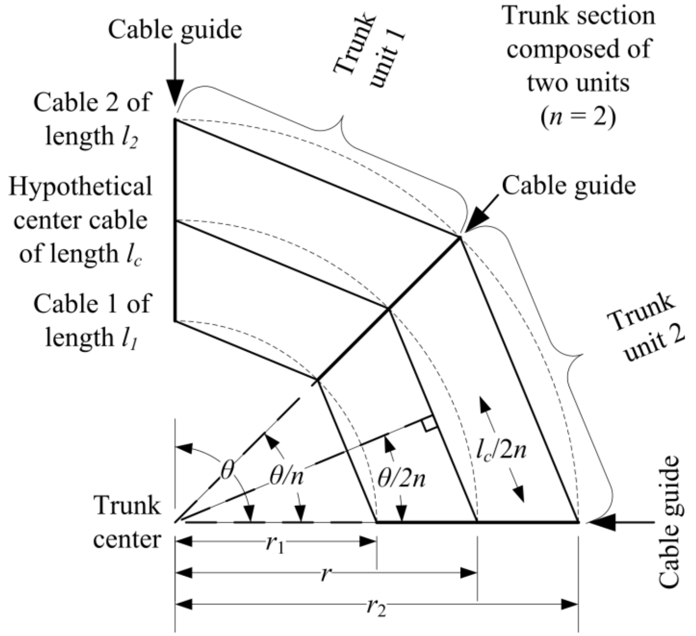
1

Case of Tendon driven actuators

FKM







$$\theta = \kappa l$$

$$l_c = 2nr\sin\left(\frac{\theta}{n}\right)$$

$$l_c = l_i + 2dn\sin\left(\frac{\theta}{n}\right)\cos\phi_i$$

$$l_c = \frac{l_1 + l_2 + l_3}{3}$$

7

ϕ The bending plane of the soft robot

$$\phi_1 = 90° - \phi$$

$$\phi_2 = 210° - \phi$$

$$\phi_3 = 330° - \phi$$

$$l_i = 2nr_i\sin\left(\frac{\theta}{n}\right)$$

8

$$\phi(q) = \tan^{-1}\left(\frac{\sqrt{3}(l_2 + l_3 - 2l_1)}{3(l_2 - l_3)}\right)$$

10

$$\kappa(q) = \frac{2\sqrt{(l^2{}_1 + l^2{}_2 + l^2{}_3 - l_1 l_2 - l_1 l_3 - l_2 l_3)}}{d(l_1 + l_2 + l_3)}$$

11

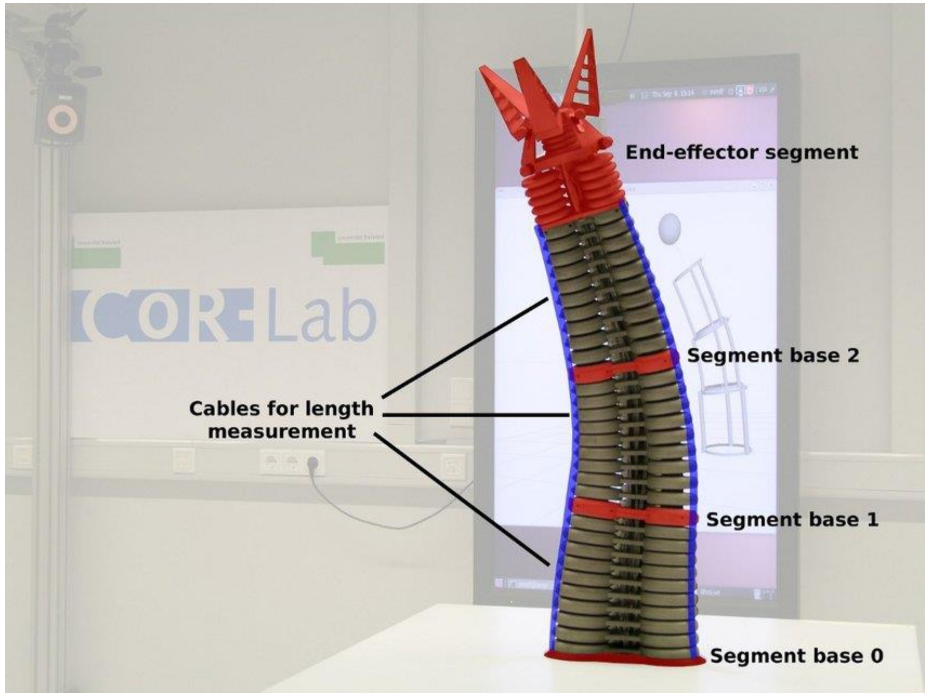
$$l(q) = \frac{2n}{\kappa}\sin^{-1}\left(\frac{l_c\kappa}{2n}\right)$$

12

Practice case

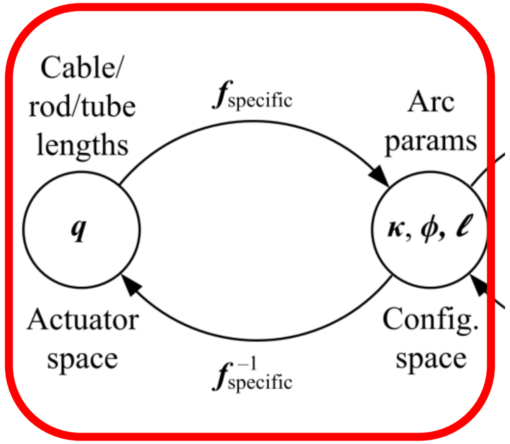
1. For each of the grey section, Identify the number of unit
2. For each section, compute the shape config. Variables
3. Compute the related shape configuration in the task configugation

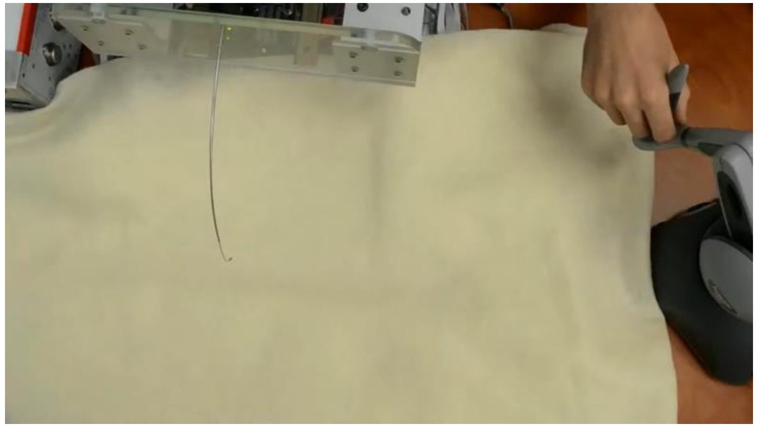
Parameters	Values {i=1, i=2, i=3}	Description
d_i	$\{0.101 \quad 0.086\}$	Radial cable distance at the portion i
L_{2k}	$\{0.101 \quad 0.086 \quad 0.071\}$	Radial cable distance at the portion i
L_{1k}	$\{0.016 \quad 0.006 \quad 0.06\}$	Lenght of the portion i
k is the cable, i the portion		

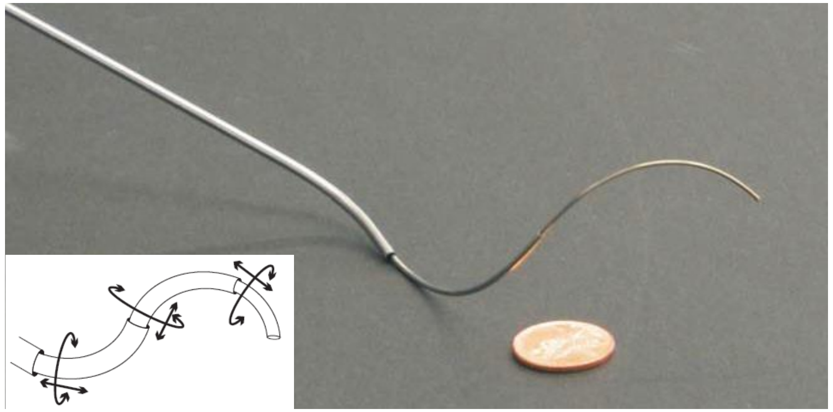


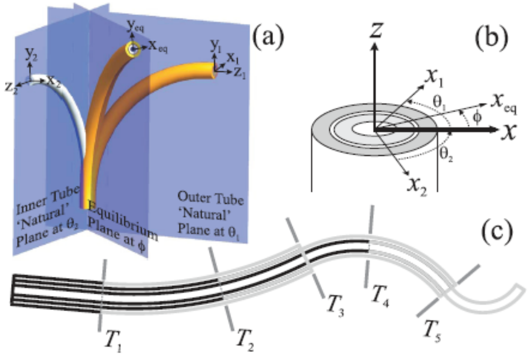
Case of concentric tube continuum robots

FKM









Translation (t) and Rotation (θ)

$$q = [t_1\ \theta_1 \cdots t_n\ \theta_n]^T$$

κ: the curvature the soft robot

κ_i: the curvature of each tube

M: the constant bending moment

E: Elastic modulus

I: Inertia of the robot

$$\kappa - \kappa_i = \frac{M}{EI}$$

13

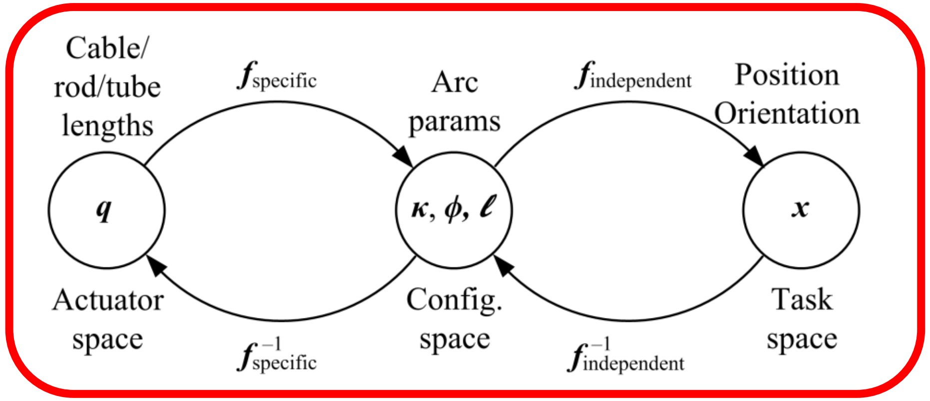
$$\kappa_x = \left(\frac{\sum E_i I_i \kappa_i \cos\theta_i}{\sum E_i I_i}\right) \quad \kappa_y = \left(\frac{\sum E_i I_i \kappa_i \sin\theta_i}{\sum E_i I_i}\right)$$

14

$$\phi(q) = \sin^{-1}\left(\frac{\kappa_y}{\kappa_x}\right)$$

15

IKM



Forward Velocity Modeling $\dot{p} = f(\dot{q}_1, \dot{q}_i, \dots, \dot{q}_n)$

FKM : $x = \boldsymbol{f}_{\boldsymbol{spec}} \circ \boldsymbol{f}_{\boldsymbol{ind}}(q) \Longrightarrow [x\ y\ z] = \boldsymbol{f}_{\boldsymbol{spec}} \circ \boldsymbol{f}_{\boldsymbol{ind}}(q)$

$\dot{p} = \mathrm{J}[\dot{q}]$ J is the Jacobean Matrix $\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \mathrm{J} \begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{pmatrix}$

$\mathrm{J} = \mathbf{J}_{\mathbf{ind}} \mathbf{J}_{\mathbf{spec}}$

$$\frac{\boldsymbol{d p_x}}{dt} = \mathrm{J} \frac{\boldsymbol{d}}{dt} [\boldsymbol{q}]$$

Therefore

$$[\boldsymbol{\Delta q}] = \mathrm{J}^{-1} [\boldsymbol{\Delta} p]$$

J^{-1} is the inverse of the Jacobean Matrix

- J must be a square matrix to be inversible
- Determinant(J) ≠ 0

To adress this issue, we use the pseudo inverse $\bar{\mathbf{J}}^{-1}$ of $\mathbf{J}$

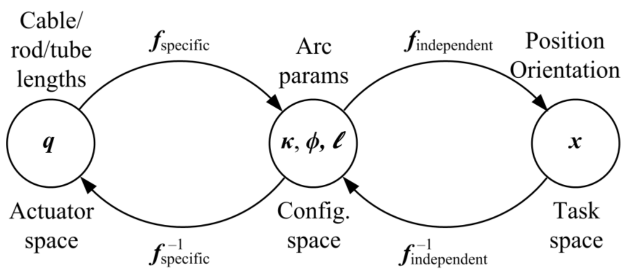
$$\bar{\mathbf{J}}^{-1} = \left(\mathbf{J}^{\mathbf{T}} \mathbf{J}\right)^{-1} \mathbf{J}^{\mathbf{T}}$$

CRIStAL Centre de Recherche en Informatique, Signal et Automatique de Lille

centralelille ÉCOLE CENTRALE DE LILLE

Application case of the IKM for the configuration space / task space mapping

IKM



$\boldsymbol{R} \in \boldsymbol{SO_3}$ $\quad$ $\boldsymbol{p} \in \mathbb{R}^3$ **(FKM)**

$$T = \begin{bmatrix} \cos\kappa l \cos\phi & -\sin\phi & \sin\kappa l \cos\phi & \frac{\cos\phi(1-\cos\kappa l)}{\kappa} \\ \cos\kappa l \sin\phi & \cos\phi & \sin\kappa l \sin\phi & \frac{\sin\phi(1-\cos\kappa l)}{\kappa} \\ -\sin\kappa l & 0 & \cos\kappa l & \frac{\sin\kappa l}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{p_x} \\ \boldsymbol{p_y} \\ \boldsymbol{p_z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f_1}(k,\phi,l) \\ \boldsymbol{f_2}(k,\phi,l) \\ \boldsymbol{f_3}(k,\phi,l) \end{bmatrix} = \begin{bmatrix} \frac{\cos\phi(1-\cos\kappa l)}{\kappa} \\ \frac{\sin\phi\ (1-\cos\kappa l)}{\kappa} \\ \frac{\sin\kappa \boldsymbol{l}}{\kappa} \end{bmatrix}$$

Therefore

$$\begin{array}{cccc} & \kappa & \phi & l \\ \boldsymbol{J_{ind}} = & \left[\begin{matrix} \frac{\boldsymbol{p_x}}{\boldsymbol{\partial\kappa}} & \frac{\boldsymbol{p_x}}{\boldsymbol{\partial\phi}} & \frac{\boldsymbol{p_x}}{\boldsymbol{\partial l}} \\ \frac{\boldsymbol{p_y}}{\boldsymbol{\partial\kappa}} & \frac{\boldsymbol{p_y}}{\boldsymbol{\partial\phi}} & \frac{\boldsymbol{p_y}}{\boldsymbol{\partial l}} \\ \frac{\boldsymbol{p_z}}{\boldsymbol{\partial\kappa}} & \frac{\boldsymbol{p_z}}{\boldsymbol{\partial\phi}} & \frac{\boldsymbol{p_z}}{\boldsymbol{\partial l}} \end{matrix}\right. & & \left.\right] \end{array}$$

$\boldsymbol{p_x} = \boldsymbol{f_1}(k,\phi,l)$

$\boldsymbol{p_y} = \boldsymbol{f_2}(k,\phi,l)$

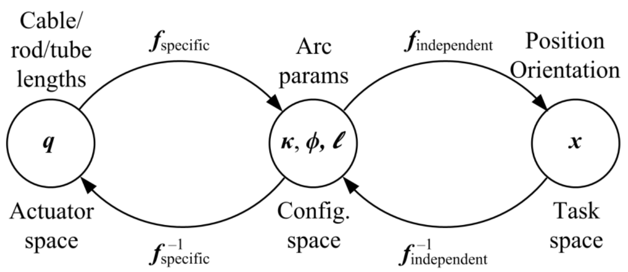
$\boldsymbol{p_z} = \boldsymbol{f_3}(k,\phi,l)$

$$\begin{bmatrix} \boldsymbol{\Delta p_x} \\ \boldsymbol{\Delta p_y} \\ \boldsymbol{\Delta p_z} \end{bmatrix} = \boldsymbol{J_{ind}} \begin{bmatrix} \boldsymbol{\Delta\kappa} \\ \boldsymbol{\Delta\phi} \\ \boldsymbol{\Delta l} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{\Delta\kappa} \\ \boldsymbol{\Delta\phi} \\ \boldsymbol{\Delta l} \end{bmatrix} = \boldsymbol{J_{ind}^{-1}} \begin{bmatrix} \boldsymbol{\Delta p_x} \\ \boldsymbol{\Delta p_y} \\ \boldsymbol{\Delta p_z} \end{bmatrix}$$

Application case of the IKM for the configuration space / task space mapping

IKM



$$f_{spec}(l) \Rightarrow \begin{cases} \phi(q) = \tan^{-1}\left(\dfrac{\sqrt{3}(l_2 + l_3 - 2l_1)}{3(l_2 - l_3)}\right) \\ \kappa(q) = \dfrac{2\sqrt{(l^2{}_1 + l^2{}_2 + l^2{}_3 - l_1 l_2 - l_1 l_3 - l_2 l_3)}}{d(l_1 + l_2 + l_3)} \\ l(q) = \dfrac{l_1 + l_2 + l_3}{3} \end{cases} \Rightarrow \begin{bmatrix} \Delta k \\ \Delta \phi \\ \Delta l \end{bmatrix} = \mathbf{J}_{\mathbf{f}_{\text{spec}}} \begin{bmatrix} \Delta l_1 \\ \Delta l_2 \\ \Delta l_3 \end{bmatrix}$$

Therefore

$$\mathbf{J}_{\mathbf{f}_{\text{spec}}} = \begin{array}{c} \begin{array}{ccc} l_1 & l_2 & l_3 \end{array} \\ \begin{bmatrix} \dfrac{\partial k}{\partial l_1} & \dfrac{\partial k}{\partial l_1} & \dfrac{\partial k}{\partial l_1} \\ \dfrac{\partial \phi}{\partial l_2} & \dfrac{\partial \phi}{\partial l_2} & \dfrac{\partial \phi}{\partial l_2} \\ \dfrac{\partial l}{\partial l_3} & \dfrac{\partial l}{\partial l_3} & \dfrac{\partial l}{\partial l_3} \end{bmatrix} \end{array} \begin{array}{l} k = f_{1spec}(l_1, l_2, l_3) \\ \phi = f_{2spec}(l_1, l_2, l_3) \\ l = f_{3spec}(l_1, l_2, l_3) \end{array}$$

$$\Rightarrow \begin{bmatrix} \Delta l_1 \\ \Delta l_2 \\ \Delta l_3 \end{bmatrix} = \mathbf{J}^{-1}_{\mathbf{f}_{\text{spec}}} J^{-1}_{ind} \begin{bmatrix} \Delta p_x \\ \Delta p_y \\ \Delta p_z \end{bmatrix}$$

Application case of the IKM for the configuration space / task space mapping

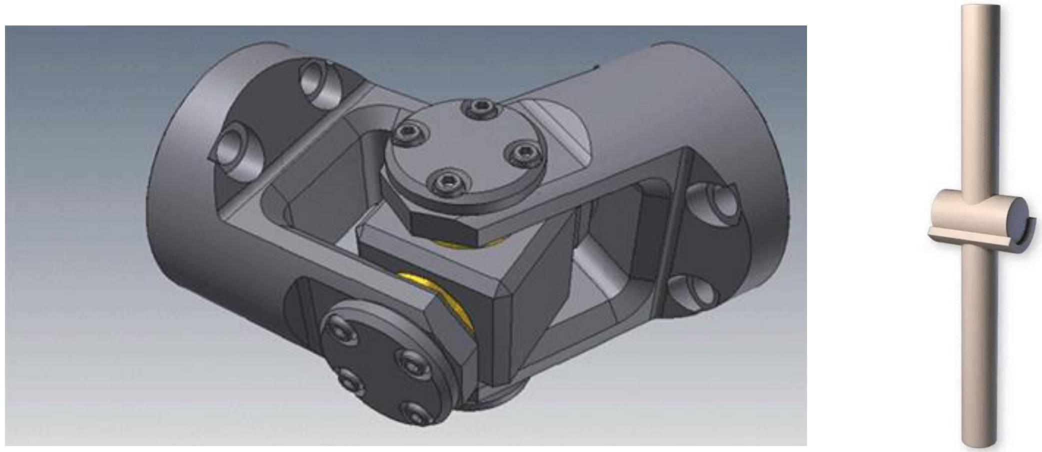


Illustration of Revolute joints

$$\vec{V}_{p\in C_n/C_0} = \vec{V}_{p\in C_n/C_{n-1}} + \vec{V}_{p\in C_{n-1}/C_{n-2}} + \cdots + \vec{V}_{p\in C_1/C_0}$$

$$\left.\frac{d\overrightarrow{Op}}{dt}\right|_{\mathbb{R}_0} = \dot{q}_1\vec{z}_0 \times \overrightarrow{O_0p} + \dot{q}_2\vec{z}_1 \times \overrightarrow{O_1p} + \cdots + \dot{q}_n\vec{z}_{n-1} \times \overrightarrow{O_{n-1}p}$$

$$\left.\frac{d\overrightarrow{Op}}{dt}\right|_{\mathbb{R}_0} = \underbrace{\left[\vec{z}_0 \times \overrightarrow{O_0p} \quad \vec{z}_1 \times \overrightarrow{O_1p} \quad \cdots \quad \vec{z}_{n-1} \times \overrightarrow{O_{n-1}p}\right]}\left[\dot{q}_1 \quad \cdots \quad \dot{q}_n\right]^T$$

$\mathbf{J}_{v-\mathbf{f}_{\mathbf{spec}}}$: **Linear velocity jacobian**

Recall: Revolute Joints

$$\vec{V}_{p\in C_m/C_{m-1}} = \left.\frac{d\overrightarrow{O_{m-1}p}}{dt}\right|_{C_0} = \vec{\omega}_{C_n/C_{m-1}} \times \overrightarrow{O_{m-1}p}$$

$$\vec{\omega}_{C_n/C_0} = \vec{\omega}_{C_n/C_{n-1}} + \vec{\omega}_{C_{n-1}/C_{n-2}} + \cdots + \vec{\omega}_{C_1/C_0}$$

$$\vec{\omega}_{C_n/C_0} = \dot{q}_1\vec{z}_0 + \dot{q}_2\vec{z}_1 + \cdots + \dot{q}_n\vec{z}_{n-1}$$

$$\vec{\omega}_{C_n/C_0} = \underbrace{\left[\vec{z}_0 \quad \vec{z}_1 \quad \cdots \quad \vec{z}_{n-1}\right]}\left[\dot{q}_1 \quad \cdots \quad \dot{q}_n\right]^T$$

$\mathbf{J}_{\omega-\mathbf{f}_{\mathbf{spec}}}$: **Angular velocity jacobian**

Application case of the IKM for the configuration space / task space mapping

Recall: Prismatic Joints

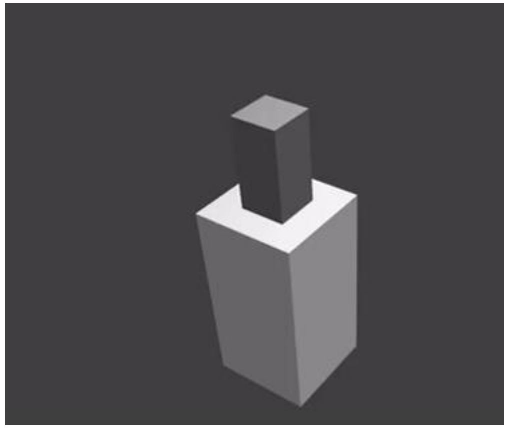


Illustration of Prismatic joints

$$\vec{V}_{p\in C_m/C_{m-1}} = \left.\frac{d\overrightarrow{O_{m-1}p}}{dt}\right|_{C_0} = \dot{d}_m \vec{z}_{m-1}$$

$$\left.\frac{d\overrightarrow{Op}}{dt}\right|_{\mathbb{R}_0} = \underbrace{[\vec{z}_0 \;\; \vec{z}_1 \;\; \cdots \;\; \vec{z}_{n-1}]}_{}[\dot{d}_1 \;\; \cdots \;\; \dot{d}_n]^T$$

$\mathbf{J}_{v-\mathrm{f_{spec}}}$: **Linear velocity jacobian**

$$\vec{\omega}_{C_n/C_0} = \underbrace{[0 \;\; 0 \cdots \;\; 0]}_{}[\dot{d}_1 \;\; \cdots \;\; \dot{d}_n]^T$$

$\mathbf{J}_{\omega-\mathrm{f_{spec}}}$: **Angular velocity jacobian**

Elastic based modeling

Kinematics is about the movement of bodies regardless of the forces/torques that cause the movement.

Objectives

- Geometric representation of the robot (Inverse kinematic modeling or IKM)
- Position of the end-Effector (Forward Kinematic modeling or FKM)

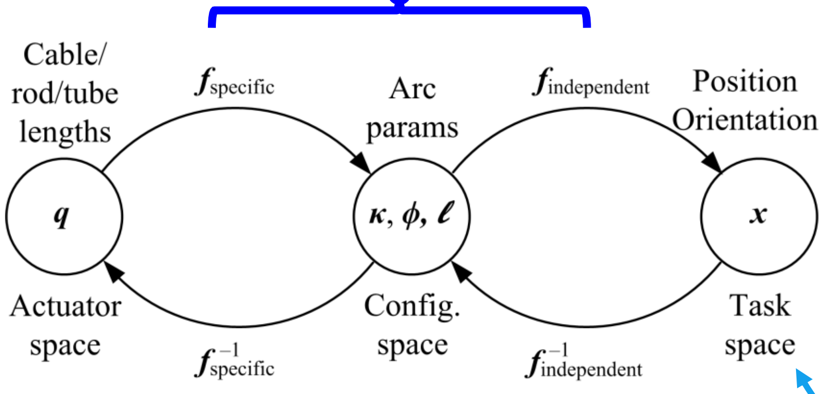
Forward Kinematic Modeling (FKM)

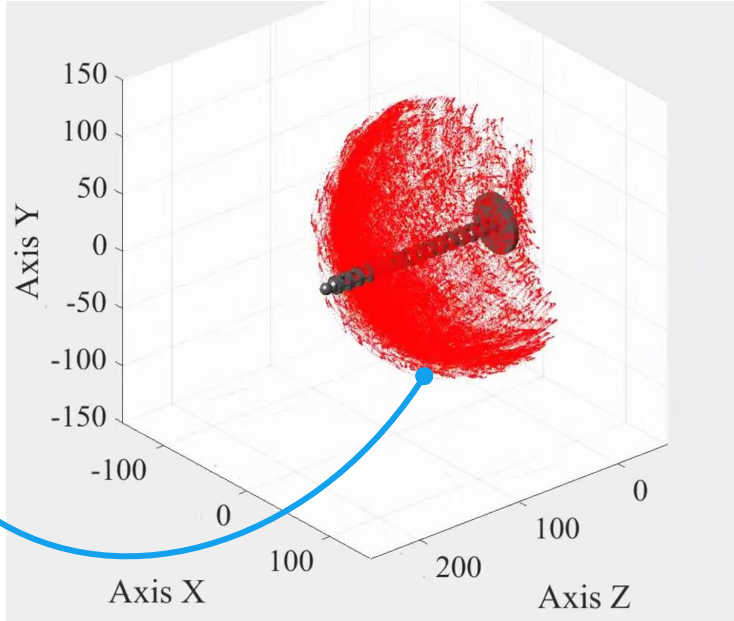
FKM

$$x = f_{ind}(\kappa, \phi, l)$$

FKM

$$\kappa, \phi, l = f_s(q)$$





Inverse Kinematic Modeling (IKM)

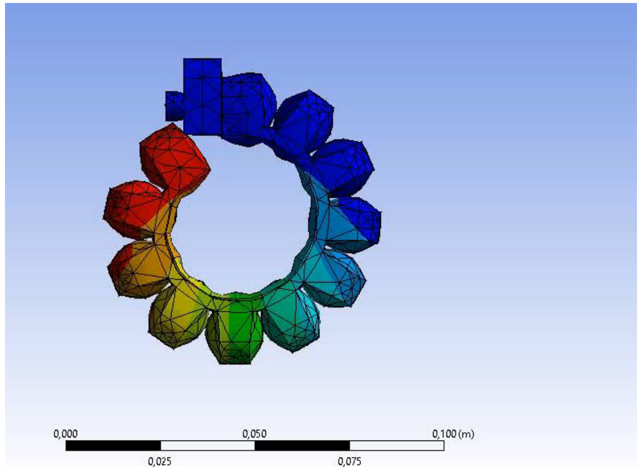
Elastic based modeling

Finite Elements Methods (FEM): ANSYS Modeling

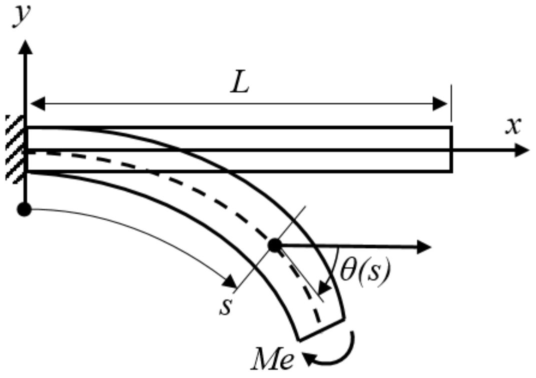
1. Open ANSYS WorkBench
2. Import the Material properties in Engineering Data
3. Import the geometry of the soft robot
4. Define the simulation environment
5. Compute and analyse the results

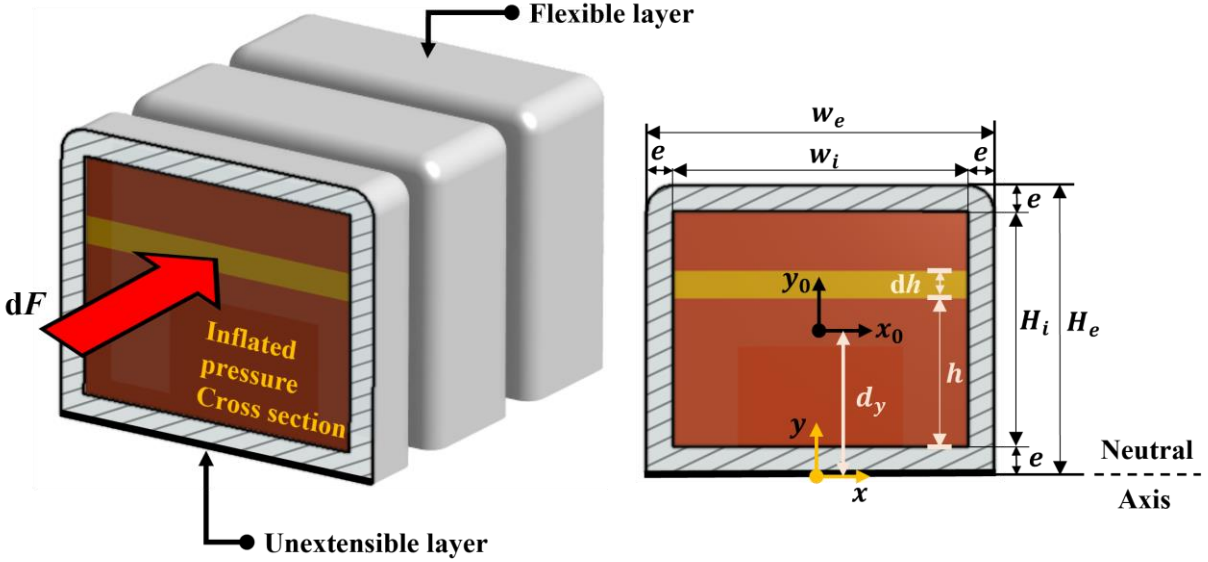
Elastic based modeling

Euler Bernoulli modeling Approch



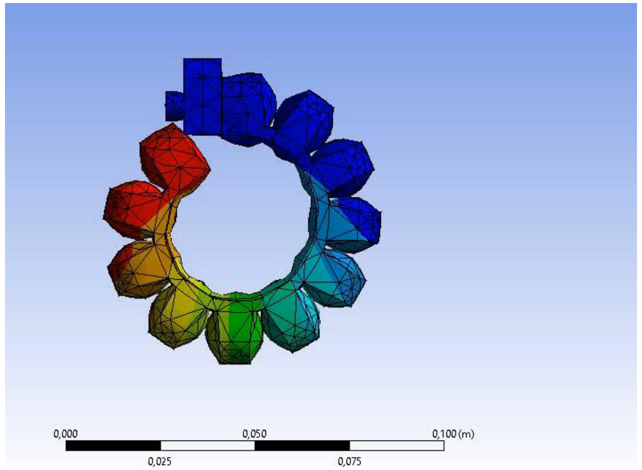
Physical actuation



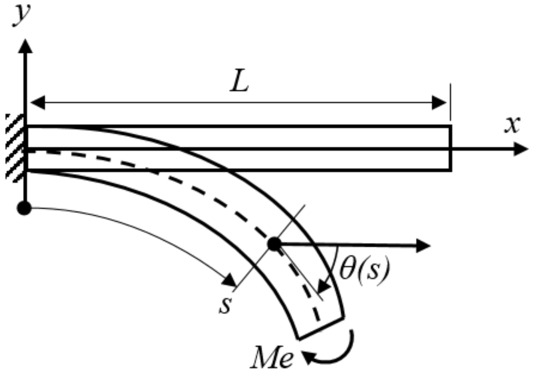


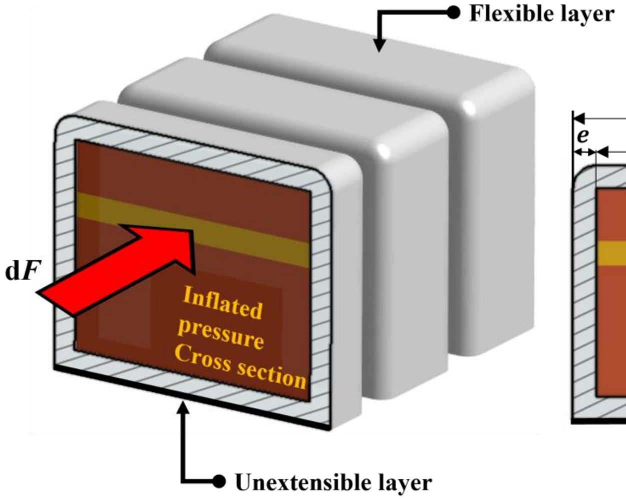
Elastic based modeling

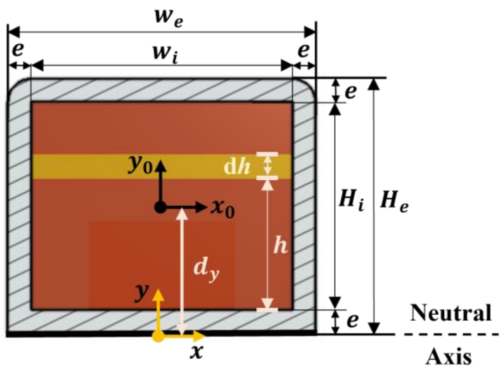
Euler Bernoulli modeling Approch



Pressure







Let define

$A = a^2$ The area of the cross section

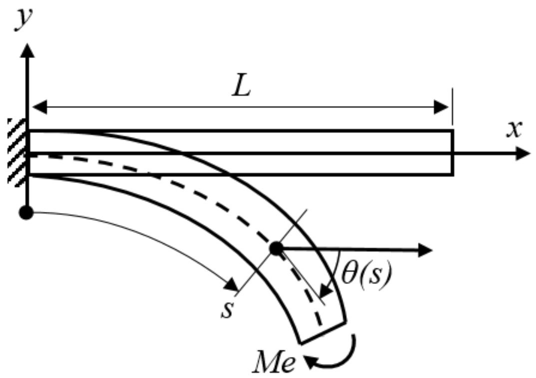
$b = w_i$, The width of the cross section

$h = H_i$ The heigth of the cross section

$v = \sqrt{\frac{b}{h}}$ The size ratio of the actuator

$$\begin{cases} dF = (va)P_{in}dh \\ dM_e = hdF \end{cases}$$

$$\{M_e = \Psi_e P_{in}$$



Elastic based modeling

Euler Bernoulli modeling Approch (Planar case)

One section modeling

Boundary condition

$$\theta(s=0)=0$$
$$M(s=L)=Me$$

$$x(s=0)=0$$
$$y(s=0)=0$$

$$\kappa(s)=\frac{Me}{EI}$$

$$\theta(s)=\frac{Me}{EI}s$$

$$\begin{cases} x' = \cos\theta(s) \\ y' = \sin\theta(s) \end{cases}$$

$$\begin{cases} x_1 = \dfrac{(EI)_1}{M_1}\sin\dfrac{M_1}{(EI)_1}s_1 \\ y_1 = \dfrac{(EI)_1}{M_1}\left(1-\cos\dfrac{M_1}{(EI)_1}s_1\right) \end{cases}$$

Two sections modeling $[L_1 + L_2 = L]$

First section modeling

Boundary condition

$$\theta_1(s_1=0)=0$$
$$M(s_1=L_1)=M_1$$

$$x_1(s_1=0)=0$$
$$y_1(s_1=0)=0$$

$$\begin{cases} x_1 = \dfrac{(EI)_1}{M_1}\sin\dfrac{M_1}{(EI)_1}s_1 \\ y_1 = \dfrac{(EI)_1}{M_1}\left(1-\cos\dfrac{M_1}{(EI)_1}s_1\right) \end{cases}$$

Second section modeling

Boundary condition

$$\theta(s_2=0)=\theta_1(s_1=L_1)$$
$$M(s_2=L_2)=M_2$$

$$x_2(s_2=0)=x_1(s_1=L_1)$$
$$y_2(s_2=0)=y_1(s_1=L_1)$$

$$\begin{cases} x_2 = \dfrac{(EI)_2}{M_2}\sin\left[\dfrac{M_2}{(EI)_2}s_2+\theta_1(L_1)\right]+x_1(L_1) \\ y_2 = \dfrac{(EI)_2}{M_2}\left(\cos\theta_1(L_1)-\cos\left[\dfrac{M_2}{(EI)_2}s_2+\theta_1(L_1)\right]\right)+y_1(L_1) \end{cases}$$

1

Planar Kinematics of a one section-soft finger

2

```
function [Pe]= FKM1 (P1)

        % Finger material and geometry properties (in USI)

E = 0.36e6;
rho = 1140;                     %masse volumique[k/m-3]
vol1 = 6.1667e-5;               %volume [m^3]
g = 9.81;
masse1 = 7.0917e-2;

        % Finger Geometry (in USI)
L1 = 87e-3;                     % Finger length [m]
e1 = 3e-3;                      %[m] Thickness
h1 = 27e-3;                     %[m] Cross section heigth
B1 = 40e-3 ;                    %[m] Air chamber width
H1 = (30e-3)+e1 ;               %[m] Air chamber total heigth
t1 = 3e-3;                      %[m] Rigid layer thickness
L1 = 87e-3;                     %longueur [m]
b1 = B1-2*e1;                   %[m] Cross section width
d12 = h1/2+ e1/2;               %[m] CoG distance from the base
S11z= B1*H1 - b1*h1;            %[m ^2] area of the material

I11 = ((B1*H1^3)-(b1*h1^3))/12; %[m ^4] Quadratic moment wrt CoG
I11z = I11 + (S11z*d12^2);      %[m ^4] Quadratic moment wrt Base
```

```
% Quadratic moment of the chamber link
h1c = 11.5e-3;          %Cross section heigth
b1c = 19e-3;            %Cross section width

H1c = 18e-3;            %Cross section heigth
B1c = 25e-3;            %Cross section width


d2 = 0.5*(h1c+t1);      %[m] CoG distance from the base
S12z= H1c*B1c-h1c*b1c;

I12=(B1c *(H1c)^3-(b1c)*(h1c)^3)/12;        %[m ^4]
I12z = I12 + (S12z*(d2)^2);                 %[m ^4]

ratio1s=4*S11z/(4*S11z+3*S12z);
ratio2s=3*S12z/(4*S11z+3*S12z);

% Mean Quadratic moment of the whole finger

%Im1 = (4*I11z + 3*I12z)/7;                 %[m ^4]
Im1 = (ratio1s *I11z + ratio1s *I12z);      %[m ^4]

Ioz1=Im1;
```

3

```
%% ------- Calculation of the End bending moment1 -----------------
% s = area section of the finger, r = shape factor

r1 = sqrt(b1/h1);
s1 = b1*h1;
a1 = sqrt(s1);

Mo1 = P1*((a1^2)*((0.5*a1+(t1+e1)*r1)/r1)); %[N.m]
M01 = (Mo1 + (L1/2)*(masse1)*g); %[N.m]

%%%%%%%%%%%%%%%%

M0=M01;
```

4

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=400;
Xe1=[];
Ye1=[];

for coord=1:n
i= (coord-1)*L1/(n-1);
X(coord) = (((E+P1*r01)*Ioz1)/M0)*sin((M0/((E+P1*r01)*Ioz1))*i);          % [m]
Y(coord) = -(((E+P1*r01)*Ioz1)/M0)*(1-cos((M0/((E+P1*r01)*Ioz1))*i));     % [m]
End

Pe=[X' Y'];

figure;
title('S-hape Kinematics (Elastic modeling –Euler Bernoulli kinematics')
hold on,
plot(Pe(:,1),Pe(:,2),'.');
grid on,
```

Planar Kinematics of a two sections-soft finger

1

```
function [Pe]= FKM2(P1,P2)

        % Finger material and geometry properties of Section N°2 (in USI)

E = 0.36e6;
rho = 1140;                          %masse volumique[k/m-3]
vol1 = 6.1667e-5;                    %volume [m^3]
g = 9.81;
masse1 = 7.0917e-2;
t = 3e-3;                            %[m] Rigid layer thickness

        % Finger Geometry (in USI)
L2 = 56e-3;                          % Finger length [m]
e2 = 2.5e-3;                                  %[m] Thickness
h2 = 17e-3;                          %[m] Cross section heigth
B2 = 20e-3 ;                         %[m] Air chamber width
H2 = 20;                             %[m] Air chamber total heigth
b2 = B2-2*e2;                        %[m] Cross section width
d2 = h2/2+ e2/2;                     %[m] CoG distance from the base
S21z= B2*H2 – b2*h2;                 %[m ^2] area of the material


I21 = ((B2*H2^3)-(b2*h2^3))/12;      %[m ^4] Quadratic moment wrt CoG
I21z = I21 + (S21z*d21^2);           %[m ^4] Quadratic moment wrt Base
```

2

```
% Quadratic moment of the chamber link
h2c = 3e-3;                 %Cross section heigth
b2c = 2e-3;                 %Cross section width

H2c = 5e-3;                 %Cross section heigth
B2c = 10e-3;                %Cross section width


d2c = 0.5*(h2c+t);          %[m] CoG distance from the base
S22z= H2c*B2c-h2c*b2c;

I22=(B2c *(H2c)^3-(b2c)*(h2c)^3)/12;          %[m ^4]
I22z = I22 + (S22z*(d2c)^2);                  %[m ^4]

ratio1s=4*S21z/(4*S21z+3*S22z);
ratio2s=3*S22z/(4*S21z+3*S22z);

% Mean Quadratic moment of the whole finger

%Im1 = (4*I21z + 3*I22z)/7;                   %[m ^4]
Im1 = (ratio21s *I21z + ratio22s *I22z);      %[m ^4]

Ioz1=Im1;
```

1

Planar Kinematics of a two sections-soft finger

2

```
.
.
        % Finger material and geometry properties of Section N°1 (in USI)

E = 0.36e6;
rho = 1140;                         %masse volumique[k/m-3]
vol1 = 6.1667e-5;                   %volume [m^3]
g = 9.81;
masse1 = 7.0917e-2;
t = 3e-3;                           %[m] Rigid layer thickness

        % Finger Geometry (in USI)
L1 = 63e-3;                         % Finger length [m]
e1 = 3e-3;                          %[m] Thickness
h1 = 17e-3;                         %[m] Cross section heigth
B1 = 40e-3 ;                        %[m] Air chamber width
H1 = 30e-3;                         %[m] Air chamber total heigth
b1 = B1-2*e1;                       %[m] Cross section width
d1 = 0.5*(h1+ t);                   %[m] CoG distance from the base
S11z= B1*H1 – b1*h1;                %[m ^2] area of the material


I11 = ((B1*H1^3)-(b1*h1^3))/12;     %[m ^4] Quadratic moment wrt CoG
I11z = I11 + (S11z*d1^2);           %[m ^4] Quadratic moment wrt Base
```

```
% Quadratic moment of the chamber link
h1c = 5e-3;             %Cross section heigth
b1c = 6e-3;             %Cross section width

H1c = 10e-3;            %Cross section heigth
B1c = 20e-3;            %Cross section width


d1c = 0.5*(h2c+t);      %[m] CoG distance from the base
S12z= H1c*B1c-h1c*b1c;

I12=(B1c *(H1c)^3-(b1c)*(h1c)^3)/12;            %[m ^4]
I12z = I12 + (S12z*(d2)^2);                     %[m ^4]

ratio1s=3*S11z/(3*S11z+2*S12z);
ratio2s=2*S12z/(3*S11z+2*S12z);

% Mean Quadratic moment of the whole finger

%Im1 = (3*I11z + 2*I12z)/5;                     %[m ^4]
Im1 = (ratio1s *I11z + ratio1s *I12z);          %[m ^4]

Ioz1=Im1;
```

4

```
%% ------- Calculation of the End bending moment M2 --------------
% s = area section of the finger, r = shape factor

r2 = sqrt(b2/h2);
s2 = b2*h2;
a2 = sqrt(s2);

Mo2 = P2*((a2^2)*((0.5*a2+(t+e2)*r2)/r2));       %[N.m]
M02 = (Mo2 + (L2/2)*(masse1)*g);                 %[N.m]

%%%%%%%%%%%%%%%%

%% ------- Calculation of the End bending moment M1 --------------
% s = area section of the finger, r = shape factor

r1 = sqrt(b1/h1);
s1 = b1*h1;
a1 = sqrt(s1);

Mo1 = P1*((a1^2)*((0.5*a1+(t+e1)*r1)/r1));       %[N.m]

M01 = (Mo1 + (L1/2)*(masse1)*g) + M02;           %[N.m]
```

5

```
%% ------- Posture calulation first portion ----------------
n=400; Xe1=[];Ye1=[];
x_L1 = ((E*Ioz1)/M01)*sin((M01/(E*Ioz1))*L1);                  % [m]
y_L1 = -((E*Ioz1)/M01)*(1-cos((M01/(E*Ioz1))*L1));             % [m]

for coord=1:n
i= (coord-1)*L1/(n-1);
X(coord) = ((E*Ioz1)/M0)*sin((M0/(E*Ioz1))*i);                % [m]
Y(coord) = -((E*Ioz1)/M0)*(1-cos((M0/(E*Ioz1))*i));           % [m]
End
Pe=[X' Y'];

%% ------- Posture calulation Second branche ----------------
Xe2=[]; Ye2=[];

for coord2=1:n
ii= (coord2-1)*L2/(n-1);
Xe2(coord2) = (((E+P2*r02)*Ioz2)/M02)*(sin((M02/((E+P2*r02)*Ioz2))*ii + …
…+Theta0)-sin(Theta0)) + x_L1;                                 % [m]
Ye2(coord2) = -(((E+P2*r02)*Ioz2)/M02)*(cos(Theta0) -
…cos((M02/((E+P2*r02)*Ioz2))*ii + Theta0)) + y_L1;             % [m]
end
```

6

```
figure;
title('Shape control validation (PH-EB/Ansys Shape')
hold on,
plot(Pe1(:,1),Pe1(:,2),'.');
hold on;
plot(Pe2(:,1),Pe2(:,2),'.');
```

Why dynamics ?

To define actuator inputs allowing the robot to perform pratically the motion defined by the kinematics analysis while interacting with the external

Euler Lagrange approach

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = [\mathcal{F}]$$

$$q = [q_1 \quad q_2 \quad d_3 \quad q_4]$$

$\boldsymbol{q}$ is the actuator variable:

$\boldsymbol{\mathcal{L}}$ is the lagrangian equation of the motion, define as follow:

$$\boldsymbol{\mathcal{L}(\dot{q}, q) = \mathrm{K_e}(\dot{q}, q) - \mathrm{P_e}(q)}$$

$\boldsymbol{K_e}$ is the kinetic energy of the robot

$\boldsymbol{P_e}$ the potential energy of the robot

Dynamic parameters

Elements	q(rad)	m(kg)	I_{i/Δ_G}	$\ddot{q}$(rad/s^{-2})	$\dot{q}$(rad/s^{-1})
i	$\boldsymbol{q}_i$	m_i	$\boldsymbol{I}_{i_\Delta}$	$\ddot{\boldsymbol{q}}_i$	$\dot{q}_i$

$$\boldsymbol{D(q)\ddot{q} + C(\dot{q}, q)\dot{q} + G(q) = \mathcal{F}}$$

$\boldsymbol{D(q)}$ is the inertia matrix

$\boldsymbol{C(\dot{q}, q)}$ is the matrix of coriolis and centrigugal effects

$\boldsymbol{G(q)}$ is the matrix of gravity effects

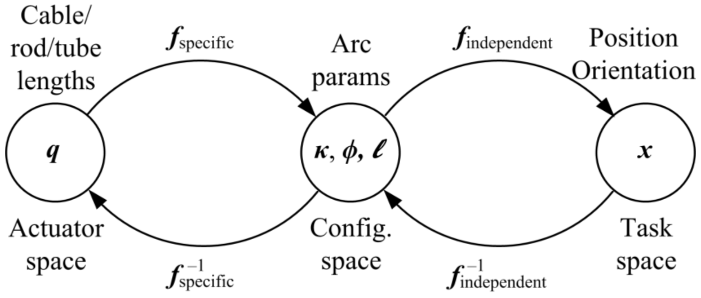
$\boldsymbol{\mathcal{F}}$ is the generalized forces at joints

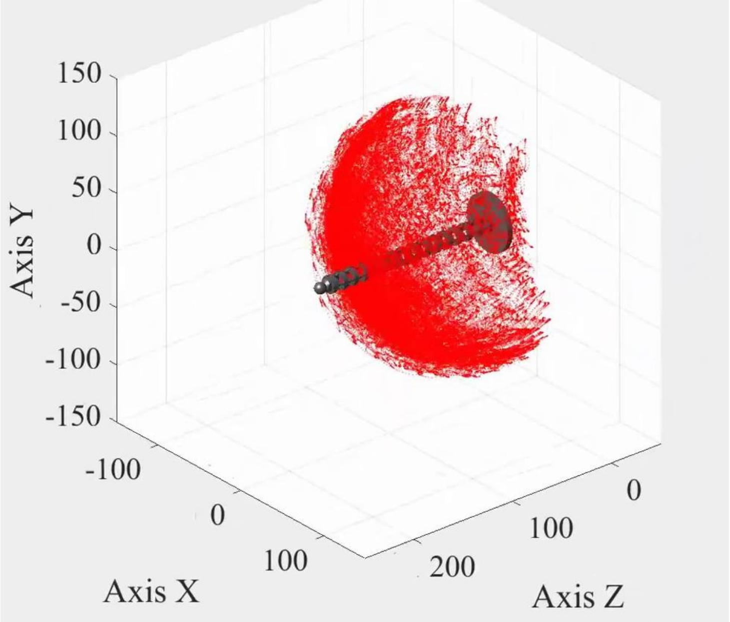
$D(q)$ is the inertia matrix

$C(\dot{q}, q)$ is the matrix of coriolis and centrigugal effects

$G(q)$ is the matrix of gravity effects

$$D(q)\ddot{q} + C(\dot{q}, q)\dot{q} + G(q) = \tau$$





$$D(q) = \sum_{1}^{n} \overbrace{m_i \mathrm{J}_{v_i}^T \mathrm{J}_{v_i}}^{\text{Linear motion}} + \underbrace{\sum_{1}^{n} \mathrm{J}_{\omega_i}^T \left[R_i^0 \mathrm{I}_i \left(R_i^0\right)^T\right] \mathrm{J}_{\omega_i}}_{\text{Rotational motion}}$$

How?

$$\frac{1}{2} m_i \dot{x}_i^2 = \frac{1}{2} m_i \, \dot{x}_i^T \dot{x}_i$$

$$\dot{x}_i = \mathrm{J}_{v_i} \dot{q}_i \qquad \frac{1}{2} m_i \dot{x}_i^2 = \frac{1}{2} m_i \mathrm{J}_{v_i}^T \underbrace{\dot{q}_i^T \dot{q}_i}_{\dot{q}_i^2} \mathrm{J}_{v_i} \Rightarrow \frac{1}{2} m_i \dot{x}_i^2 = \frac{1}{2} m_i \mathrm{J}_{v_i}^T \mathrm{J}_{v_i} \dot{q}_i^2$$

Please retrieve the case of the rotional motion

$$D(q) = \begin{bmatrix} d_{11} & . & . \\ d_{12} & . & d_{42} \\ . & . & . \\ . & . & d_{44} \end{bmatrix} \qquad C(\dot{q}, q) = \begin{bmatrix} C_{11} & . & . \\ C_{12} & . & C_{42} \\ . & . & . \\ . & . & C_{44} \end{bmatrix}$$

$$\boxed{C_{kj}(q, \dot{q}) = \sum_{1}^{n} c_{ijk}(q)\dot{q}_i \qquad \underbrace{c_{ijk} = \frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right\}}_{\text{Christoffel symbol}}}$$

$$C = \begin{Bmatrix} C_{11}(q,\dot{q}) = \sum_{1}^{4} c_{i11}(q)\dot{q}_i & C_{12}(q,\dot{q}) = \sum_{1}^{4} c_{i21}(q)\dot{q}_i & C_{13}(q,\dot{q}) = \sum_{1}^{4} c_{i31}(q)\dot{q}_i & C_{14}(q,\dot{q}) = \sum_{1}^{4} c_{i41}(q)\dot{q}_i \\ C_{21}(q,\dot{q}) = \sum_{1}^{4} c_{i12}(q)\dot{q}_i & C_{22}(q,\dot{q}) = \sum_{1}^{4} c_{i22}(q)\dot{q}_i & C_{23}(q,\dot{q}) = \sum_{1}^{4} c_{i32}(q)\dot{q}_i & C_{24}(q,\dot{q}) = \sum_{1}^{4} c_{i42}(q)\dot{q}_i \\ C_{31}(q,\dot{q}) = \sum_{1}^{4} c_{i13}(q)\dot{q}_i & C_{32}(q,\dot{q}) = \sum_{1}^{4} c_{i23}(q)\dot{q}_i & C_{33}(q,\dot{q}) = \sum_{1}^{4} c_{i33}(q)\dot{q}_i & C_{34}(q,\dot{q}) = \sum_{1}^{4} c_{i43}(q)\dot{q}_i \\ C_{41}(q,\dot{q}) = \sum_{1}^{4} c_{i14}(q)\dot{q}_i & C_{42}(q,\dot{q}) = \sum_{1}^{4} c_{i24}(q)\dot{q}_i & C_{43}(q,\dot{q}) = \sum_{1}^{4} c_{i34}(q)\dot{q}_i & C_{44}(q,\dot{q}) = \sum_{1}^{4} c_{i44}(q)\dot{q}_i \end{Bmatrix}$$

The Potential energy $\mathbf{P_e(q)}$

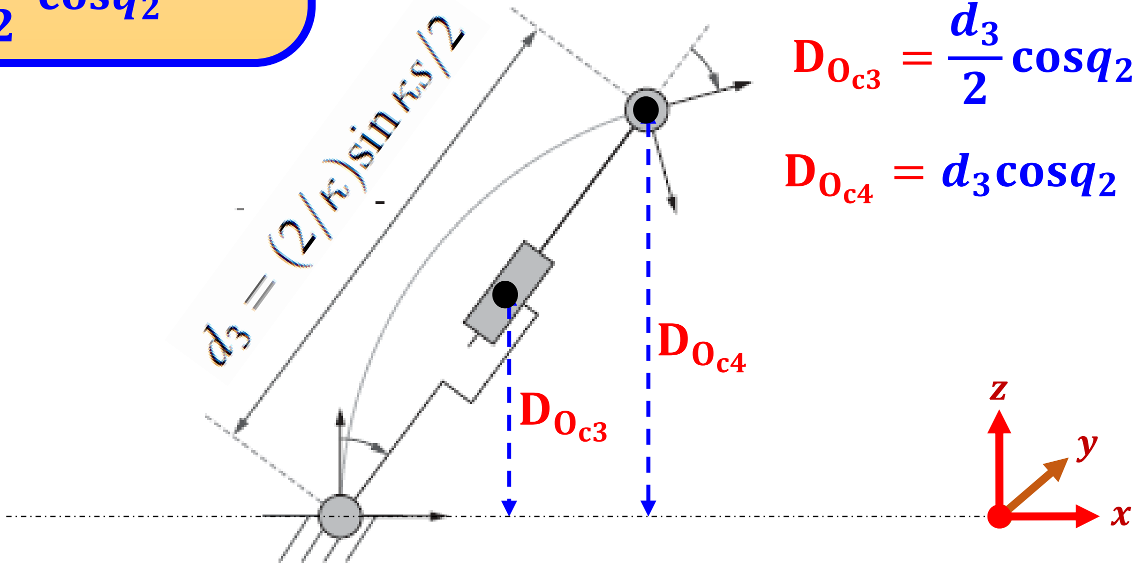
$$\mathbf{P_e(q)} = \mathbf{m_1 \cdot g \cdot 0} + \boldsymbol{m_2} \cdot \mathbf{g \cdot 0} + \mathbf{m_3 \cdot g \cdot D_{O_{c3}}} + \mathbf{m_4 \cdot g \cdot D_{O_{c4}}}$$

$$q = [q_1 \quad q_2 \quad d_3 \quad q_4]$$
$$q_1 = \boldsymbol{\phi}$$
$$q_2 = \frac{\boldsymbol{\kappa l}}{\mathbf{2}}$$

In T_0^4,
$$\mathbf{P_{4z}} = \boldsymbol{d_3 \cos q_2}$$

In T_0^3,
$$\mathbf{P_{3z}} = \frac{\boldsymbol{d_3}}{\mathbf{2}} \boldsymbol{\cos q_2}$$

$$\boldsymbol{G}(\mathbf{q}) = \begin{bmatrix} \frac{\partial \mathbf{P_e}}{\partial q_1} \\ \frac{\partial \mathbf{P_e}}{\partial q_2} \\ \frac{\partial \mathbf{P_e}}{\partial q_3} \\ \frac{\partial \mathbf{P_e}}{\partial q_4} \end{bmatrix} = \mathbf{g} \cdot \begin{bmatrix} \mathbf{0} \\ -\mathbf{m_3} \frac{\boldsymbol{d_3}}{\mathbf{2}} \sin \boldsymbol{q_2} - \mathbf{m_4} \boldsymbol{d_3} \sin \boldsymbol{q_2} \\ \frac{\mathbf{m_3}}{\mathbf{2}} \cos \boldsymbol{q_2} + \mathbf{m_4} \cos \boldsymbol{q_2} \\ \mathbf{0} \end{bmatrix}$$



$$\mathbf{D_{O_{c3}}} = \frac{\boldsymbol{d_3}}{\mathbf{2}} \boldsymbol{\cos q_2}$$

$$\mathbf{D_{O_{c4}}} = \boldsymbol{d_3 \cos q_2}$$

The efforts $\mathcal{F}$

The causality

The causes ⇄ The effects

The effects must read The causes

Torques create a rotations

Forces create a linear motions

$$\mathcal{F} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ F_3 \\ \tau_4 \end{bmatrix}$$

The Dynamics equations of the manipulators of the manipulator are given by

$$D(q)\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{d}_3 \\ \ddot{q}_4 \end{bmatrix} + C(\dot{q}, q)\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{d}_3 \\ \dot{q}_4 \end{bmatrix} + G(q) = \begin{bmatrix} \tau_1 \\ \tau_2 \\ F_3 \\ \tau_4 \end{bmatrix}$$