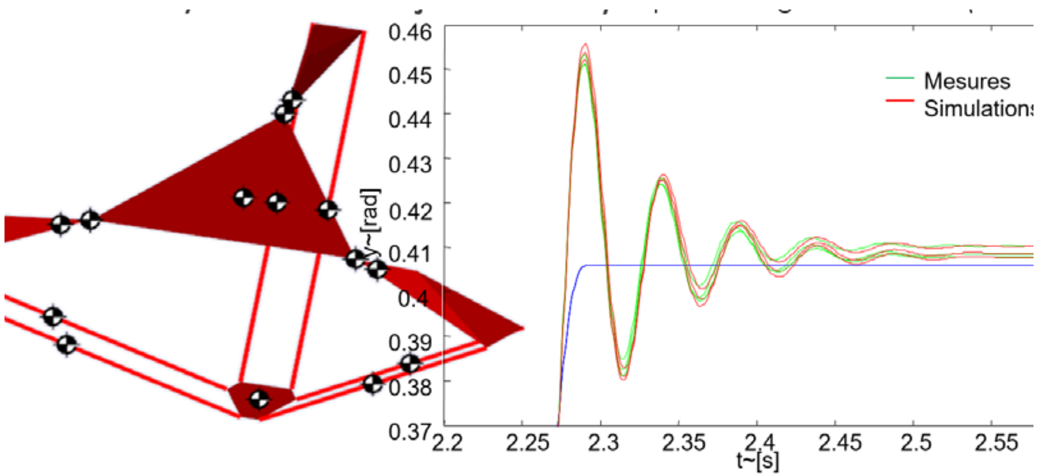
Modélisation dynamique de robots

Dr. Ing. Mohamed Bouri
IEEE Senior Member
EPFL, Laboratoire de Systèmes Robotiques

Part II



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Dynamics of robot manipulators

Organisation of the lecture

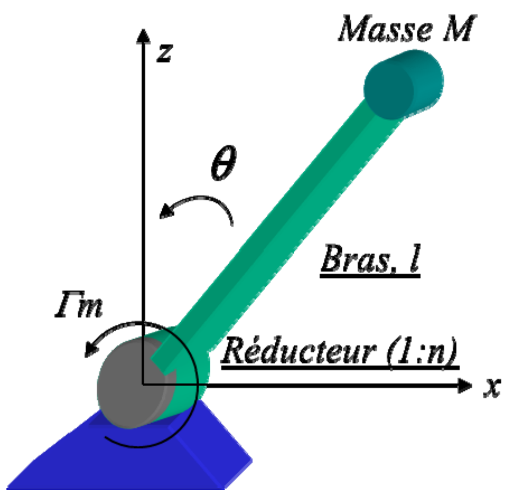
Part 1 – Definitions related to dynamic modelling

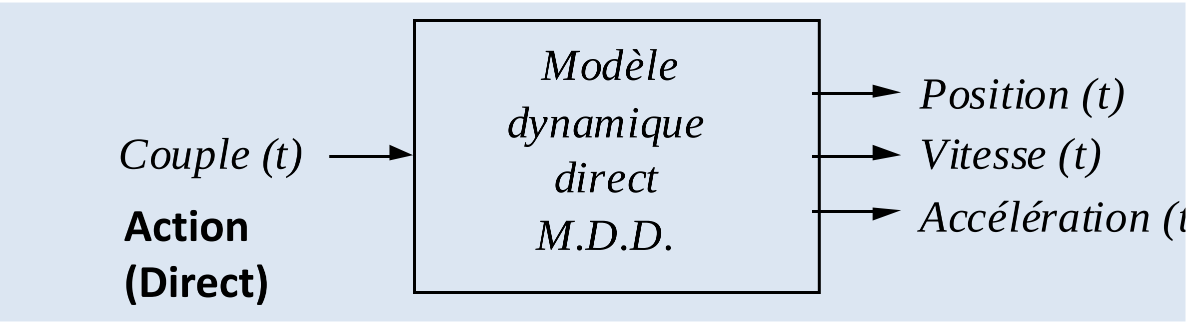
Part 2 – Dynamic models : Inverse and Direct, through example

Part 3 – Implementation

Part 4 – Development of the dynamic model
- Lagrange approach
- Newton-Euler

Modèle dynamique direct d'un axe de robot $\theta, \dot{\theta}, \ddot{\theta}$?





$$\theta(t) = f_D(\Gamma_m)$$

$$J_T\ddot{\theta} = n\Gamma_m - M_b g \frac{l}{2}\sin(\theta) - Mgl\sin(\theta) - k_{vis}\dot{\theta}$$

Pour trouver $\theta, \dot{\theta}, \ddot{\theta}$

Il faut :

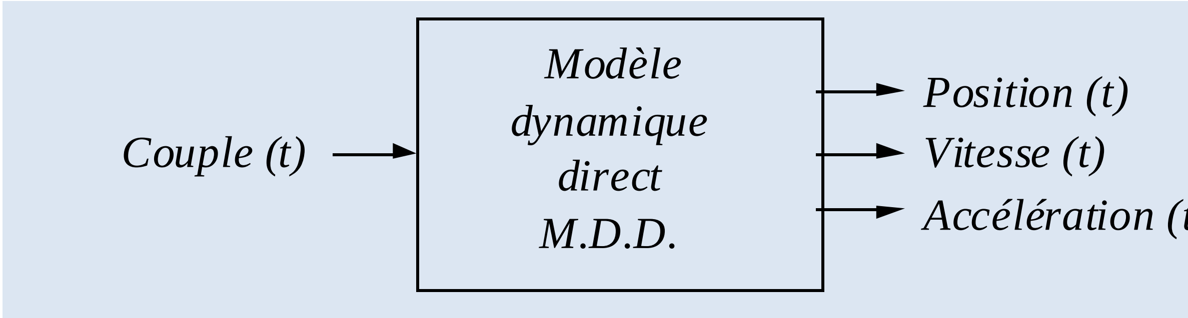
- Résoudre l'équation différentielle de second ordre
- Simuler le système de second ordre par blocs intégrateurs.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Modèle dynamique inverse d'un axe de robot

$$\Gamma_m(t) = f_I(\theta, \dot{\theta}, \ddot{\theta})$$

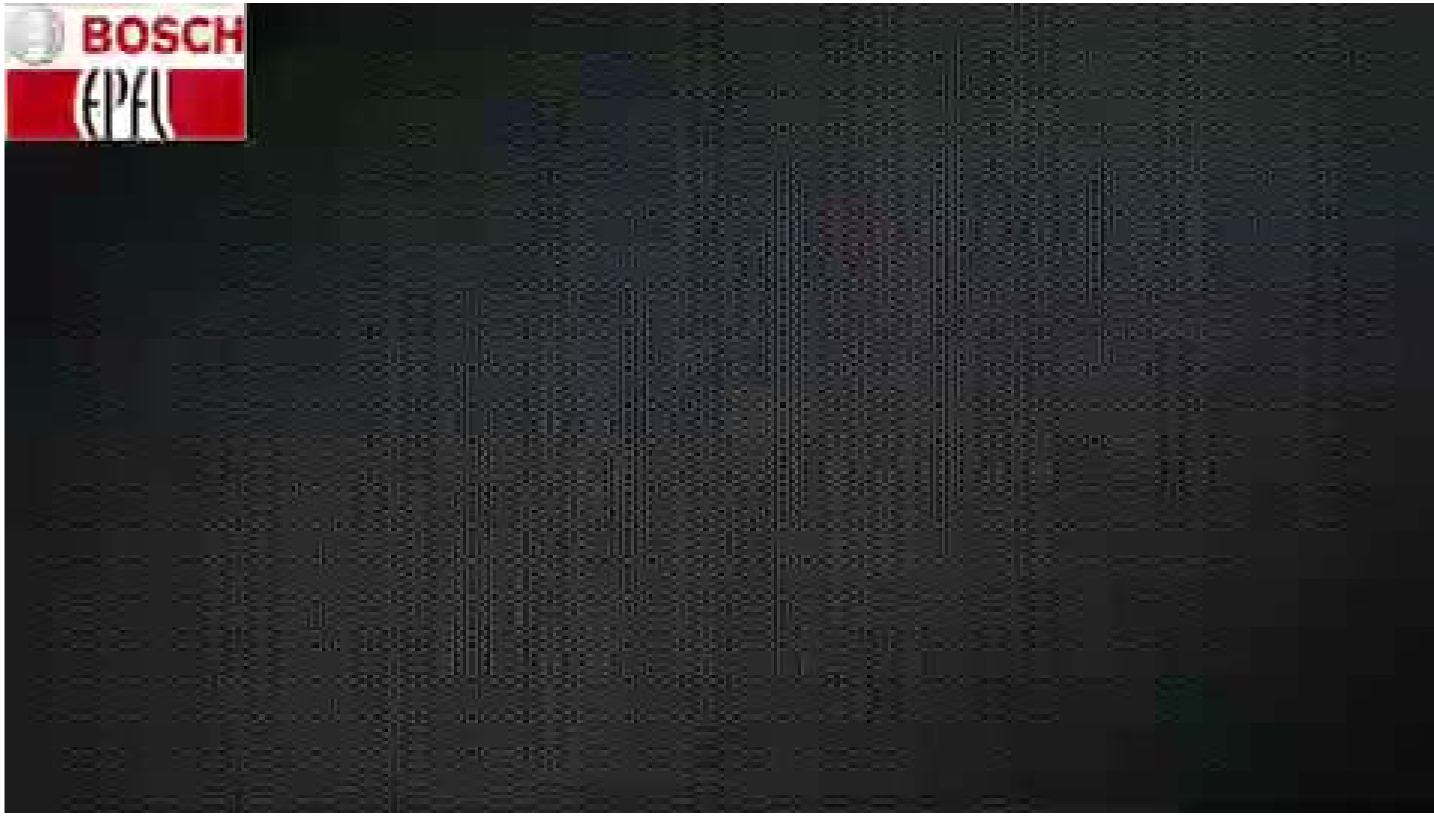
Equation Algébrique



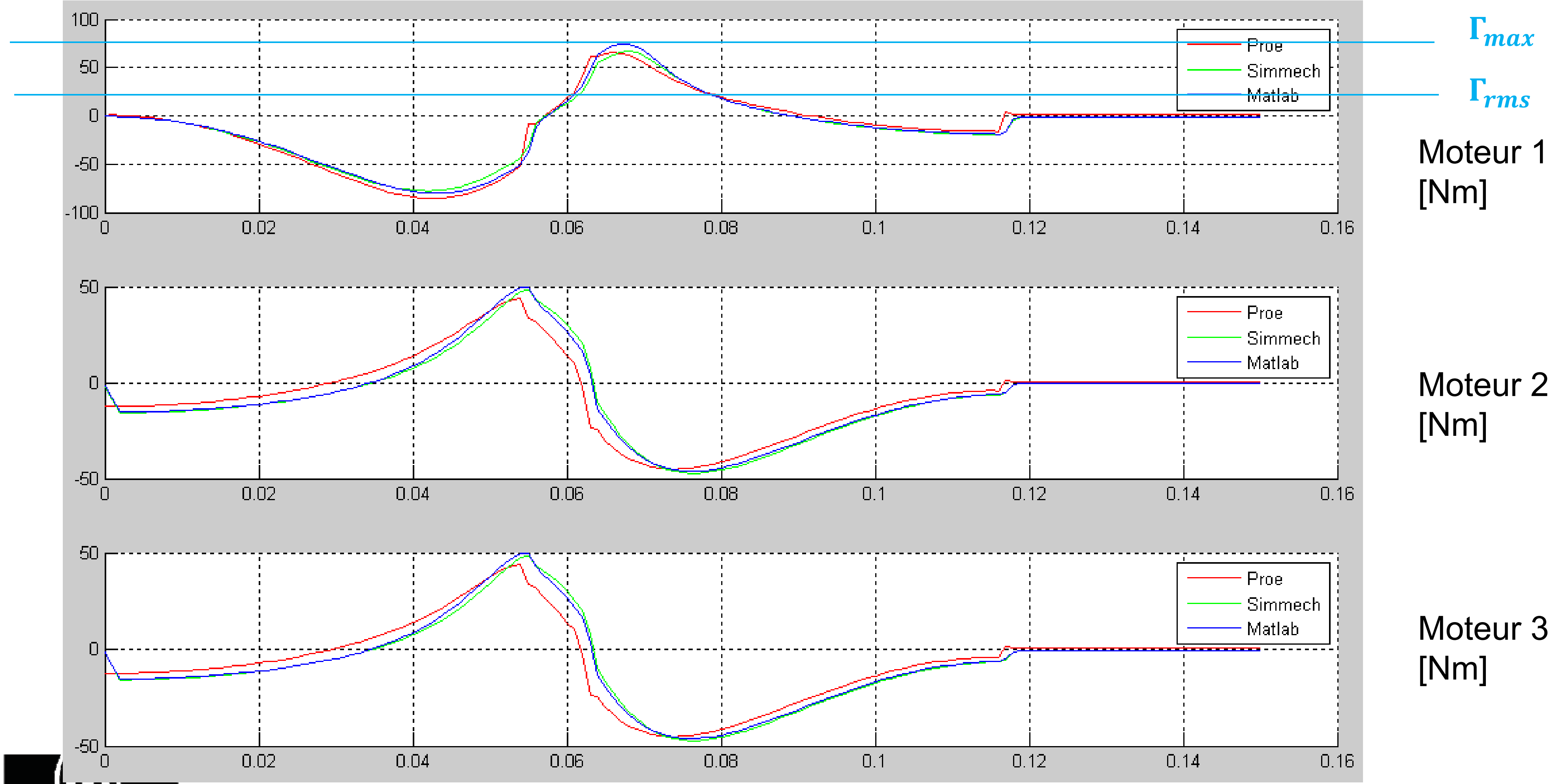
Pour les valeurs d'une trajectoire désirée ce couple s'appelle couple a priori

$$\Gamma_m = f_I(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)$$

Quels sont les couples nécessaires à la réalisation d'une trajectoire donnée ?

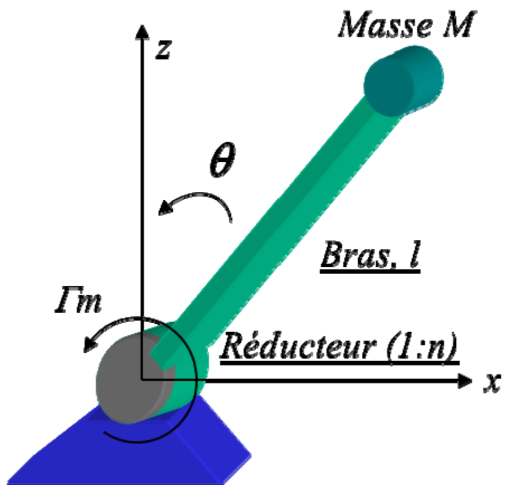


Exemples de couples dynamiques a priori suite à une trajectoire **<u>«Ellipse», 15g avec 1kg de charge</u>**



Exercise – 1 dof arm

Draw the Simulink direct and inverse dynamic models of the following robotic arm.



SetParams1Arm.m

Fichier Edition Format Affichage ?

```
Im = 10e-4;    %inertia of motor
Iarm = 150e-4; %inertia of arm

m = 1;
l = 0.25;
M = 1;

I_load = M * l *l;

Itot = Im + Iarm + I_load;

g = 10;

Amp = 5;
freq = 1;
w = 2*pi*freq;
```

Matlab Parameter File

Dynamics of robot manipulators

Organisation of the lecture

Part 1 – Definitions related to dynamic modelling ☑

Part 2 – Dynamic models : Inverse and Direct, through an example ☑

Part 3 – Implementation

- The general expression of each dynamic model may be given by :

$$\Gamma = B(q) \cdot \ddot{q} + G(q) + C(q,\dot{q}) + F(q,\dot{q}) + K(q,\dot{q})$$

where:

Γ	vector of the generalized couple
$q, \dot{q}, \ddot{q}$	vector of the generalized displacement and its temporal derivatives
$B(q)$	inertia matrix of robot depending on position
$G(q)$	vector due to gravity
$C(q,q)$	matrix/vector due to coriolis effect and centrifugal force
$F(q,\dot{q})$	matrix/vector due to friction
$K(q,\dot{q})$	matrix/vector with the stiffness of the robot depending on position an acceleration

- **Disturbances are not included** because they are not known a priori
- **Disturbances may be reduced/compensated** if they are measured before actuating the robot,
- Some disturbances can be statistical determined and a compensation is possible

Dynamic Model

implementation

Implementation

There is no unique solution

- Goals of an implementation:
  - Reduce computational cost
  - Optimisation through knowledge of materials used (Mflops, MIPS ...)
  - Knowledge of movement (precision, work cadence...)

- By doing:
  - Implement the formulas recursively/iteratively (implicit forms)
  - Re-use formulations on multiple processors at the same time
  - Create look-up tables

Implementation

- No need for real-time calculation of the matrices (lookup-table in memory)
- Sensors information allow a real-time calculation of the speed and acceleration vectors
- Number of calculations proportional to speed of displacement (interpolation / memory accesses)
- Size of lookup-table depends on number of discrete positions -> balance between size of memory and calculations

Implementation

- Possibility of simplification:
  - Eliminate non-contributing expressions
  - $\sin(\theta)=\theta$ and $\cos(\theta)=1$ for small θ
  - Omit small masses and inertias
  - Omit friction
  - Omit secondary affects of missing stiffness

- Simplification introduces errors which have to be compensated with an on-line regulator

- Good compromises lead to a good result

Dynamics of robot manipulators

Organisation of the lecture

Part 1 – Definitions related to dynamic modelling ✓

Part 2 – Dynamic models : Inverse and Direct, through example ✓

Part 3 – Implementation ✓

Part 4 – Development of the dynamic model
- Lagrange approach
- Newton-Euler

Development of the dynamic model

Two systematic approaches will be studied

Approche de Lagrange

Approche de Neuton-Euler

The dynamic models use different coordinates:

- Generalized coordinate system (Joint coordinates)
- Cartesian coordinate system (use Jacobian)

Simplification of analysis by using hypothesis:

- Stiff structures
- Constant masses
- Inertias of rotor of motors negligible

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Lagrangian approach

- Connects movement and the work of the energy of a system

$$L = T - U \quad \Rightarrow \quad \Gamma_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \left(\frac{\partial L}{\partial q_i}\right)$$

where

q_i: generalised coordinates of each joint (i.e. angle)

Γ_i: generalised torque: sum of external forces and of dissipative forces (friction)

T: total kinetic energy of the tobot

U: total potential energy of the robot

- Absolute knowledge of total energy and external work

- To do so:
  - Clever choice of generalised coordinates
  - Identification of generalised torque
  - Establish kinetic and potential energy of all elements of the robot

- To accomplish point three
  - ➡ determine all positions, all speeds, the masses and inertias

- Speeds at the extremities of each link are given thanks to the Jacobian matrices

- Kinetic energy is given by linking the speed of the joints to the speed of the centre of gravitation of every element

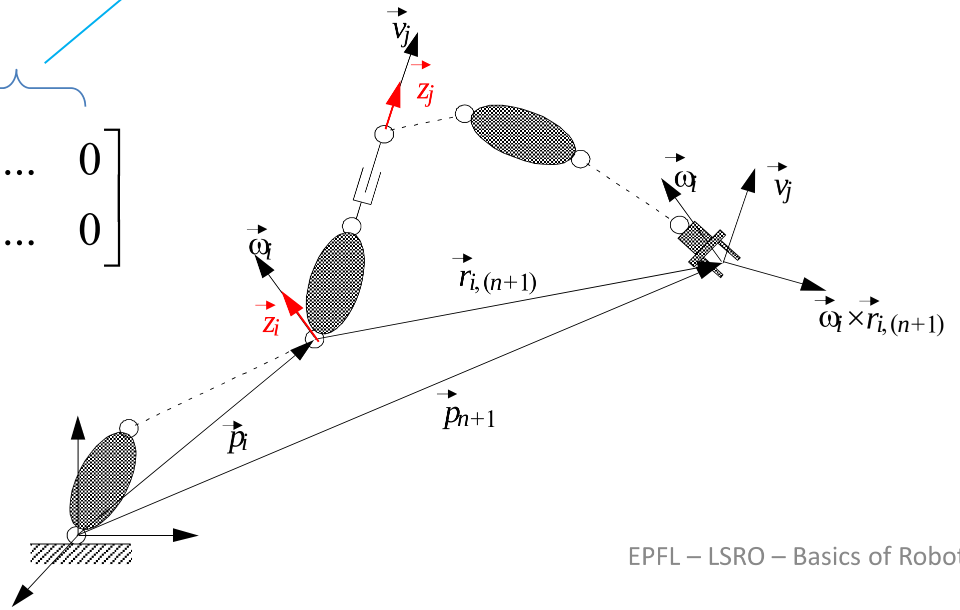
- Individual Jacobians for each segment are easy do develop:

$$v_i = J_p^i \, \dot{q} \qquad \omega_i = J_o^i \, \dot{q}$$

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = J^i \dot{q} = \begin{bmatrix} J_p^i \\ J_O^i \end{bmatrix} \dot{q}$$

$$J^i = \begin{bmatrix} J_p^i \\ J_O^i \end{bmatrix} = \begin{bmatrix} J_{p,1}^i & J_{p,2}^i & \dots & J_{p,i}^i & 0 & \dots & 0 \\ J_{O,1}^i & J_{O,2} & \dots & J_{O,i}^i & 0 & \dots & 0 \end{bmatrix}$$

The velocities of the segment i (*v_i and ω_i*) are independent from the joint velocities for j>i.



- Total energy is given by the translations of all gravitational centres and the rotations of the elements.

- Translation energy:

$$T_T = \frac{1}{2}\sum_{i=1}^{n} m_i v_i^T v_i = \frac{1}{2}\sum_{i=1}^{n} m_i \dot{q}^T J_p^{(i)T} J_p^{(i)} \dot{q}$$

- Rotational energy:

$$T_R = \frac{1}{2}\sum_{i=1}^{n} \omega_i^T \left(R_i I_i R_i^T\right)\omega_i = \frac{1}{2}\sum_{i=1}^{n} \dot{q}^T J_O^{(i)T} \left(R_i I_i R_i^T\right) J_O^{(i)} \dot{q}$$

I_i : inertial tensor of arm i

R_i : rotational matrix between inertial referential (base of robot) and the referential of arm i

- Total kinetic energy is:

$$T_T = T_T + T_R = \frac{1}{2}\sum_{i=1}^{n}\left(m_i \dot{q}^T J_p^{(i)T} J_p^{(i)} \dot{q} + \dot{q}^T J_O^{(i)T} R_i I_i R_i^T J_O^{(i)} \dot{q}\right)$$

- Hypothesis: arms are rigid → potential energy due to gravitational forces

$$U = -\sum_{i=1}^{n} m_i \cdot g \cdot p_i$$

where:

m_i mass of arm i

g gravity vector in the inertial referential (base of the robot) ie, $g = [0 \quad 0 \quad -g_0]^T$

p_i position vector of the center of the mass i in relation to the inertial referential

- Express total kinetic energy with this inertial matrix

$$T = \frac{1}{2}\dot{q}^T B(q)\dot{q}$$

- Inertial matrix depends on the configuration of the robot:

$$B(q) = \sum_{i=1}^{n}\left(m_i J_p^{(i)T} J_p^{(i)} + J_O^{(i)T} R_i I_i R_i^T J_O^{(i)}\right)$$

- Scalar form of kinetic energy

$$T = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}\dot{q}_i\dot{q}_j$$

b_{ij} represents the elements of the matrix $B(q)$

Parenthesis : The Inertial Matrix

$$T = T_T + T_R = \frac{1}{2}\sum_{i=1}^{n}\left(m_i \dot{q}^T J_p^{(i)T} J_p^{(i)} \dot{q} + \dot{q}^T J_O^{(i)T} R_i I_i R_i^T J_O^{(i)} \dot{q}\right)$$

$$T = \frac{1}{2}\dot{q}^T B(q)\dot{q}$$

$$\Rightarrow \quad B(q) = \sum_{i=1}^{n}\left(m_i J_p^{(i)T} J_p^{(i)} + J_O^{(i)T} R_i I_i R_i^T J_O^{(i)}\right)$$

- After development of the Lagrange equation we find

$$\Gamma = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{d}{dt}\left(\frac{\partial U}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i}$$

$$\frac{d}{dt}\left(\frac{\partial U}{\partial \dot{q}_i}\right) = 0$$

potential energy depends only on the position

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) = \frac{d}{dt}\left(\sum_{j=1}^{n} b_{ij}\dot{q}_j\right) = \sum_{j=1}^{n} b_{ij}\ddot{q}_j + \sum_{j=1}^{n} \frac{d\left(b_{ij}\right)}{dt}\dot{q}_j$$

$$= \sum_{j=1}^{n} b_{ij}\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} \frac{\partial\left(b_{ij}\right)}{\partial q_k}\dot{q}_k\dot{q}_j$$

$$\Gamma = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{d}{dt}\left(\frac{\partial U}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i}$$

$$\frac{\partial T}{\partial q_i} = \frac{\partial}{\partial q_i}\left(\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}\dot{q}_i\dot{q}_j\right) = \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n}\frac{\partial b_{jk}}{\partial q_i}\dot{q}_k\dot{q}_j$$

$$\frac{\partial U}{\partial q_i} = -\sum_{j=1}^{n} m_j\left(g \cdot \frac{\partial p_j}{\partial q_i}\right) = -\sum_{j=1}^{n} m\,\mathbf{g}\,J_{p,i}^{j}$$

- Finally we get the dynamic model of a robot:

$$\Gamma_i = \underbrace{\sum_{j=1}^{n} b_{ij}\ddot{q}_j}_{\text{a}} - \underbrace{\sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j}_{\text{b}} - \underbrace{\sum_{j=1}^{n} m_j g J_{p,i}^{j}}_{\text{c}}$$

- $\boldsymbol{\Gamma_i}$ represents the vector with the generalized forces

- Expression is valid for the whole robot

Inverse dynamic model dependencies

$$\Gamma_i = \underbrace{\sum_{j=1}^{n} b_{ij}\ddot{q}_j}_{a} - \underbrace{\sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j}_{b} - \underbrace{\sum_{j=1}^{n} m_j g J_{p,i}^{j}}_{c}$$

- <u>**Acceleration terms**</u>
  - b_{ii} represents the inertia of the articulation *i*; all other articulations are considered as blocked
  - b_{ij} represents the effects of the acceleration of one articulation on the other

- <u>**Square terms of the joint velocities**</u>
  - Centrifugal force on articulation *i* due to the speed of articulation *j*

$$\left(\frac{\partial b_{ij}}{\partial q_j} - \frac{1}{2}\frac{\partial b_{jj}}{\partial q_i}\right)\dot{q}_j^2$$

$$\left(\frac{\partial b_{ii}}{\partial q_i} - \frac{1}{2}\frac{\partial b_{ii}}{\partial q_i}\right) = 0 \quad \text{because} \quad \frac{\partial b_{ii}}{\partial q_i} = 0$$

  - Other terms represent the Coriolis effects on *i* due to speed of *j* and *k*

- <u>**Gravitational acceleration terms**</u>

- Clever construction of the robot can reduce coupling of the different articulations

- Add to the generalized forces Γ (vector of motor torques) , the viscous and dry frictions, and the contact forces on the terminal point :

$$\Gamma - J^T(q)\cdot h = B(q)\ddot{q} + C(q,\dot{q})\cdot\dot{q} + F_v\dot{q} + F_s\,\mathrm{sgn}(\dot{q}) - G(q)$$

Projection of the Force at the end effector

Viscous friction

Dry friction

- Elements of the matrix *C* are : c_{ij}

$$\sum_{j=1}^{n} c_{ij}\dot{q}_j = \sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial b_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j$$

- Elements of the vector *G* are:

$$g_i(q) = \sum_{j=1}^{n} m_j q J_{p,i}^{j}$$

- Furthermore:

Γ : vetor of motor torques

h : vector of forces/torques taking effect at the end effector

F_v : diagonal matrix with coefficents for viscous friction

F_s : diagonal matrix with coefficents for dry friction

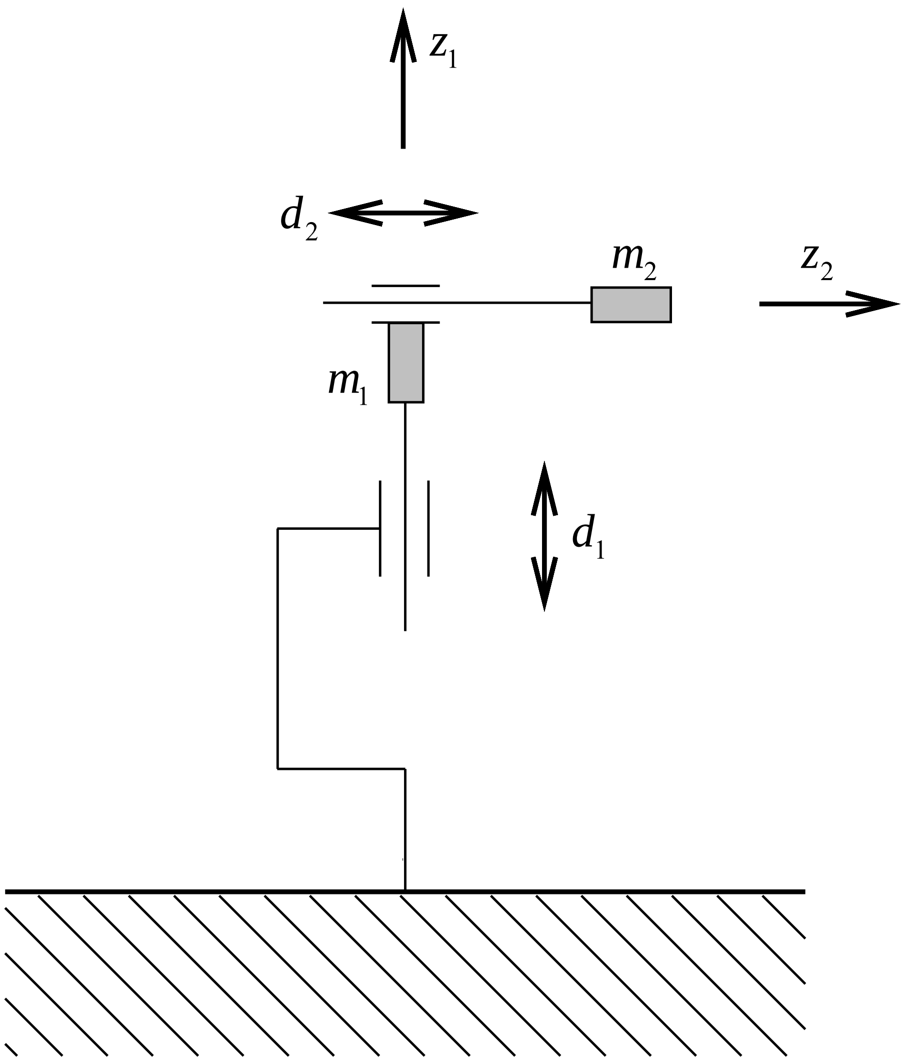
Approche de Lagrange - Résumé :

1. Décrire le robot-
   1. Identifier le point outil (TCP) et les articulations diverses.
   2. Identifier les segments du robot.
2. Définir le de base du robot.
3. Définir les se référentiel ns positif et négatif de chaque articulation.
4. Ecrire les Jacobiens locaux ramenés à chaque segment.
5. Déduire la matrice d'inertie
   1. Grâce aux éléments inertiels de chaque segment (masse et tenseur d'inertie)
   2. Grâce aux Jacobiens ramenés aux segments (pnt 4).
6. Déduire les différentes composantes de l'équation dynamique (Coriolis, Centrifuge et Gravité).

Example:

‘Two-link cartesian arm’

Detailed development is provided in an independent document



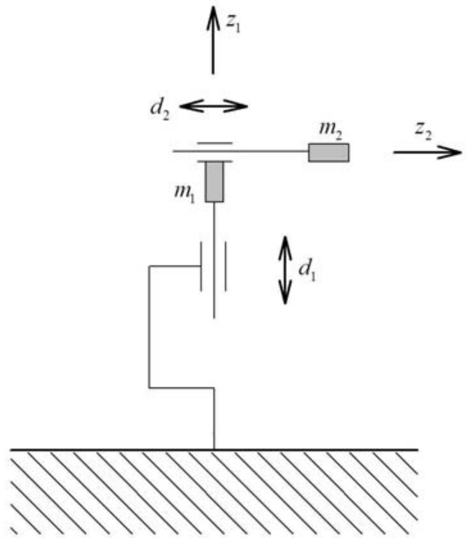
z_1
d_2
m_2
z_2
m_1
d_1

- Generalized coordinates: $q = \left[d_1 \ d_2 \right]^T$

- Gravitational vector: $g = \left[0 \ 0 \ \text{-}g_0 \right]^T$

$$J_p^1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad J_p^2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad J_O^2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad J_O^2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- Inertial matrix: $B = \begin{bmatrix} m_1 + m_2 & 0 \\ 0 & m_2 \end{bmatrix}$



- B is constant → no Centrifugal forces, no Coriolis effect

- Dynamic model is calculated:

$$\begin{bmatrix} m_1 + m_2 & 0 \\ 0 & m_2 \end{bmatrix} \cdot \begin{bmatrix} \ddot{d}_1 \\ \ddot{d}_2 \end{bmatrix} + \begin{bmatrix} \left(m_1 + m_2\right) g_0 \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Newton-Euler approach

- Based on the general theorem of mechanics: depends on the forces applied on the center of the mass

- Equation of Newton $f = m\dot{v}_c$

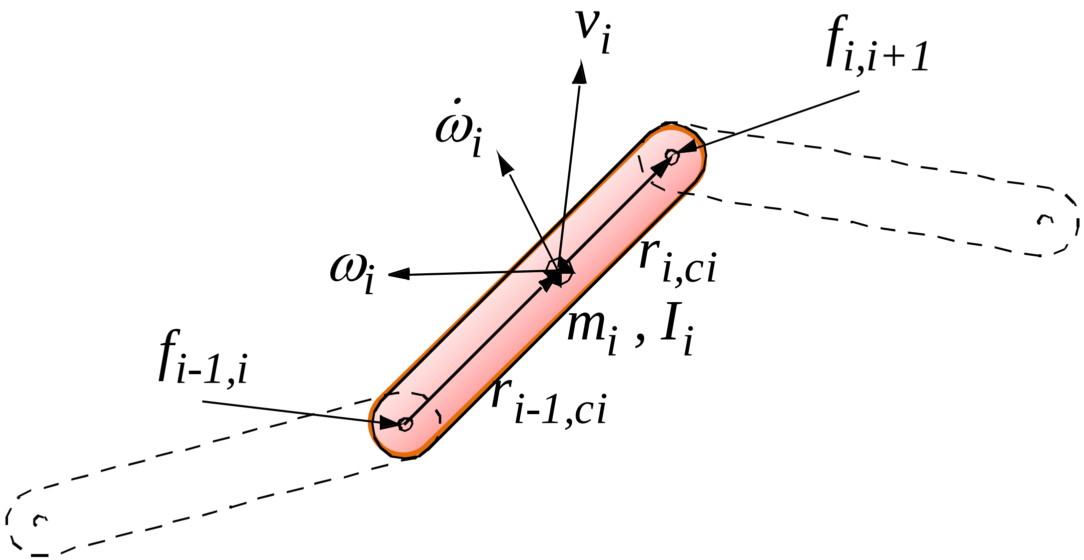
- Equation of Euler $\mu = I_C\dot{\omega} + \omega \times I_C \omega$

where:

$f,\ \mu$:	vector of forces/torques acting on gravitational center
I_C :	moment of inertia of a body
$_C$:	index of a measured entity in relation to the gravitational center
$\times$:	cross product

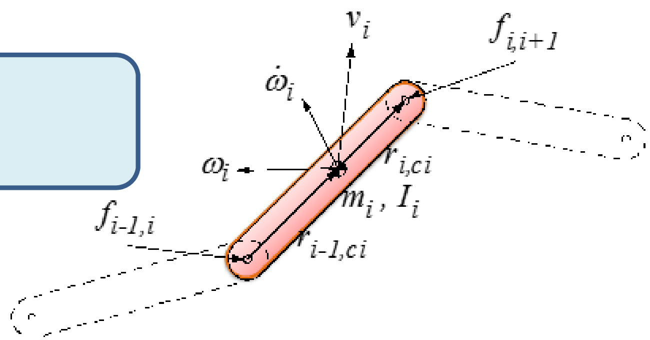
- We have only to develop these two equations

- Leads to the two following iterative identities



- Linear displacement equation

$$f_{i-1,i} - f_{i,i+1} + m_i g - m_i \dot{v}_{ci} = 0 \quad i \in [0, n]$$



where:

i : index of actual body

$f_{j,k}$: interaction force between two bodies

m_i : mass of the i^{th} body

g : gravitational acceleration vector

v_{ci} : linear speed vector of gravitational center

- Angular displacement equation

$$\mu_{i-1,i} - \mu_{i,i+1} + \left(r_{i,ci} \times f_{i,i+1}\right) - \left(r_{i-1,ci} \times f_{i-1,i}\right) - I_{ci}\dot{\omega}_i - \left(\omega_i \times I_{ci}\omega_i\right) = 0 \quad i \in [0,n]$$

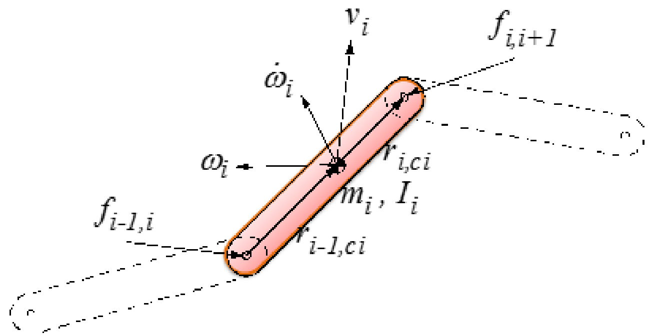
where:

$\mu_{j,k}$: interaction torque between two bodies

$r_{j,ck}$: position vector for finding the kth center of gravity in the jth mass of the ith original reference

I_i : inertial matrix in relation to the center of gravity

ω_i : angular speed of ith center of mass



Approche de Newton-Euler - Résumé :

1. Décrire le robot-
    1. Identifier le point outil (TCP) et les articulations diverses.
    2. Identifier les segments du robot.
2. Définir le référentiel de base du robot.
3. Définir les sens positif et négatif de chaque articulation.
4. Ecrire les équations de Newton et d'Euler pour chaque segment.
5. Résoudre le système d'équations formé et déduire le modèle dynamique formé par l'expression des couples articulaires pour chaque segment.

Example:
‘Two-link cartesian arm’

Detailed development is provided in an independent document

- Body 0 (base)
  - $v_{c0} = \dot{v}_{c0} = \omega_0 = \dot{\omega}_0$
  - All counter reactions are provided by the base
  - $g = \begin{bmatrix} 0 & 0 & -g_0 \end{bmatrix}^T$

- Body 1
  - $v_{c1} = \begin{bmatrix} 0 & 0 & \dot{d}_1 \end{bmatrix}^T \; ; \; \dot{v}_{c1} = \begin{bmatrix} 0 & 0 & \ddot{d}_1 \end{bmatrix}^T \; ; \; \omega_1 = \dot{\omega}_1 = 0$
    $f_{0,1} = \begin{bmatrix} fx_{0,1} & fy_{0,1} & \tau_1 \end{bmatrix}^T \; ; \; \mathrm{f}_{1,2} = \begin{bmatrix} \tau_2 & fy_{0,1} & fz_{1,2} \end{bmatrix}^T$
  - $f_{0,1} - f_{1,2} + m_1 g - m_1 \dot{v}_{c1} = 0$
  - τ_1 is the force generated by the first motor

- Body 2
  - $v_{c2} = \begin{bmatrix} \dot{d}_2 & 0 & \dot{d}_1 \end{bmatrix}^T$; $\dot{v}_{c2} = \begin{bmatrix} \ddot{d}_2 & 0 & \ddot{d}_1 \end{bmatrix}^T$; $\omega_2 = \dot{\omega}_2 = 0$; $f_{2,3} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$
  - $f_{1,2} = 0$ because there is no force acting on the tip of the robot
  - τ_2 is the force generated by the second motor
  - $f_{1,2} - f_{2,3} + m_2 g - m_2 \dot{v}_{c2} = 0$

- After resolving the system, we get the same result as with the Lagrange approach:

$$\begin{bmatrix} m_1 + m_2 & 0 \\ 0 & m_2 \end{bmatrix} \cdot \begin{bmatrix} \ddot{d}_1 \\ \ddot{d}_2 \end{bmatrix} + \begin{bmatrix} (m_1 + m_2) g_0 \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Discussion of the two methods

- Analytical power and equivalency of the two methods
- Punctual consideration of bodies
- **Newton-Euler approach considers internal reactions between adjacent bodies**
  - **Very useful to size the joints**
- Lagrange approach is more systematic
- Lagrange approach does not directly consider the forces, but the work, the stored energy and the generalized coordinates
- The mathematical models is directly useable for implementation in a computer/microprocessor/microcontroller