Control of industrial robots

Robot dynamics

Prof. Paolo Rocco (paolo.rocco@polimi.it)
Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria
Introduction

- With these slides we will derive the dynamic model of the manipulator.

- The dynamic model accounts for the relation between the sources of motion (forces and moments) and the resulting motion (positions and velocities).

- Systematic methods exist to derive the dynamic model of the manipulators, which will be reviewed here, along with the main properties of such model.
Kinetic energy

Consider a point with mass $m$, whose position is described by vector $\mathbf{p}$ with respect to a $xyz$ frame.

We define kinetic energy of the point the quantity:

$$T = \frac{1}{2} m \mathbf{p}^T \mathbf{p}$$

Similarly, for a system of points:

$$T = \frac{1}{2} \sum_{i=1}^{n} m_i \mathbf{p}_i^T \mathbf{p}_i$$

Consider now a rigid body, with mass $m$, volume $V$ and density $\rho$. The kinetic energy is defined with the integral:

$$T = \frac{1}{2} \int_V \mathbf{p}^T \mathbf{p} \rho dV$$
Potential energy

A system of position forces (i.e. depending only on the positions of the points of application) is said to be conservative if the work made by each force does not depend on the trajectory followed by the point of application, but only on the initial and final positions. In this case the elementary work coincides with the differential, with changed sign, of a function called potential energy:

\[dW = -dU\]

An example of a system of conservative forces is the gravitational force. For a point mass we have the potential energy:

\[U = -mg_0^T p\]

where \(g_0\) is the gravity acceleration vector.

For a rigid body:

\[U = -\int_V g_0^T p \rho dV = -mg_0^T p_i\]

where \(p_i\) is the position of the center of mass.
Let us consider a system of \( r \) rigid bodies (as for example, the links of a robot). If all these bodies are free to move in space, the motion of the system is, at each time instant, described by means of \( 6r \) coordinates \( x \).

Suppose now that limitations exist in the motion of the bodies of the system (as for example the presence of a joint, which eliminates five out of the six relative degrees of freedom between two consecutive links). A constraint thus exists on the motion of the bodies, which we will express with the relation:

\[
h(x) = 0
\]

Such a constraint is said to be holonomic (it depends only on position coordinates, not velocities) and stationary (it does not change with time).
Free coordinates

$h(x) = 0$

If the constraints $h$ are composed of $s$ scalar equations and they are all continuously differentiable, it is possible, by means of the constraints, to eliminate $s$ coordinates from the system equations.

The remaining $n = 6r - s$ coordinates are called free, or Lagrangian, or natural, or generalized coordinates. $n$ is the number of degrees of freedom of the mechanical system.

For example in a robot with 6 joints, out of the 36 original coordinates, 30 are eliminated by virtue of the constraints imposed by the 6 joints. The remaining 6 are the Lagrangian coordinates (typically the joint variables used in the kinematic model).
Lagrange’s equations

Given a system of rigid bodies, whose positions and orientations can be expressed by means of \( n \) generalized coordinates \( q_i \), we define Lagrangian of the mechanical system the quantity:

\[
L(q, \dot{q}) = T(q, \dot{q}) - U(q)
\]

where \( T \) and \( U \) are the kinetic and the potential energies, respectively. Let then \( \xi_i \) be the generalized forces associated with the generalized coordinates \( q_i \). The elementary work performed by the forces acting on the system can be expressed as:

\[
dW = \sum_{i=1}^{n} \xi_i dq_i
\]

It can be proven that the dynamics of the system of bodies is expressed by the following Lagrange’s equations:

\[
\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \xi_i, \quad i = 1, \ldots, n
\]
An example

Let us consider a system composed of a motor rigidly connected to a load, subjected to the gravitational force. Let:

- $I_m$ and $I_l$ the moments of inertia of the motor and the load with respect to the motor spinning axis
- $m$ the mass of the load
- $l$ the distance of the center of mass of the load from the axis of the motor.

**Kinetic energy of the motor**

$$T_m = \frac{1}{2} I_m \dot{\theta}^2$$

**Kinetic energy of the load**

$$T_c = \frac{1}{2} I_l \dot{\theta}^2$$
Gravitational potential energy:

\[ U = -mg_l^T \mathbf{p}_l = -m[0 \ -g \begin{bmatrix} l \cos \vartheta \\ l \sin \vartheta \end{bmatrix}] = mgl \sin \vartheta \]

Lagrangian:

\[ L = T_m + T_c - U = \frac{1}{2}(l_m + l)\dot{\vartheta}^2 - mgl \sin \vartheta \]

Lagrange’s equations:

\[ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\vartheta}} \right) - \frac{\partial L}{\partial \vartheta} = \tau \quad \Rightarrow \quad \frac{d}{dt} \left( (l + l_m)\dot{\vartheta} \right) + mgl \cos \vartheta = \tau \]

Then:

\[ (l + l_m)\ddot{\vartheta} + mgl \cos \vartheta = \tau \]

This equation can be easily interpreted as the equilibrium of moments around the rotation axis.
Kinetic energy of a link

The contribution of kinetic energy of a single link can be computed with the following integral:

\[
T_i = \frac{1}{2} \int_{V_i} p_i^* \dot{p}_i^* \rho dV
\]

\[p_i^*\] generic point along the link

Position of the center of mass:

\[p_{l_i} = \frac{1}{m_i} \int_{V_i} p_i^* \rho dV\]

Velocity of the generic point:

\[
\dot{p}_i^* = \dot{p}_{l_i} + \omega_i \times r_i = \dot{p}_{l_i} + S(\omega_i)r_i
\]

where:

\[
S(\omega_i) = \begin{bmatrix}
0 & -\omega_{iz} & \omega_{iy} \\
\omega_{iz} & 0 & -\omega_{ix} \\
-\omega_{iy} & \omega_{ix} & 0
\end{bmatrix}
\]

(skew-symmetric matrix)
Kinetic energy of a link

Translational contribution:

\[
\frac{1}{2} \int_{V_i} \dot{\mathbf{p}}_i^T \dot{\mathbf{p}}_i \rho dV = \frac{1}{2} m_i \mathbf{p}_i^T \dot{\mathbf{p}}_i 
\]

Mutual contribution:

\[
\int_{V_i} \dot{\mathbf{p}}_i^T \mathbf{S}(\omega_i) r_i \rho dV = \dot{\mathbf{p}}_i^T \mathbf{S}(\omega_i) \int_{V_i} (\mathbf{p}^*_i - \mathbf{p}_i) \rho dV = 0
\]

Rotational contribution:

\[
\frac{1}{2} \int_{V_i} \mathbf{r}_i^T \mathbf{S}^T(\omega_i) \mathbf{S}(\omega_i) r_i \rho dV = \frac{1}{2} \omega_i^T \left( \int_{V_i} \mathbf{S}^T(\mathbf{r}_i) \mathbf{S}(\mathbf{r}_i) dV \right) \omega_i = \frac{1}{2} \omega_i^T \mathbf{I}_i \omega_i
\]

Note:

\[
\mathbf{S}(\omega_i) r_i = -\mathbf{S}(\mathbf{r}_i) \omega_i
\]

Then:

\[
T_i = \frac{1}{2} m_i \mathbf{p}_i^T \dot{\mathbf{p}}_i + \frac{1}{2} \omega_i^T \mathbf{I}_i \omega_i \quad \text{König Theorem}
\]

Inertia tensor

We define inertia tensor the matrix:

\[
I_i = \begin{bmatrix}
\int \left( r_{iy}^2 + r_{iz}^2 \right) \rho dV & - \int r_{ix} r_{iy} \rho dV & - \int r_{ix} r_{iz} \rho dV \\
-\int r_{ix} r_{iy} \rho dV & \int \left( r_{ix}^2 + r_{iz}^2 \right) \rho dV & - \int r_{iy} r_{iz} \rho dV \\
-\int r_{ix} r_{iz} \rho dV & -\int r_{iy} r_{iz} \rho dV & \int \left( r_{ix}^2 + r_{iy}^2 \right) \rho dV
\end{bmatrix} = \\
\begin{bmatrix}
I_{ixx} & -I_{ixy} & -I_{ixz} \\
-I_{ixy} & I_{iyy} & -I_{iyz} \\
-I_{ixz} & -I_{iyz} & I_{izz}
\end{bmatrix}
\]

(symmetric matrix)

The inertia tensor, if expressed in the base frame, depends on the robot configuration. If the angular velocity is expressed with reference to a frame rigidly attached to the link (for example the DH frame):

\[
\omega_i^j = R_i^T \omega_i
\]

the inertia tensor referred to this frame is a constant matrix. Moreover it results:

\[
I_i = R_i I_i^j R_i^T
\]
Sum of the contributions

Let us sum the translational and rotational contributions:

\[ T_i = \frac{1}{2} m_i \dot{p}_{l_i}^T \ddot{p}_{l_i} + \frac{1}{2} \omega_i^T R_i l_i R_i^T \omega_i \]

Linear velocity:

\[ \dot{p}_{l_i} = j_{P1}^{(l_i)} q_1 + \ldots + j_{P_i}^{(l_i)} q_i = J_t^{(l_i)} q \]

\[ J_t^{(l_i)} = \begin{bmatrix} j_{P1}^{(l_i)} & \ldots & j_{P_i}^{(l_i)} & 0 & \ldots & 0 \end{bmatrix} \]

Angular velocity:

\[ \omega_i = j_{O1}^{(l_i)} q_1 + \ldots + j_{Oi}^{(l_i)} q_i = J_o^{(l_i)} q \]

\[ J_o^{(l_i)} = \begin{bmatrix} j_{O1}^{(l_i)} & \ldots & j_{Oi}^{(l_i)} & 0 & \ldots & 0 \end{bmatrix} \]

Columns of the Jacobian:

\[ \begin{bmatrix} j_{Pj}^{(l_i)} \\ j_{Oj}^{(l_i)} \end{bmatrix} = \begin{bmatrix} z_{j-1} \\ 0 \end{bmatrix} \quad \text{prismatic joint} \]

\[ \begin{bmatrix} j_{Pj}^{(l_i)} \\ j_{Oj}^{(l_i)} \end{bmatrix} = \begin{bmatrix} z_{j-1} \times (p_{l_i} - p_{j-1}) \\ z_{j-1} \end{bmatrix} \quad \text{rotational joint} \]
Inertia matrix

Substituting expressions for the linear and angular velocities:

\[ T_i = \frac{1}{2} m_i \dot{q}_i^T J_P^{(l_i)}^T J_P^{(l_i)} \dot{q}_i + \frac{1}{2} \dot{q}_i^T J_O^{(l_i)}^T R_i l_i R_i^T J_O^{(l_i)} \dot{q}_i \]

By summing the contributions of all the links we obtain the kinetic energy of the whole arm:

\[ T = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n b_{ij}(\dot{q}_i \dot{q}_j) = \frac{1}{2} \dot{q}^T B(q) \dot{q} \]

where:

\[ B(q) = \sum_{i=1}^n \left( m_i J_P^{(l_i)}^T J_P^{(l_i)} + J_O^{(l_i)}^T R_i l_i R_i^T J_O^{(l_i)} \right) \]

is the inertia matrix of the manipulator.

\[
\begin{cases}
\text{symmetric} \\
\text{positive definite} \\
\text{depends on } q
\end{cases}
\]
Potential energy

The potential energy of a rigid link is related just to the gravitational force:

\[ U_i = - \int_V \vec{g}_0^T \rho_i^* \rho dV = - m_i \vec{g}_0^T \vec{p}_i \]

where \( \vec{g}_0 \) is the gravity acceleration vector expressed in the base frame.

The potential energy of the whole manipulator is then the sum of the single contributions:

\[ U = \sum_{i=1}^{n} U_i = - \sum_{i=1}^{n} m_i \vec{g}_0^T \vec{p}_i \]
Motion equations

The Lagrangian of the manipulator is:

\[
L(q, \dot{q}) = T(q, \dot{q}) - U(q) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} b_{ij}(q) \dot{q}_i \dot{q}_j + \sum_{i=1}^{n} m_i g_0^T p_{ii}(q)
\]

If we differentiate the Lagrangian:

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

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

Furthermore:

\[
\frac{\partial T}{\partial q_i} = \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\partial b_{jk}(q)}{\partial q_i} \dot{q}_k \dot{q}_j \quad \frac{\partial U}{\partial q_i} = -\sum_{j=1}^{n} m_j g_0^T \frac{\partial p_{ij}}{\partial q_i} = -\sum_{j=1}^{n} m_j g_0^T j_{P_i}^{(ij)}(q) = g_i(q)
\]
From Lagrange’s equations we obtain:

\[
\sum_{j=1}^{n} b_{ij}(q)\dddot{q}_j + \sum_{j=1}^{n} \sum_{k=1}^{n} h_{ijk}(q)\dddot{q}_k \dddot{q}_j + g_i(q) = \dddot{\xi}_i \quad i = 1, \ldots, n
\]

where: \( h_{ijk} = \frac{\partial b_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial b_{jk}}{\partial q_i} \)

gravitational term, depends only on the joint positions

**Acceleration terms:**

\( b_{ii} \): inertia moment as “seen” from the axis of joint \( i \)

\( b_{ij} \): effect of the acceleration of joint \( j \) on the joint \( i \)

**Centrifugal and Coriolis terms:**

\( h_{ij}q_j^2 \): centrifugal effect induced at joint \( i \) by the velocity of joint \( j \)

\( h_{ijk}q_jq_k \): Coriolis effect induced at joint \( i \) by the velocities of joints \( j \) and \( k \)

\( h_{iii} = 0 \)
Non conservative forces

Besides the gravitational conservative forces, other forces act on the manipulator:

- actuation torques \( \tau \)
- viscous friction torques \(-F_v \dot{q}\)
- static friction torques \(-f_s(q, \dot{q})\)

\(F_v\) diagonal matrix of viscous friction coefficients
\(f_s(q, \dot{q})\) function that models the static friction at the joint
In vector form the **dynamic model** can be expressed as follows:

\[
B(q)\ddot{q} + C(q, \dot{q})\dot{q} + F_v \dot{q} + f_s (q, \dot{q}) + g(q) = \tau
\]

where \( C \) is a suitable \( n \times n \) matrix, whose elements satisfy the equation:

\[
\sum_{j=1}^{n} c_{ij} \dot{q}_j = \sum_{j=1}^{n} \sum_{k=1}^{n} h_{ijk} \dot{q}_k \dot{q}_j \quad \text{\( C \) is not symmetric in general}
\]
Computation of the elements of $C$

The choice of matrix $C$ is not unique. One possible choice is the following one:

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

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

The generic element of $C$ is:

$$c_{ij} = \sum_{k=1}^{n} c_{ijk} \dot{q}_k$$

where:

$$c_{ijk} = \frac{1}{2} \left( \frac{\partial b_{ij}}{\partial q_k} + \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right) \text{ Christoffel symbols of the first kind}$$
Skew-symmetry of matrix $\dot{B} - 2C$

The previous choice of matrix $C$ allows to prove an important property of the dynamic model of the manipulator. Matrix:

$$N(q, \dot{q}) = \dot{B}(q) - 2C(q, \dot{q})$$

is skew-symmetric:

$$w^T N(q, \dot{q}) w = 0, \quad \forall w$$

In fact:

$$c_{ij} = \frac{1}{2} \sum_{k=1}^{n} \frac{\partial b_{ij}}{\partial q_k} \dot{q}_k + \frac{1}{2} \sum_{k=1}^{n} \left( \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right) \dot{q}_k = \frac{1}{2} \dot{b}_{ij} + \frac{1}{2} \sum_{k=1}^{n} \left( \frac{\partial b_{ik}}{\partial q_j} - \frac{\partial b_{jk}}{\partial q_i} \right) \dot{q}_k$$

$$n_{ij} = \dot{b}_{ij} - 2c_{ij} = \sum_{k=1}^{n} \left( \frac{\partial b_{jk}}{\partial q_i} - \frac{\partial b_{ik}}{\partial q_j} \right) \dot{q}_k$$

$$n_{ij} = -n_{ji}$$

(skew-symmetric matrix)
Energy conservation

The equation:

\[ \dot{q}^T N(q, \dot{q}) \dot{q} = 0 \]

(particular case of the previous one) is valid whatever the choice of matrix \( C \) is.
From the energy conservation principle, the derivative of the kinetic energy equals the power generated by all the forces acting at the joint of the manipulator:

\[ \frac{1}{2} \frac{d}{dt} \left( \dot{q}^T B(q) \dot{q} \right) = \dot{q}^T \left( \tau - F_v \dot{q} - f_s(q, \dot{q}) - g(q) \right) \]

Taking the derivative at the left hand side and using the equation of the model:

\[ \frac{1}{2} \frac{d}{dt} \left( \dot{q}^T B(q) \dot{q} \right) = \frac{1}{2} \dot{q}^T B(q) \dot{q} + \dot{q}^T B(q) \ddot{q} = \frac{1}{2} \dot{q}^T \left( B(q) - 2C(q, \dot{q}) \right) \ddot{q} \]

\[ + \dot{q}^T \left( \tau - F_v \dot{q} - f_s(q, \dot{q}) - g(q) \right) \]

from which the equation follows.
Linearity in the dynamical parameters

If we assume a simplified expression for the static friction function:

\[ f_s(q, \dot{q}) = F_s \text{sgn}(\dot{q}) \]

it is possible to prove that the dynamic model of the manipulator is linear with respect to a suitable set of dynamic parameters (masses, moments of inertia). We can then write:

\[ \tau = Y(q, \dot{q}, \ddot{q})\pi \]

\( \pi \): vector of \( p \) constant parameters

\( Y \): \( n \times p \) matrix, function of joint positions, velocities and accelerations (regression matrix)
Consider a two-link Cartesian manipulator, characterized by link masses $m_1$ and $m_2$.

The vector of generalized coordinates is:

$$q = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

The Jacobians needed for the computation of the inertia matrix are the following ones:

$$J^{(l_1)}_p = \begin{bmatrix} j^{(l_1)}_p \\ 0 \end{bmatrix} = \begin{bmatrix} z_0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$J^{(l_2)}_p = \begin{bmatrix} j^{(l_2)}_p \\ j^{(l_2)}_{l_2} \end{bmatrix} = \begin{bmatrix} z_0 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$$

while there are no contributions to the angular velocities.
Computing the inertia matrix with the general formula, we obtain:

\[
B = m_1 J_p^{(l_1)} J_p^{(l_1)} + m_2 J_p^{(l_2)} J_p^{(l_2)} = \begin{bmatrix}
m_1 + m_2 & 0 \\
0 & m_2
\end{bmatrix}
\]

As \( B \) is constant, \( C=0 \), i.e. there are no centrifugal and Coriolis terms. Then, since:

\[
g_0 = \begin{bmatrix}
0 \\
0 \\
-g
\end{bmatrix}
\]

the vector of the gravitational terms is:

\[
g = \begin{bmatrix}
(m_1 + m_2)g \\
0
\end{bmatrix}
\]
Two-link Cartesian manipulator

If there are no friction torques and no forces at the end-effector:

\[
\begin{align*}
(m_1 + m_2)\ddot{d}_1 + (m_1 + m_2)g &= f_1 \\
m_2\ddot{d}_2 &= f_2
\end{align*}
\]

\(f_1\) e \(f_2\): forces which act along the generalized coordinates
Let us consider a two-link planar manipulator:

- masses: $m_1$ and $m_2$
- lengths: $a_1$ and $a_2$
- distances of the centers of mass from the joint axes: $l_1$ and $l_2$
- moments of inertia around axes passing through the centers of mass and parallel to $z_0$: $I_1$ and $I_2$

Generalized coordinates: $\mathbf{q} = \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \end{bmatrix}$

The Jacobians needed for the computation of the inertia matrix depend on the following vectors:

- $\mathbf{p}_{l_1} = \begin{bmatrix} l_1 c_1 \\ l_1 s_1 \\ 0 \end{bmatrix}$
- $\mathbf{p}_{l_2} = \begin{bmatrix} a_1 c_1 + l_2 c_{12} \\ a_1 s_1 + l_2 s_{12} \\ 0 \end{bmatrix}$
- $\mathbf{p}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$
- $\mathbf{p}_1 = \begin{bmatrix} a_1 c_1 \\ a_1 s_1 \\ 0 \end{bmatrix}$
- $\mathbf{z}_0 = \mathbf{z}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
Two-link planar manipulator

Link 1

\[
J^{(l_1)}_P = \begin{bmatrix} j^{(l_1)}_P & 0 \end{bmatrix} = \begin{bmatrix} z_0 \times (p_{l_1} - p_0) & 0 \end{bmatrix} = \begin{bmatrix} -l_1 s_1 & 0 \\ l_1 c_1 & 0 \\ 0 & 0 \end{bmatrix}
\]

\[
J^{(l_1)}_O = \begin{bmatrix} j^{(l_1)}_O & 0 \end{bmatrix} = \begin{bmatrix} z_0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}
\]

Link 2

\[
J^{(l_2)}_P = \begin{bmatrix} j^{(l_2)}_P & j^{(l_2)}_P \end{bmatrix} = \begin{bmatrix} z_0 \times (p_{l_2} - p_0) & z_1 \times (p_{l_2} - p_1) \end{bmatrix} = \begin{bmatrix} -a_1 s_1 - l_2 s_{12} & -l_2 s_{12} \\ a_1 c_1 + l_2 c_{12} & l_2 c_{12} \\ 0 & 0 \end{bmatrix}
\]

\[
J^{(l_2)}_O = \begin{bmatrix} j^{(l_2)}_O & j^{(l_2)}_O \end{bmatrix} = \begin{bmatrix} z_0 & z_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}
\]
Taking into account that the angular velocity vectors $\omega_1$ and $\omega_2$ are aligned with $z_0$, it is not necessary to compute rotation matrices $R_i$, so that the computation of the inertia matrix gives:

$$B(q) = m_1 J_{p}^{(l_1)^T} J_{p}^{(l_1)} + m_2 J_{p}^{(l_2)^T} J_{p}^{(l_2)} + l_1 J_{o}^{(l_1)^T} J_{o}^{(l_1)} + l_2 J_{o}^{(l_2)^T} J_{o}^{(l_2)} = \begin{bmatrix} b_{11}(\theta) & b_{12}(\theta) \\ b_{21}(\theta) & b_{22} \end{bmatrix}$$

$$b_{11} = m_1 l_1^2 + l_1 + m_2 \left( a_1^2 + l_2^2 + 2a_1 l_2 c_2 \right) + l_2 \quad \text{depends on } \theta_2$$

$$b_{12} = b_{21} = m_2 \left( l_2^2 + a_1 l_2 c_2 \right) + l_2 \quad \text{depends on } \theta_2$$

$$b_{22} = m_2 l_2^2 + l_2 \quad \text{constant!}$$
Two-link planar manipulator

From the inertia matrix it is simple to compute the Christoffel symbols:

\[ c_{111} = \frac{1}{2} \frac{\partial b_{11}}{\partial q_1} = 0 \]

\[ c_{112} = c_{121} = \frac{1}{2} \frac{\partial b_{11}}{\partial q_2} = -m_2 a_1 l_2 s_2 = h \]

\[ h = -m_2 a_1 l_2 s_2 \]

\[ c_{122} = \frac{\partial b_{12}}{\partial q_2} - \frac{1}{2} \frac{\partial b_{22}}{\partial q_1} = -m_2 a_1 l_2 s_2 = h \]

\[ c_{211} = \frac{\partial b_{21}}{\partial q_1} - \frac{1}{2} \frac{\partial b_{11}}{\partial q_2} = m_2 a_1 l_2 s_2 = -h \]

\[ c_{212} = c_{221} = \frac{1}{2} \frac{\partial b_{22}}{\partial q_1} = 0 \]

\[ c_{222} = \frac{1}{2} \frac{\partial b_{22}}{\partial q_2} = 0 \]
The expression of matrix $C$ is then:

$$C(q, \dot{q}) = \begin{bmatrix} h\dot{\theta}_2 & h(\dot{\theta}_1 + \dot{\theta}_2) \\ -h\dot{\theta}_1 & 0 \end{bmatrix} = \begin{bmatrix} -m_2a_1l_2s_2\dot{\theta}_2 & -m_2a_1l_2s_2(\dot{\theta}_1 + \dot{\theta}_2) \\ m_2a_1l_2s_2\dot{\theta}_1 & 0 \end{bmatrix}$$

We can verify that matrix $N$ is skew-symmetric:

$$N(q, \dot{q}) = \dot{B}(q) - 2C(q, \dot{q}) = \begin{bmatrix} 2h\dot{\theta}_2 & h\dot{\theta}_2 \\ h\dot{\theta}_2 & 0 \end{bmatrix} - 2\begin{bmatrix} h\dot{\theta}_2 & h(\dot{\theta}_1 + \dot{\theta}_2) \\ -h\dot{\theta}_1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -2h\dot{\theta}_1 - h\dot{\theta}_2 \\ 2h\dot{\theta}_1 + h\dot{\theta}_2 & 0 \end{bmatrix} = -N^T(q, \dot{q})$$

Furthermore, since $g_0 = [0 \ -g \ 0]^T$, the vector of the gravitational terms is:

$$g = \begin{bmatrix} (m_1l_1 + m_2a_1)gc_1 + m_2gl_2c_{12} \\ m_2gl_2c_{12} \end{bmatrix}$$
Two-link planar manipulator

Without friction at the joints, the motion equations are:

\[
\begin{align*}
\left( m_1 l_1^2 + l_1 + m_2 \left( a_1^2 + l_2^2 + 2a_1 l_2 c_2 \right) + l_2 \right) \ddot{\theta}_1 + \left( m_2 \left( l_2^2 + a_1 l_2 c_2 \right) + l_2 \right) \ddot{\theta}_2 + \\
-2m_2 a_1 l_2 s_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 a_1 l_2 s_2 \dot{\theta}_2^2 + \\
+ \left( m_1 l_1 + m_2 a_1 \right) g c_1 + m_2 g l_2 c_{12} = \tau_1
\end{align*}
\]

\[
\begin{align*}
\left( m_2 \left( l_2^2 + a_1 l_2 c_2 \right) + l_2 \right) \ddot{\theta}_1 + \left( m_2 l_2^2 + l_2 \right) \ddot{\theta}_2 + \\
+ m_2 a_1 l_2 s_2 \dot{\theta}_1^2 + m_2 l_2 g c_{12} = \tau_2
\end{align*}
\]

\(\tau_i\) : torques applied at the joints
Two-link planar manipulator

By simple inspection, we can obtain the dynamical parameters with respect to which the model is linear, i.e. those parameters for which we can write:

$$\tau = Y(q, \dot{q}, \ddot{q})\pi$$

We have:

$$\pi = [\pi_1 \quad \pi_2 \quad \pi_3 \quad \pi_4 \quad \pi_5]^T$$

$$\pi_1 = m_1l_1$$
$$\pi_2 = l_1 + m_1l_1^2$$
$$\pi_3 = m_2$$
$$\pi_4 = m_2l_2$$
$$\pi_5 = l_2 + m_2l_2^2$$
Two-link planar manipulator

\[ Y(q, \dot{q}, \ddot{q}) = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} \\ 0 & 0 & 0 & y_{24} & y_{25} \end{bmatrix} \]

\[ y_{11} = gc_1 \]
\[ y_{12} = \ddot{\theta}_1 \]
\[ y_{13} = a_1^2 \dddot{\theta}_1 + a_1 g c_1 \]
\[ y_{14} = 2a_1 c_2 \dddot{\theta}_1 + a_1 c_2 \dddot{\theta}_2 - 2a_1 s_2 \dot{\theta}_1 \dot{\theta}_2 - a_1 s_2 \dddot{\theta}_2 + g c_{12} \]
\[ y_{15} = \dddot{\theta}_1 + \dddot{\theta}_2 \]
\[ y_{24} = a_1 c_2 \dddot{\theta}_1 + a_1 s_2 \dddot{\theta}_1 + g c_{12} \]
\[ y_{25} = \dddot{\theta}_1 + \dddot{\theta}_2 \]

The coefficients of \( Y \) depend on \( \theta_1, \theta_2 \), their first and second derivatives, \( g \) and \( a_1 \).
Identification of dynamical parameters

The linearity of the dynamic model with respect to the dynamical parameters:

\[ \tau = Y(q, \dot{q}, \ddot{q}) \pi \]

allows to setup a procedure for the experimental identification of the same parameters, which are usually unknown or uncertain.

Suitable motion trajectories must be executed, along which the joint positions \( q \) are recorded, the velocities \( \dot{q} \) are measured or obtained by numerical differentiation, and the accelerations \( \ddot{q} \) are obtained with filtered (also non-causal) differentiation. Also the torques \( \tau \) are measured, directly (with suitable sensors) or indirectly, from the measurements of currents in the motors.

Suppose to have the measurements (direct or indirect ones) of all the variables for the time instants \( t_1, \ldots, t_N \).
Identification of dynamical parameters

With $N$ measurement sets:

$$
\bar{\tau} = \begin{bmatrix}
\bar{\tau}(t_1) \\
\vdots \\
\bar{\tau}(t_N)
\end{bmatrix} = \begin{bmatrix}
Y(t_1) \\
\vdots \\
Y(t_N)
\end{bmatrix} \pi = \bar{Y} \pi
$$

Solving with a least-squares technique:

$$
\pi = \left(\bar{Y}^T \bar{Y}\right)^{-1} \bar{Y}^T \bar{\tau}
$$

- Left pseudo-inverse matrix of $\bar{Y}$

- Only the elements of $\pi$ for which the corresponding column is different from zero can be identified.
- Some parameters are identifiable only in combination with other ones.
- The trajectories to be used must be sufficiently rich (good conditioning of matrix $\bar{Y}^T \bar{Y}$): they explore the robot workspace and involve all components in the dynamic model.
Newton-Euler formulation

An alternative way to formulate the dynamic model of the manipulator is the Newton-Euler method. It is based on balances of forces and moments acting on the single link, due to the interactions with the nearby links in the kinematic chain.

We obtain a system of equations that might be solved in a recursive way, propagating the velocities and accelerations from the base to the end effector, while the forces and moments in the opposite way:

Recursion makes Newton-Euler algorithm computationally efficient.
Definition of the parameters

Let us consider the generic link $i$ of the kinematic chain:

We define the following parameters:

- $m_i$ and $I_i$ mass and inertia tensor of the link
- $r_{i-1,Ci}$ vector from the origin of frame $(i-1)$ to the center of mass $C_i$
- $r_{i,Ci}$ vector from the origin of frame $i$ to the center of mass $C_i$
- $r_{i-1,i}$ vector from the origin of frame $(i-1)$ to the origin of frame $i$
Definition of the variables

\( \mathbf{\dot{p}}_{Ci} \) linear velocity of the center of mass \( C_i \)
\( \mathbf{\dot{p}}_i \) linear velocity of the origin of frame \( i \)
\( \mathbf{\dot{\omega}}_i \) angular velocity of the link
\( \mathbf{\ddot{p}}_{Ci} \) linear acceleration of the center of mass \( C_i \)
\( \mathbf{\ddot{p}}_i \) linear acceleration of the origin of frame \( i \)
\( \mathbf{\ddot{\omega}}_i \) angular acceleration of the link
\( g_0 \) gravity acceleration

\( \mathbf{f}_i \) force exerted by link \( i-1 \) on link \( i \)
\( -\mathbf{f}_{i+1} \) force exerted by link \( i+1 \) on link \( i \)
\( \mathbf{\mu}_i \) moment exerted by link \( i-1 \) on link \( i \) with respect to the origin of frame \( i-1 \)
\( -\mathbf{\mu}_{i+1} \) moment exerted by link \( i+1 \) on link \( i \) with respect to the origin of frame \( i \)

All vectors are expressed in the base frame.
Newton-Euler formulation

Newton’s equation (translational motion of the center of mass)

\[ f_i - f_{i+1} + m_i g_i = m_i \ddot{p}_{C_i} \]

Euler’s equation (rotational motion)

\[ \mu_i + f_i \times r_{i-1,C_i} - \mu_{i+1} - f_{i+1} \times r_{i,C_i} = \frac{d}{dt} \left( I_i \omega_i \right) = I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i) \]

Generalized force at joint \( i \):

\[ \tau_i = \begin{cases} f_i^T z_{i-1} & \text{prismatic joint} \\ \mu_i^T z_{i-1} & \text{rotational joint} \end{cases} \]
Accelerations of a link

Propagation of the velocities:

\[
\omega_i = \begin{cases} \omega_{i-1} & \text{prismatic joint} \\ \omega_{i-1} + \dot{\theta}_i \mathbf{z}_{i-1} & \text{rotational joint} \end{cases}
\]

\[
\dot{p}_i = \begin{cases} \dot{p}_{i-1} + \ddot{d}_i \mathbf{z}_{i-1} + \omega_i \times \mathbf{r}_{i-1,i} & \text{prismatic joint} \\ \dot{p}_{i-1} + \omega_i \times \mathbf{r}_{i-1,i} & \text{rotational joint} \end{cases}
\]

Propagation of the accelerations:

\[
\dot{\omega}_i = \begin{cases} \dot{\omega}_{i-1} & \text{prismatic joint} \\ \dot{\omega}_{i-1} + \ddot{\theta}_i \mathbf{z}_{i-1} + \dot{\theta}_i \omega_{i-1} \times \mathbf{z}_{i-1} & \text{rotational joint} \end{cases}
\]

\[
\ddot{p}_i = \begin{cases} \ddot{p}_{i-1} + \dddot{d}_i \mathbf{z}_{i-1} + 2\dot{\omega}_i \omega_i \times \mathbf{z}_{i-1} + \dot{\omega}_i \times \mathbf{r}_{i-1,i} + \omega_i \times (\omega_i \times \mathbf{r}_{i-1,i}) & \text{prismatic joint} \\ \dddot{p}_{i-1} + \dddot{\omega}_i \times \mathbf{r}_{i-1,i} + \omega_i \times (\omega_i \times \mathbf{r}_{i-1,i}) & \text{rotational joint} \end{cases}
\]

\[
\dddot{p}_{C_i} = \dddot{p}_i + \dddot{\omega}_i \times \mathbf{r}_{i,C_i} + \omega_i \times (\omega_i \times \mathbf{r}_{i,C_i}) \quad \text{center of mass}
\]

Note: derivative of a vector \( a_i \) attached to the moving frame \( i \)

\[
\frac{d}{dt} a_i(t) = \frac{d}{dt} R_i(t) a_i^i = S(\omega_i(t)) R_i(t) a_i^i = \omega_i(t) \times a_i(t)
\]
A **forward recursion** of velocities and accelerations is made:

- initial conditions on $\omega_0, \mathbf{p}_0 - g_0, \omega_0$
- computation of $\omega_i, \dot{\omega}_i, \mathbf{p}_i, \ddot{\mathbf{p}}_i$

A **backward recursion** of forces and moments is made:

- terminal conditions on $f_{n+1}$ and $\mu_{n+1}$
- computations:

$$ f_i = f_{i+1} + m_i \ddot{\mathbf{p}}_{C_i} $$

$$ \mu_i = -f_i \times (r_{i-1,i} + r_{i,C_i}) + \mu_{i+1} + f_{i+1} \times r_{i,C_i} + l_i \dot{\omega}_i + \omega_i \times (l_i \omega_i) $$

The generalized force at joint $i$ is computed:

$$ \tau_i = \begin{cases} 
  f_i^T \mathbf{z}_{i-1} + F_{vi} \dot{d}_i + f_{si} & \text{prismatic joint} \\
  \mu_i^T \mathbf{z}_{i-1} + F_{vi} \dot{\theta}_i + f_{si} & \text{rotational joint} 
\end{cases} $$
Local reference frames

Up to now we have supposed that all the vectors are referred to the base frame.

It is more convenient to express the vectors with respect to the current frame on link $i$. In this way, vectors $r_{i-1,i}$ and $r_{i,Ci}$ and the inertia tensor $I_i$ are constant, which makes the algorithm computationally more efficient.

The equations are modified in some terms (we need to multiply the vectors by suitable rotation matrices) but nothing changes in the nature of the method.
Let us consider again a two-link planar manipulator with rotational joints, whose model has been already derived with the Euler-Lagrange method:

- masses: \( m_1 \) and \( m_2 \)
- lengths: \( a_1 \) and \( a_2 \)
- distances of the centers of mass from the joint axes: \( l_1 \) and \( l_2 \)
- moments of inertia around axes passing through the centers of mass and parallel to \( z_0 \): \( I_1 \) and \( I_2 \)

Initial conditions for the forward recursion of velocities and accelerations:

\[
\ddot{p}_0^0 - g_0^0 = \begin{bmatrix} 0 & g & 0 \end{bmatrix}^T, \quad \omega_0^0 = \dot{\omega}_0^0 = 0
\]

Initial conditions for the backward recursion of forces and moments:

\[
f_3^3 = 0, \quad \mu_3^3 = 0
\]
Definition of vectors and matrices

Let us refer all the quantities to the current frame on the link. We derive these constant vectors:

\[
\begin{align*}
\mathbf{r}_{1,C_1}^1 &= \begin{bmatrix} l_1 - a_1 \\ 0 \\ 0 \end{bmatrix}, & \mathbf{r}_{0,1}^1 &= \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix}, & \mathbf{r}_{2,C_2}^2 &= \begin{bmatrix} l_2 - a_2 \\ 0 \\ 0 \end{bmatrix}, & \mathbf{r}_{1,2}^2 &= \begin{bmatrix} a_2 \\ 0 \end{bmatrix}
\end{align*}
\]

The rotation matrices are:

\[
\begin{align*}
\mathbf{R}_1^0 &= \begin{bmatrix} c_1 & -s_1 & 0 \\ s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{R}_1^1 &= \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \mathbf{R}_3^2 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{align*}
\]
Forward recursion: link 1

\[ \omega_1 = R_{1}^{0T} (\omega_0 + \dot{\theta}_1 z_0) = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \]

\[ \dot{\omega}_1 = R_{1}^{0T} (\dot{\omega}_0 + \dot{\theta}_1 z_0 + \dot{\theta}_1 \omega_0 \times z_0) = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 \end{bmatrix} \]

\[ \ddot{p}_1 = R_{1}^{0T} \ddot{p}_0 + \dot{\omega}_1 \times r_{0,1} + \omega_1 \times (\omega_1 \times r_{0,1}) = \begin{bmatrix} -a_1 \dot{\theta}_1^2 + gs_1 \\ a_1 \dot{\theta}_1 + gc_1 \\ 0 \end{bmatrix} \]

\[ \ddot{p}_{C_1} = \ddot{p}_1 + \dot{\omega}_1 \times r_{1,C_1} + \omega_1 \times (\omega_1 \times r_{1,C_1}) = \begin{bmatrix} -l_1 \dot{\theta}_1^2 + gs_1 \\ l_1 \dot{\theta}_1 + gc_1 \\ 0 \end{bmatrix} \]
Forward recursion: link 2

\[
\begin{align*}
\omega_2^2 &= R_2^{1T} (\omega_1^1 + \dot{\theta}_2 z_0) = \\
&= \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix} \\

\ddot{\omega}_2 &= R_2^{1T} (\ddot{\omega}_1^1 + \ddot{\theta}_2 z_0 + \dot{\theta}_2 \omega_1^1 \times z_0) = \\
&= \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_1 + \ddot{\theta}_2 \end{bmatrix} \\

\dddot{p}_2 &= R_2^{1T} \dddot{p}_1^1 + \ddot{\omega}_2^2 \times r_{1,2}^2 + \omega_2^2 \times (\omega_2^2 \times r_{1,2}^2) = \\
&= \begin{bmatrix} a_1 s_2 \dddot{\theta}_1 - a_1 c_2 \dddot{\theta}_1^2 - a_2 (\dddot{\theta}_1 + \dddot{\theta}_2)^2 + g s_{12} \\ a_1 c_2 \dddot{\theta}_1 + a_2 (\dddot{\theta}_1 + \dddot{\theta}_2) + a_1 s_2 \dddot{\theta}_1^2 + g c_{12} \\ 0 \end{bmatrix} \\

\dddot{p}_{C2} &= \dddot{p}_2^2 + \ddot{\omega}_2^2 \times r_{2,C2}^2 + \omega_2^2 \times (\omega_2^2 \times r_{2,C2}^2) = \\
&= \begin{bmatrix} a_1 s_2 \dddot{\theta}_1 - a_1 c_2 \dddot{\theta}_1^2 - l_2 (\dddot{\theta}_1 + \dddot{\theta}_2)^2 + g s_{12} \\ a_1 c_2 \dddot{\theta}_1 + l_2 (\dddot{\theta}_1 + \dddot{\theta}_2) + a_1 s_2 \dddot{\theta}_1^2 + g c_{12} \\ 0 \end{bmatrix}
\end{align*}
\]
Backward recursion: link 2

\[ f_2^2 = R_3^2 f_3^3 + m_2 \ddot{p}_{C_2}^2 = m_2 \ddot{p}_{C_2}^2 = \begin{bmatrix} m_2 \left( a_1 s_2 \ddot{\theta}_1 - a_1 c_2 \ddot{s}_2 - l_2 \left( \ddot{\theta}_1 + \ddot{\theta}_2 \right)^2 + g s_{12} \right) \\
\quad m_2 \left( a_1 c_2 \ddot{\theta}_1 + l_2 \left( \ddot{\theta}_1 + \ddot{\theta}_2 \right) a_1 s_2 \ddot{s}_2 + g c_{12} \right) \\
\quad 0 \end{bmatrix} \]

\[ \mu_2^2 = -f_2^2 \times \left( r_{1,2}^2 + r_{2,c_2}^2 \right) + R_3^2 \mu_3^3 + R_3^2 f_3^3 \times r_{2,c_2}^2 + l_2^2 \dot{\omega}_2^2 + \omega_2^2 \times \left( \dot{l}_2^2 \omega_2^2 \right) = \]

\[ = \begin{bmatrix} l_2 + m_2 l_2^2 \left( \ddot{\theta}_1 + \ddot{\theta}_2 \right) + m_2 a_1 l_2 c_2 \ddot{s}_2 + m_2 a_1 l_2 s_2 \ddot{\theta}_1^2 + m_2 l_2 g c_{12} \end{bmatrix} \]

\[ \tau_2 = \mu_2^2 T R_2^{1T} z_0 = \]

\[ = \left( l_2 + m_2 \left( l_2^2 + a_1 l_2 c_2 \right) \right) \ddot{\theta}_1 + \left( l_2 + m_2 l_2^2 \right) \ddot{\theta}_2 + m_2 a_1 l_2 s_2 \ddot{s}_2 + m_2 l_2 g c_{12} \]

(identical to the equation obtained with Euler-Lagrange method).
Backward recursion: link 1

\[ f_1^1 = R_2^1 f_2^2 + m_2 \ddot{p}_C^1 = \left[ -m_2 l_2 s_2 \left( \ddot{\theta}_1 + \ddot{\theta}_2 \right) - m_1 l_1 \ddot{\theta}_1 - m_2 a_1 \ddot{\theta}_1 - m_2 l_2 c_2 \left( \ddot{\theta}_1 + \ddot{\theta}_2 \right)^2 + \left( m_1 + m_2 \right) g s_1 \\
\left( m_1 l_1 + m_2 a_1 \right) \ddot{\theta}_1 + m_2 l_2 c_2 \left( \ddot{\theta}_1 + \ddot{\theta}_2 \right) - m_2 l_2 s_2 \left( \ddot{\theta}_1 + \ddot{\theta}_2 \right)^2 + \left( m_1 + m_2 \right) g c_1 \right] \]

\[ \mu_1^1 = -f_1^1 \times \left( r_{0,1}^1 + r_{1,C_1}^1 \right) + R_2^1 \mu_2 + R_2^1 f_2^2 \times r_{1,C_1}^1 + l_1^1 \omega_1 + \omega_1^1 \times \left( l_1^1 \omega_1^1 \right) = \left[ \begin{array}{c}
* \\
\left( l_1 + m_2 a_1^2 + m_1 l_1^2 + m_2 a_1 l_2 c_2 \right) \ddot{\theta}_1 + \left( l_2 + m_2 a_1 l_2 c_2 + m_2 l_2^2 \right) \ddot{\theta}_2 \\
+ m_2 a_1 l_2 s_2 \ddot{\theta}_1 - m_2 a_1 l_2 s_2 \left( \ddot{\theta}_1 + \ddot{\theta}_2 \right)^2 + \left( m_1 l_1 + m_2 a_1 \right) g c_1 + m_2 l_2 g c_{12} \end{array} \right] \]

\[ \tau_1 = \mu_1^{1T} R_1^{0T} z_0 = \left( l_1 + m_1 l_1^2 + l_2 + m_2 \left( a_1^2 + l_2^2 + 2a_1 l_2 c_2 \right) \right) \ddot{\theta}_1 + \left( l_2 + m_2 \left( l_2^2 + a_1 l_2 c_2 \right) \right) \ddot{\theta}_2 \\
- 2m_2 a_1 l_2 s_2 \ddot{\theta}_1 \ddot{\theta}_2 - m_2 a_1 l_2 s_2 \ddot{\theta}_2 + \left( m_1 l_1 + m_2 a_1 \right) g c_1 + m_2 l_2 g c_{12} \]

(identical to the equation obtained with Euler-Lagrange method).
Euler-Lagrange vs. Newton-Euler

**Euler-Lagrange formulation**

- it is systematic and easy to understand
- it returns the equations of motion in an analytic and compact form, separating the inertia matrix, the Coriolis and centrifugal terms, the gravitational terms. All these elements are useful for the design of a model based controller
- it lends itself to the introduction into the model of more complex effects (like joint or link deformation)

**Newton-Euler formulation**

- it is a computationally efficient recursive method
Direct and inverse dynamics

\[ B(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + F_v \dot{q} + f_s(q, \dot{q}) = \tau \]

Direct dynamics
For given joint torques \( \tau(t) \), determine the joint accelerations \( \ddot{q}(t) \) and, if initial positions \( q(t_0) \) and velocities \( \dot{q}(t_0) \) are known, the positions \( q(t) \) and the velocities \( \dot{q}(t) \).

- Problem whose solution is useful in order to compute the simulation model of the robot manipulator
- It can be solved both with Euler-Lagrange and with Newton-Euler approaches

Inverse dynamics
For given accelerations \( \ddot{q}(t) \), velocities \( \dot{q}(t) \) and positions \( q(t) \) determine the joint torques \( \tau(t) \) needed for motion generation.

- Problem whose solution is useful for trajectory planning and model based control
- It can be efficiently solved with the Newton-Euler formulation

Control of industrial robots – Robot dynamics – P. Rocco [51]
Computation of direct and inverse dynamics

Computation of the inverse dynamics can be easily done both with the Euler-Lagrange method and with the Newton-Euler one.

As for the computation of the direct dynamics, let us rewrite the dynamic model of the manipulator in these terms:

\[ B(q) \ddot{q} + n(q, \dot{q}) = \tau \]

where:

\[ n(q, \dot{q}) = C(q, \dot{q}) \dot{q} + g(q) + F_v \dot{q} + f_s(q, \dot{q}) \]

We thus have to numerically integrate the explicit system of differential equations:

\[ \ddot{q} = B(q)^{-1}(\tau - n(q, \dot{q})) \]

where all the elements needed to build the system are directly computed by the Euler-Lagrange method.
Computation of direct and inverse dynamics

How to compute the direct dynamics with the Newton-Euler method?

Newton-Euler script (Matlab, C, …):

$$\tau = \text{NE}(q, \dot{q}, \ddot{q})$$

With the current values of $q$ and $\dot{q}$, a first iteration of the script is performed, setting $\ddot{q} = 0$. In this way the torques $\tau$ computed by the method directly return the vector $n$.

Then we set $g_0 = 0$ inside the script (in order to eliminate the gravitational effects) and $\dot{q} = 0$ (in order to eliminate Coriolis, centrifugal and friction effects). $n$ iterations of the script are performed, with $\ddot{q}_i = 1$ and $\ddot{q}_j = 0, j \neq i$. This way matrix $B$ is formed column by column and all elements to form the system of equations are available.