FOUNDATIONS OF ROBOTICS
Dynamics

B. Siciliano, L. Villani, G. Oriolo, A. De Luca

These slides are distributed as companion material to the textbook
Foundations of Robotics by B. Siciliano, L. Villani, G. Oriolo, A. De Luca
published by Springer Nature.
Modification is not allowed without permission

Outline

- Direct Dynamics and Inverse Dynamics
- Euler–Lagrange Formulation
- Properties of Dynamic Model
- Dynamic Model of Simple Manipulator Structures
- Identification of Dynamic Parameters
- Newton–Euler Formulation
- Summary of Dynamic Formulations
- Dynamic Scaling of Trajectories
- Task-Space Dynamic Model
- Constrained Dynamic Model

Direct Dynamics and Inverse Dynamics

- The dynamic model of a robot provides a description of the relationship between the generalized forces $\boldsymbol{\xi}$ acting on the system and the motion of the structure, as described by the time evolution of the generalized coordinates $\boldsymbol{q}$
- In this sense, dynamics connects the causes (input) to their effects (output) on the mechanical system, while kinematics is intended to describe robot motion without accounting for its cause
- Consider for example a 2R planar arm with revolute joints, initially at rest on a horizontal plane (without gravity)
- A desired motion is specified for the second joint as $q_2(t)$, with $t \in [0, T]$, while the first joint should remain at rest
- At a kinematic level, one can generate $\dot{q}_2(t)$ or $\ddot{q}_2(t)$, with $\dot{q}_1(t) = \ddot{q}_1(t) = 0$
- At a dynamic level, one needs a suitable torque $\tau_2(t)$, but also a torque $\tau_1(t)$ to avoid the first link moving due to the inertial couplings

- Availability of a dynamic model is important for the following reasons
  - used at the design stage to optimize the distribution of masses and assess the inertial and gravitational loads acting on the joints and links
  - allows proper choice of the size of the motors to deliver the torque needed for the desired level of velocity/acceleration of the joints or of end-effector tools
  - used as a digital twin for simulating the behavior of a physical robot under different inputs, both in open space and during interaction with the environment
  - used to plan feasible optimal trajectories and interaction forces, and to design control laws that are provably good for achieving the target robot behaviors
- Assuming for simplicity that neither interaction nor dissipative forces are present ($\boldsymbol{\xi} = \boldsymbol{\tau}$), the dynamic model of a robot with n generalized coordinates $\boldsymbol{q}$ can be expressed in the generic form

$$\boldsymbol{\phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{\tau}$$

  a set of n second-order differential equations, with actuating torques $\boldsymbol{\tau} \in I\!R^n$ on the right-hand side
- This form allows formulating the direct dynamics and inverse dynamics problems

- The direct dynamics problem consists in determining, at any time t and with the robot in a state $(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$, the joint accelerations $\ddot{\boldsymbol{q}}(t)$ resulting from the application of known joint torques $\boldsymbol{\tau}(t)$
- A simulation of robot motion over a time interval $[t_0, t_f]$ is the repeated solution of the direct dynamics problem with time step T_s, followed by numerical integration of the computed $\ddot{\boldsymbol{q}}(t)$ to obtain $\dot{\boldsymbol{q}}(t + T_s)$ and $\boldsymbol{q}(t + T_s)$
- The experimental solution is obtained by commanding the chosen torque profiles $\boldsymbol{\tau}(t)$ and recording the measured $\boldsymbol{q}(t)$
- If the robot is not available, the solution $\boldsymbol{q}(t)$ can be found by resorting to the numerical simulation of the dynamic model

- The inverse dynamics problem consists in determining, at any time t, the joint torque $\boldsymbol{\tau}(t)$ that is needed to generate a joint acceleration $\ddot{\boldsymbol{q}}(t)$ when the robot is at a configuration $\boldsymbol{q}(t)$ with velocity $\dot{\boldsymbol{q}}(t)$
- The straightforward solution requires the algebraic evaluation of the dynamic model by plugging in $(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t))$
- Two main formulations are considered
  - Euler–Lagrange (EL) method, an energy-based approach
  - Newton–Euler (NE) method, based on the inertial balance of forces and moments along the structure

Euler–Lagrange Formulation

- In the EL formulation, the equations of motion are derived in a systematic way from energy arguments
- Once a set of generalized coordinates $\boldsymbol{q} = (q_1, \ldots, q_n)$ has been chosen, the Lagrangian function of the mechanical system is defined as

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{U}(\boldsymbol{q})$$

  where $\mathcal{T}$ is the kinetic energy and $\mathcal{U}$ is the potential energy of the system
- EL equations

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \xi_i \qquad i = 1, \ldots, n$$

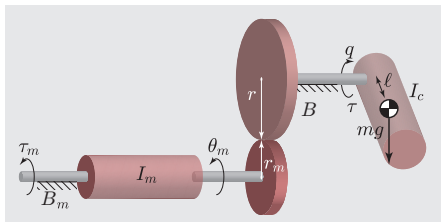
  where ξ_i is the nonconservative generalized force that affects coordinate q_i
- Vector form

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}}\right)^T = \boldsymbol{\xi}$$

Example

- Single link with gear
  - reduction ratio

$$n_r = \frac{r}{r_m} = \frac{\dot{\theta}_m}{\dot{q}} > 1$$



  - energies

$$\mathcal{T} = \frac{1}{2} I \dot{q}^2 + \frac{1}{2} I_m \dot{\theta}_m^2 = \frac{1}{2} \left(I + I_m n_r^2 \right) \dot{q}^2$$

$$\mathcal{U} = mgh = mg\ell(1 - \cos q) \qquad \Rightarrow \qquad \mathcal{L} = \frac{1}{2} \left(I + I_m n_r^2 \right) \dot{q}^2 - mg\ell(1 - \cos q)$$

$$\left(I + I_m n_r^2 \right) \ddot{q} + mg\ell \sin q = \xi \qquad \text{with} \qquad \xi = \tau - \left(B + B_m n_r^2 \right) \dot{q}$$

$$\left(I + I_m n_r^2 \right) \ddot{q} + mg\ell \sin q + \left(B + B_m n_r^2 \right) \dot{q} = \tau$$

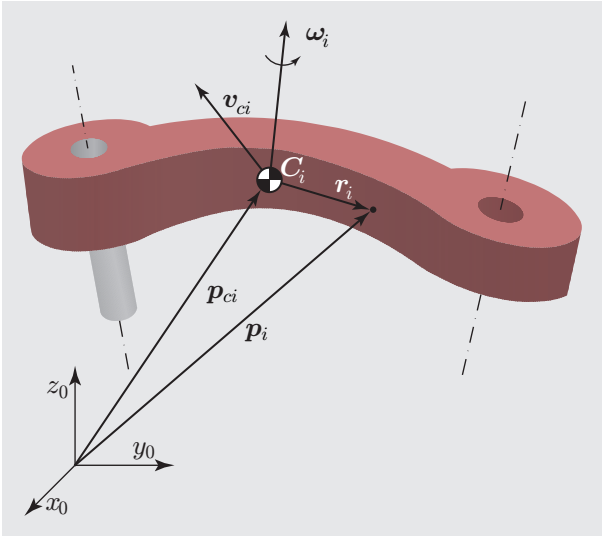
Kinetic Energy

- Assume that all elements of the manipulator (links, motors, transmissions) are rigid
- The total kinetic energy is given by the sum of the contributions due to the motion of each link and of each actuator

$$\mathcal{T} = \sum_{i=1}^{n} (\mathcal{T}_i + \mathcal{T}_{mi})$$

  - $\mathcal{T}_i$ is the kinetic energy of link i
  - $\mathcal{T}_{mi}$ is the kinetic energy of the corresponding motor i

- Kinematic description of link for kinetic energy computation



- Link contributions
- The generic i-th link is a body $\mathcal{B}_i$ with mass

$$m_i = \int_{\mathcal{B}_i} dm$$

- The position $\boldsymbol{p}_{ci} \in I\!R^3$ of its center of mass (CoM) is

$$\boldsymbol{p}_{ci} = \frac{1}{m_i} \int_{\mathcal{B}_i} \boldsymbol{p}_i \, dm$$

  - $\boldsymbol{p}_i \in I\!R^3$ is the position of an elementary particle of mass dm in the body $\mathcal{B}_i$
  - all vectors are expressed in the base frame of the robot

- The kinetic energy of link i is

$$\mathcal{T}_i = \frac{1}{2} \int_{\mathcal{B}_i} \boldsymbol{v}_i^T \boldsymbol{v}_i \, dm$$

  where $\boldsymbol{v}_i = \dot{\boldsymbol{p}}_i$ is the velocity of the elementary particle of mass dm

- This velocity can be expressed as

$$\boldsymbol{v}_i = \boldsymbol{v}_{ci} + \boldsymbol{\omega}_i \times \boldsymbol{r}_i = \boldsymbol{v}_{ci} + \boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i$$

  - $\boldsymbol{v}_{ci} = \dot{\boldsymbol{p}}_{ci}$
  - $\boldsymbol{\omega}_i$ is the angular velocity of link i
  - $\boldsymbol{r}_i = \boldsymbol{p}_i - \boldsymbol{p}_{ci}$

- The translational contribution is

$$\frac{1}{2}\int_{\mathcal{B}_i} \boldsymbol{v}_{ci}^T \boldsymbol{v}_{ci}\, dm = \frac{1}{2}\, m_i \boldsymbol{v}_{ci}^T \boldsymbol{v}_{ci}$$

- The mutual contribution is

$$\frac{1}{2}\left(2\int_{\mathcal{B}_i} \boldsymbol{v}_{ci}^T \boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\, dm\right) = \boldsymbol{v}_{ci}^T \boldsymbol{S}(\boldsymbol{\omega}_i)\int_{\mathcal{B}_i} (\boldsymbol{p}_i - \boldsymbol{p}_{ci})\, dm = 0$$

being

$$\int_{\mathcal{B}_i} \boldsymbol{p}_i\, dm = \int_{\mathcal{B}_i} dm$$

- The rotational contribution is

$$\frac{1}{2}\int_{\mathcal{B}_i} \boldsymbol{r}_i^T \boldsymbol{S}^T(\boldsymbol{\omega}_i)\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i\, dm = \frac{1}{2}\,\boldsymbol{\omega}_i^T\left(\int_{\mathcal{B}_i} \boldsymbol{S}^T(\boldsymbol{r}_i)\boldsymbol{S}(\boldsymbol{r}_i)\, dm\right)\boldsymbol{\omega}_i$$

- The angular velocity of link i is expressed in a frame attached to the link

$$^i\boldsymbol{\omega}_i = \boldsymbol{R}_i^T(\boldsymbol{q})\,\boldsymbol{\omega}_i$$

$$\boldsymbol{\omega}_i^T \boldsymbol{S}^T(\boldsymbol{r}_i)\boldsymbol{S}(\boldsymbol{r}_i)\boldsymbol{\omega}_i = {}^i\boldsymbol{\omega}_i^T \boldsymbol{S}^T({}^i\boldsymbol{r}_i)\boldsymbol{S}({}^i\boldsymbol{r}_i)^i\boldsymbol{\omega}_i$$

  where property $\boldsymbol{S}(\boldsymbol{\omega}_i)\boldsymbol{r}_i = -\boldsymbol{S}(\boldsymbol{r}_i)\boldsymbol{\omega}_i$ has been used

- Vector $^i\boldsymbol{r}_i = (r_{ix}, r_{iy}, r_{iz})$ does not depend on $\boldsymbol{q}$

$$\boldsymbol{S}({}^i\boldsymbol{r}_i) = \begin{pmatrix} 0 & -r_{iz} & r_{iy} \\ r_{iz} & 0 & -r_{ix} \\ -r_{iy} & r_{ix} & 0 \end{pmatrix}$$

- The rotational contribution can be written as

$$\frac{1}{2}\,{}^{i}\boldsymbol{\omega}_i^T \left(\int_{\mathcal{B}_i} \boldsymbol{S}^T({}^{i}\boldsymbol{r}_i)\boldsymbol{S}({}^{i}\boldsymbol{r}_i)\,dm \right) {}^{i}\boldsymbol{\omega}_i = \frac{1}{2}\,{}^{i}\boldsymbol{\omega}_i^T\,{}^{i}\boldsymbol{I}_i\,{}^{i}\boldsymbol{\omega}_i$$

- The constant symmetric matrix represents the inertia tensor relative to the CoM

$$\begin{aligned} {}^{i}\boldsymbol{I}_i &= \begin{pmatrix} \int \left(r_{iy}^2 + r_{iz}^2\right) dm & -\int r_{ix}r_{iy}\,dm & -\int r_{ix}r_{iz}\,dm \\ * & \int \left(r_{ix}^2 + r_{iz}^2\right) dm & -\int r_{iy}r_{iz}\,dm \\ * & * & \int \left(r_{ix}^2 + r_{iy}^2\right) dm \end{pmatrix} \\ &= \begin{pmatrix} I_{ixx} & I_{ixy} & I_{ixz} \\ * & I_{iyy} & I_{iyz} \\ * & * & I_{izz} \end{pmatrix} \end{aligned}$$

- Properties of the link inertia tensor
  - it is the inertia tensor relative to the CoM of link i, expressed in a frame attached to the link
  - if the frame axes coincide with the central axes of inertia, the off-diagonal terms are zero and ${}^{i}\boldsymbol{I}_i$ is diagonal
  - the rotational contribution can also be written in the base frame as

$$\frac{1}{2}\,{}^{i}\boldsymbol{\omega}_i^T\,{}^{i}\boldsymbol{I}_i\,{}^{i}\boldsymbol{\omega}_i = \frac{1}{2}\,\boldsymbol{\omega}_i^T\boldsymbol{I}_i\,\boldsymbol{\omega}_i$$

    with the link inertia tensor in the base frame given by

$$\boldsymbol{I}_i = \boldsymbol{R}_i(\boldsymbol{q})\,{}^{i}\boldsymbol{I}_i\,\boldsymbol{R}_i^T(\boldsymbol{q})$$

- The kinetic energy of link i is

$$\mathcal{T}_i = \frac{1}{2}\,m_i\,\boldsymbol{v}_{ci}^T\boldsymbol{v}_{ci} + \frac{1}{2}\,\boldsymbol{\omega}_i^T\boldsymbol{I}_i\,\boldsymbol{\omega}_i$$

- The CoM linear velocity and the link angular velocity can be written as

$$\boldsymbol{v}_{ci} = \boldsymbol{j}^i_{p1}\dot{q}_1 + \cdots + \boldsymbol{j}^i_{pi}\dot{q}_i = \boldsymbol{J}^i_p\,\dot{\boldsymbol{q}}$$
$$\boldsymbol{\omega}_i = \boldsymbol{j}^i_{o1}\dot{q}_1 + \cdots + \boldsymbol{j}^i_{oi}\dot{q}_i = \boldsymbol{J}^i_o\,\dot{\boldsymbol{q}}$$

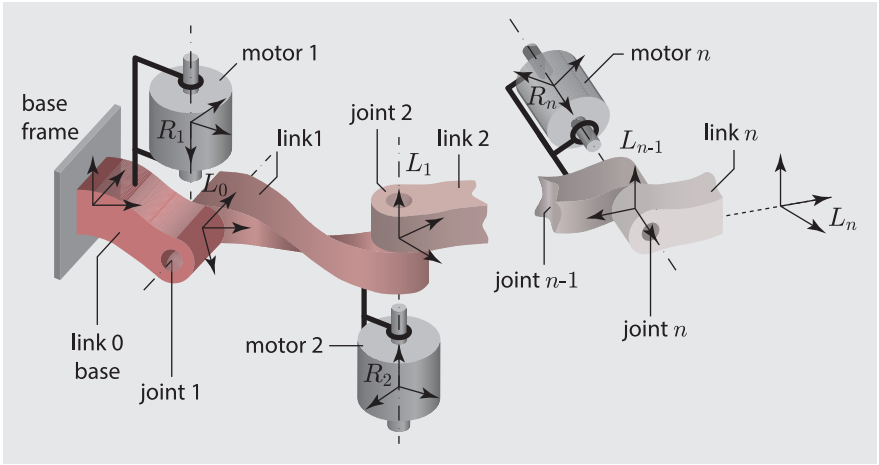
- By construction, the last $n-i$ columns of $\boldsymbol{J}^i_p$ and $\boldsymbol{J}^i_o$ are zero

$$\begin{pmatrix} \boldsymbol{j}^i_{pj} \\ \boldsymbol{j}^i_{oj} \end{pmatrix} = \begin{cases} \begin{pmatrix} \boldsymbol{z}_{j-1} \\ \boldsymbol{0} \end{pmatrix} & \text{for a prismatic joint} \\ \begin{pmatrix} \boldsymbol{z}_{j-1} \times (\boldsymbol{p}_{ci} - \boldsymbol{p}_{j-1}) \\ \boldsymbol{z}_{j-1} \end{pmatrix} & \text{for a revolute joint} \end{cases}$$

- Making explicit the dependence on $\boldsymbol{q}$, the kinetic energy is written as

$$\mathcal{T}_i = \frac{1}{2}\, m_i\, \dot{\boldsymbol{q}}^T \boldsymbol{J}_p^{i\,T}(\boldsymbol{q}) \boldsymbol{J}^i_p(\boldsymbol{q})\, \dot{\boldsymbol{q}} + \frac{1}{2}\, \dot{\boldsymbol{q}}^T \boldsymbol{J}_o^{i\,T}(\boldsymbol{q}) \boldsymbol{R}_i(\boldsymbol{q})\, {}^i\boldsymbol{I}_i\, \boldsymbol{R}_i^T(\boldsymbol{q})\, \boldsymbol{J}^i_o(\boldsymbol{q})\, \dot{\boldsymbol{q}}$$

- Motor contributions



- Consider n electric motors, each driving a single joint
- Standard assumptions
  - A1. the rotors are uniform bodies having their CoM on the rotation axis
  - A2. motor i moves link i and is mounted before the link itself, usually on link $i-1$
  - A3. the angular velocity of each rotor is only due to its own spinning
- The static mass of each motor can be included in the mass of the carrying link, modifying the link CoM and inertia
- The transmission is rigid, so the rotor speed is proportional to the joint speed

$$\dot{\theta}_{mi} = n_{ri}\dot{q}_i$$

- The angular kinetic energy of rotor i is

$$\mathcal{T}_{mi} = \frac{1}{2} I_{mi} n_{ri}^2 \dot{q}_i^2$$

where I_{mi} is the rotor inertia about its spinning axis

- The total kinetic energy of a manipulator is written as the quadratic form

$$\mathcal{T} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}(\boldsymbol{q})\dot{q}_i\dot{q}_j = \frac{1}{2}\,\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- $\boldsymbol{M}(\boldsymbol{q})$ is the $n \times n$ robot inertia matrix, which can be decomposed as

$$\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{M}_l(\boldsymbol{q}) + \boldsymbol{M}_m$$

- Configuration-dependent inertia matrix of the links

$$\boldsymbol{M}_l(\boldsymbol{q}) = \sum_{i=1}^{n}\Big(m_i\boldsymbol{J}_p^{i\,T}(\boldsymbol{q})\boldsymbol{J}_p^i(\boldsymbol{q}) + \boldsymbol{J}_o^{i\,T}(\boldsymbol{q})\boldsymbol{R}_i(\boldsymbol{q})^i\boldsymbol{I}_i\,\boldsymbol{R}_i^T(\boldsymbol{q})\,\boldsymbol{J}_o^i(\boldsymbol{q})\Big)$$

- Constant inertia mastrix of the motors

$$\boldsymbol{M}_m = \boldsymbol{N}_r\boldsymbol{I}_m\boldsymbol{N}_r \qquad \boldsymbol{N}_r = \mathrm{diag}\{n_{r1},\ldots,n_{rn}\} \qquad \boldsymbol{I}_m = \mathrm{diag}\{I_{m1},\ldots,I_{mn}\}$$

- Properties of robot inertia matrix
  - $\boldsymbol{M}(\boldsymbol{q})$ is symmetric
  - $\boldsymbol{M}(\boldsymbol{q})$ is positive definite
  - for serial manipulators, $\boldsymbol{M}(\boldsymbol{q})$ is independent of the first joint variable q_1
  - for serial manipulators, the element m_{nn} is constant
  - efficient computation can be obtained using a recursion for the link velocities

Potential Energy

- Under the rigid-structure assumption, the potential energy $\mathcal{U}$ is only due to gravity
- The total potential energy is the sum of the link contributions

$$\mathcal{U} = \sum_{i=1}^{n} \mathcal{U}_i$$

- The potential energy of the motors does not appear explicitly because each motor mass is included in the mass of the link on which it is mounted
  - due to assumption A1, when the motors are spinning the CoMs of the rotors do not change position (height in the gravity potential field)

- The potential energy of link i is

$$\mathcal{U}_i = -\int_{\mathcal{B}_i} \boldsymbol{g}_0^T \boldsymbol{p}_i \, dm = -m_i \, \boldsymbol{g}_0^T \boldsymbol{p}_{ci}$$

  - $\boldsymbol{g}_0$ is the gravity acceleration vector expressed in the base frame
  - $\boldsymbol{g}_0 = (0, 0, -g)$ when the upward vertical axis is z_0

- The total potential energy can be written as

$$\mathcal{U} = -\sum_{i=1}^{n} m_i \, \boldsymbol{g}_0^T \boldsymbol{p}_{ci} + \mathcal{U}_0$$

  - $\mathcal{U}$ depends on the joint variables $\boldsymbol{q}$ through the positions $\boldsymbol{p}_{ci}$
  - $\mathcal{U}_0$ is a constant used to set the zero level of potential energy at a convenient configuration

Equations of Motion

- Lagrangian

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{U}(\boldsymbol{q})$$

- The Euler–Lagrange derivatives are

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right)^T = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

$$\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}}\right)^T = \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\left(\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\right)\right)^T - \boldsymbol{g}(\boldsymbol{q})$$

with the gravitational vector

$$\boldsymbol{g}(\boldsymbol{q}) = \left(\frac{\partial \mathcal{U}}{\partial \boldsymbol{q}}\right)^T = \nabla_{\boldsymbol{q}}\, \mathcal{U}(\boldsymbol{q})$$

- The dynamic model is

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\xi}$$

with the velocity vector

$$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \frac{1}{2}\left(\frac{\partial}{\partial \boldsymbol{q}}\left(\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\right)\right)^T$$

- Componentwise, we have for $i = 1, \ldots, n$

$$\begin{aligned} c_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \sum_{j=1}^{n} \frac{dm_{ij}}{dt}\,\dot{q}_j - \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{\partial m_{jk}}{\partial q_i}\,\dot{q}_j \dot{q}_k \\ &= \sum_{j=1}^{n}\sum_{k=1}^{n}\left(\frac{\partial m_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial m_{jk}}{\partial q_i}\right)\dot{q}_j \dot{q}_k \end{aligned}$$

$$g_i(\boldsymbol{q}) = -\sum_{j=1}^{n} m_j\, \boldsymbol{g}_0^T \frac{\partial \boldsymbol{p}_{cj}}{\partial q_i} = -\sum_{j=1}^{n} m_j\, \boldsymbol{g}_0^T\, \boldsymbol{J}_{pi}^j(\boldsymbol{q})$$

- Putting all terms together

$$\sum_{j=1}^{n} m_{ij}(\boldsymbol{q})\,\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} c_{ijk}(\boldsymbol{q})\,\dot{q}_j\dot{q}_k + g_i(\boldsymbol{q}) = \xi_i \qquad i = 1,\ldots,n$$

  where the coefficients $c_{ijk}(\boldsymbol{q})$ are the Christoffel symbols of the first kind

$$c_{ijk}(\boldsymbol{q}) = \frac{1}{2}\left(\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{jk}}{\partial q_i}\right)$$

- The model is linear in $\ddot{\boldsymbol{q}}$, quadratic in $\dot{\boldsymbol{q}}$, and nonlinear in $\boldsymbol{q}$

- Inertial terms
  - $m_{ii}(\boldsymbol{q})$ is the inertia seen at joint i with all other joints blocked
  - $m_{ij}(\boldsymbol{q})$ is inertial coupling between joints i and j, with $m_{ji} = m_{ij}$
- Velocity terms
  - $c_{ijj}(\boldsymbol{q})\dot{q}_j^2$ accounts for centrifugal effect on joint i due to joint j
  - $c_{iii} = 0$ since $\partial m_{ii}/\partial q_i = 0$
  - $c_{ijk}(\boldsymbol{q})\dot{q}_j\dot{q}_k$ accounts for Coriolis effect on joint i due to joints j and k
  - $c_{ijk} = c_{ikj}$
- Gravitational terms
  - $g_i(\boldsymbol{q})$ accounts for gravity effect on joint i
- Nonconservative terms
  - ξ_i in the right-hand side includes active and dissipative torques performing work on q_i

- Nonconservative torques

$$\boldsymbol{\xi} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{int}} + \boldsymbol{\tau}_B$$

  - $\boldsymbol{\tau}$: actuator torques produced by the motors
  - $\boldsymbol{\tau}_{\text{int}}$: interaction torques resulting from interaction forces and moments with the environment
  - $\boldsymbol{\tau}_B$: dissipative torques due to friction
- Viscous friction $\boldsymbol{\tau}_{B,v} = -\boldsymbol{B}_v \dot{\boldsymbol{q}}$
  - when considering viscous friction on both the motor and the link side of the transmission

$$\boldsymbol{B}_v = \boldsymbol{B}_j + \boldsymbol{N}_r \boldsymbol{B}_m \boldsymbol{N}_r$$

- Coulomb friction $\boldsymbol{\tau}_{B,c} = -\boldsymbol{B}_c \,\textbf{sign}\,(\dot{\boldsymbol{q}})$
- Static friction $\boldsymbol{\tau}_{B,s}$
- $\boldsymbol{B}_v$ and $\boldsymbol{B}_c$ are diagonal with nonnegative elements

$$\Downarrow$$

$$\boldsymbol{\xi} = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{int}} + \boldsymbol{\tau}_B = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{int}} - \boldsymbol{B}_v \dot{\boldsymbol{q}} - \boldsymbol{B}_c \,\textbf{sign}\,(\dot{\boldsymbol{q}}) - \boldsymbol{\tau}_{B,s}$$

- Resulting equations of motion

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{int}} + \boldsymbol{\tau}_B$$

- In the absence of interaction forces and friction

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

- Furthermore, for $i = 1, \ldots, n$

$$c_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}^T \boldsymbol{C}_i(\boldsymbol{q})\dot{\boldsymbol{q}}, \qquad i = 1, \ldots, n$$

  with the symmetric matrix $\boldsymbol{C}_i(\boldsymbol{q})$ given by

$$\boldsymbol{C}_i(\boldsymbol{q}) = \frac{1}{2}\left(\left(\frac{\partial \boldsymbol{m}_i}{\partial \boldsymbol{q}}\right) + \left(\frac{\partial \boldsymbol{m}_i}{\partial \boldsymbol{q}}\right)^T - \left(\frac{\partial \boldsymbol{M}}{\partial q_i}\right)\right)$$

  where $\boldsymbol{m}_i$ is the i-th column of $\boldsymbol{M}$

- Constant elements in $\boldsymbol{M}(\boldsymbol{q})$ do not contribute to $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$

Closed Kinematic Chains

- A closed-chain manipulator can be viewed through an equivalent tree-structured open-chain robot with n joints
- The closed-chain robot has m DoFs and $k = n - m$ independent constraints
- Assume the constraints can be explicitly solved as

$$\boldsymbol{\varrho} = \boldsymbol{\varrho}(\boldsymbol{q})$$

  - $\boldsymbol{q} \in I\!R^m$ independent joint variables
  - $\boldsymbol{\varrho} \in I\!R^k$ dependent joint variables
- Equivalent open-chain joint vector

$$\boldsymbol{q}_o = (\boldsymbol{q}, \boldsymbol{\varrho}) \in I\!R^n$$

- Assume also that only the independent joints $\boldsymbol{q}$ are actuated

- Consider a pair of twin robots
  - the closed-chain robot actuated only in $\boldsymbol{q}$
  - an equivalent open-chain robot fully actuated in $\boldsymbol{q}_o$
- Neglect dissipative and interaction forces
- Constraint forces perform no virtual work on admissible virtual displacements

$$\boldsymbol{\tau}^T \delta \boldsymbol{q} = \boldsymbol{\tau}_o^T \delta \boldsymbol{q}_o$$

  - $\boldsymbol{\tau} \in I\!R^m$ torques for the closed-chain robot
  - $\boldsymbol{\tau}_o \in I\!R^n$ torques for the open-chain twin

- Generalized velocities are related by

$$\dot{\boldsymbol{q}}_o = \boldsymbol{\Upsilon}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

with

$$\boldsymbol{\Upsilon}(\boldsymbol{q}) = \begin{pmatrix} \boldsymbol{I} \\ \dfrac{\partial \boldsymbol{\varrho}(\boldsymbol{q})}{\partial \boldsymbol{q}} \end{pmatrix}$$

and hence $\delta \boldsymbol{q}_o = \boldsymbol{\Upsilon}(\boldsymbol{q})\delta \boldsymbol{q}$

- Substituting $\delta \boldsymbol{q}_o = \boldsymbol{\Upsilon}(\boldsymbol{q})\delta \boldsymbol{q}$ into the virtual-work equality

$$\boldsymbol{\tau}^T \delta \boldsymbol{q} = \boldsymbol{\tau}_o^T \boldsymbol{\Upsilon}(\boldsymbol{q})\delta \boldsymbol{q}$$

yields the torque relationship

$$\boldsymbol{\tau} = \boldsymbol{\Upsilon}^T(\boldsymbol{q})\boldsymbol{\tau}_o$$

- The torque of the equivalent fully actuated open-chain robot is

$$\boldsymbol{\tau}_o = \boldsymbol{M}_o(\boldsymbol{q}_o)\ddot{\boldsymbol{q}}_o + \boldsymbol{c}_o(\boldsymbol{q}_o, \dot{\boldsymbol{q}}_o) + \boldsymbol{g}_o(\boldsymbol{q}_o)$$

- Use the time derivative of $\dot{\boldsymbol{q}}_o = \boldsymbol{\Upsilon}(\boldsymbol{q})\dot{\boldsymbol{q}}$

$$\ddot{\boldsymbol{q}}_o = \boldsymbol{\Upsilon}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{\Upsilon}}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- Combine torque mapping and open-chain dynamics

$$\boldsymbol{\tau} = \boldsymbol{\Upsilon}^T(\boldsymbol{q})\boldsymbol{M}_o(\boldsymbol{q}_o)\boldsymbol{\Upsilon}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{\Upsilon}^T(\boldsymbol{q})\left(\boldsymbol{M}_o(\boldsymbol{q}_o)\dot{\boldsymbol{\Upsilon}}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{c}_o(\boldsymbol{q}_o, \dot{\boldsymbol{q}}_o) + \boldsymbol{g}_o(\boldsymbol{q}_o)\right)$$

- Using $\boldsymbol{q}_o = (\boldsymbol{q}, \boldsymbol{\varrho}(\boldsymbol{q}))$ and $\dot{\boldsymbol{q}}_o = \boldsymbol{\Upsilon}(\boldsymbol{q})\dot{\boldsymbol{q}}$, this expression can be rewritten using only independent state variables $(\boldsymbol{q}, \dot{\boldsymbol{q}})$

- Closed-chain dynamic model
  - add motor inertia for the m actuated joints as an additive term
  - equations of motion in m independent coordinates

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

where

$$\begin{aligned}
\boldsymbol{M}(\boldsymbol{q}) &= \boldsymbol{\Upsilon}^T(\boldsymbol{q})\boldsymbol{M}_o(\boldsymbol{q})\boldsymbol{\Upsilon}(\boldsymbol{q}) + \boldsymbol{M}_m \\
\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \boldsymbol{\Upsilon}^T(\boldsymbol{q})\left(\boldsymbol{c}_o(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{M}_o(\boldsymbol{q})\dot{\boldsymbol{\Upsilon}}(\boldsymbol{q})\dot{\boldsymbol{q}}\right) \\
\boldsymbol{g}(\boldsymbol{q}) &= \boldsymbol{\Upsilon}^T(\boldsymbol{q})\,\boldsymbol{g}_o(\boldsymbol{q})
\end{aligned}$$

with the (reflected) motor inertias

$$\boldsymbol{M}_m = \boldsymbol{N}_r\boldsymbol{I}_m\boldsymbol{N}_r$$

Properties of Dynamic Model

- Two notable properties of the dynamic model are presented
  - skew-symmetry of a special matrix, for a suitable factorization of the velocity terms
  - linearity of the dynamic model in terms of a set of parameters
- These properties are useful for dynamic parameter identification and for designing control algorithms

Skew-Symmetry and Passivity

- The quadratic dependence on $\dot{\boldsymbol{q}}$ of $\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ implies

$$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$$

  with infinitely many valid factorizations, each with a corresponding matrix $\boldsymbol{N}$

- The particular choice

$$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \qquad \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{pmatrix} \dot{\boldsymbol{q}}^T \boldsymbol{C}_1(\boldsymbol{q}) \\ \dot{\boldsymbol{q}}^T \boldsymbol{C}_2(\boldsymbol{q}) \\ \vdots \\ \dot{\boldsymbol{q}}^T \boldsymbol{C}_n(\boldsymbol{q}) \end{pmatrix} \qquad c_{ik}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{j=1}^{n} c_{ijk}(\boldsymbol{q})\, \dot{q}_j$$

  with Christoffel symbols c_{ijk}, has the property that following matrix is skew-symmetric

$$\boldsymbol{S}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\,\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

$$\Rightarrow \quad \boldsymbol{x}^T \left(\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\,\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \right) \boldsymbol{x} = 0 \quad \text{for all } \boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{x} \in I\!R^n$$

- Proof

$$\dot{m}_{ij}(\boldsymbol{q}) = \sum_{k=1}^{n} \frac{\partial m_{ij}}{\partial q_k}\, \dot{q}_k$$

$$2\, c_{ij}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{k=1}^{n} 2\, c_{ikj}(\boldsymbol{q})\, \dot{q}_k = \sum_{k=1}^{n} \left(\frac{\partial m_{ij}}{\partial q_k} + \frac{\partial m_{ik}}{\partial q_j} - \frac{\partial m_{kj}}{\partial q_i} \right) \dot{q}_k$$

$$s_{ij}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{m}_{ij}(\boldsymbol{q}) - 2\, c_{ij}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{k=1}^{n} \left(\frac{\partial m_{kj}}{\partial q_i} - \frac{\partial m_{ik}}{\partial q_j} \right) \dot{q}_k$$

$$s_{ji}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{k=1}^{n} \left(\frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{jk}}{\partial q_i} \right) \dot{q}_k = -s_{ij}(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

- An exchange property is also valid in this case

$$\boldsymbol{C}(\boldsymbol{q}, \boldsymbol{v})\boldsymbol{w} = \boldsymbol{C}(\boldsymbol{q}, \boldsymbol{w})\boldsymbol{v} \quad \text{for all } \boldsymbol{v}, \boldsymbol{w} \in I\!R^n$$

- Principle of energy conservation
  - total energy (Hamiltonian function)

$$\mathcal{H}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \mathcal{U}(\boldsymbol{q}) = \frac{1}{2}\, \dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q}) \dot{\boldsymbol{q}} + \mathcal{U}(\boldsymbol{q})$$

  - its time derivative in the absence of nonconservative torques is

$$\begin{aligned}
\dot{\mathcal{H}} &= \dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \frac{1}{2}\, \dot{\boldsymbol{q}}^T \dot{\boldsymbol{M}}(\boldsymbol{q}) \dot{\boldsymbol{q}} + \frac{\partial \mathcal{U}}{\partial \boldsymbol{q}}\, \dot{\boldsymbol{q}} \\
&= -\, \dot{\boldsymbol{q}}^T \left(\boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \right) + \frac{1}{2}\, \dot{\boldsymbol{q}}^T \dot{\boldsymbol{M}}(\boldsymbol{q}) \dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T \boldsymbol{g}(\boldsymbol{q}) \\
&= \frac{1}{2}\, \dot{\boldsymbol{q}}^T \left(\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \right) \dot{\boldsymbol{q}}
\end{aligned}$$

- $\dot{\mathcal{H}} = 0$ is equivalent to

$$\dot{\boldsymbol{q}}^T \Big(\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \Big) \dot{\boldsymbol{q}} = 0 \quad \text{for all } \boldsymbol{q}, \dot{\boldsymbol{q}} \in I\!R^n$$

- This identity is related to the passivity property of the robot dynamics
- It is weaker than the skew-symmetry property obtained for $\boldsymbol{N} = \boldsymbol{C}$
- In the presence of nonconservative torques $\boldsymbol{\xi}$

$$\dot{\mathcal{H}} = \dot{\boldsymbol{q}}^T \boldsymbol{\xi}$$

- In particular
  - when $\boldsymbol{\tau} = \boldsymbol{0}$ and $\boldsymbol{\xi} = -\boldsymbol{B}_v \dot{\boldsymbol{q}}$, then $\dot{\mathcal{H}} = -\dot{\boldsymbol{q}}^T \boldsymbol{B}_v \dot{\boldsymbol{q}} \leq 0$ (pure dissipation of energy)
  - when $\boldsymbol{\tau} \neq \boldsymbol{0}$ and there is no friction, then $\dot{\mathcal{H}} = \dot{\boldsymbol{q}}^T \boldsymbol{\tau}$

Linearity in the Dynamic Parameters

- To consider both types of joints, a pivot point $\boldsymbol{p}_{pi}$ is conveniently defined as

$$\boldsymbol{p}_{pi} = \begin{cases} \boldsymbol{p}_i & \text{for a prismatic joint} \\ \boldsymbol{p}_{i-1} & \text{for a revolute joint} \end{cases}$$

- Velocity composition rule

$$\boldsymbol{v}_{ci} = \boldsymbol{v}_{pi} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{pi} = \boldsymbol{v}_{pi} - \boldsymbol{S}(\boldsymbol{r}_{pi})\boldsymbol{\omega}_i \qquad \text{with} \qquad \boldsymbol{r}_{pi} = \boldsymbol{p}_{ci} - \boldsymbol{p}_{pi}$$

- Constant vector both in the case of prismatic and revolute joints

$${}^i\boldsymbol{r}_{pi} = \begin{pmatrix} r_{pi,x} & r_{pi,y} & r_{pi,z} \end{pmatrix}^T = \boldsymbol{R}_i^T \boldsymbol{r}_{pi}$$

- The kinetic energy of link i can be rewritten using the pivot point as

$$\mathcal{T}_i = \frac{1}{2}\, m_i\, \boldsymbol{v}_{pi}^T \boldsymbol{v}_{pi} + \boldsymbol{\omega}_i^T \boldsymbol{R}_i\, \boldsymbol{S}(m_i\, {}^i\boldsymbol{r}_{pi})\, \boldsymbol{R}_i^T \boldsymbol{v}_{pi} + \frac{1}{2}\, \boldsymbol{\omega}_i^T \boldsymbol{R}_i\, {}^i\bar{\boldsymbol{I}}_i\, \boldsymbol{R}_i^T \boldsymbol{\omega}_i$$

- Constant quantities in $\mathcal{T}_i$
  - the link mass m_i, a scalar known as the zero-th moment of inertia
  - the first moment of inertia
$$m_i {}^i\boldsymbol{r}_{pi} = \begin{pmatrix} m_i\, r_{pi,x} \\ m_i\, r_{pi,y} \\ m_i\, r_{pi,z} \end{pmatrix}$$
  - the inertia tensor ${}^i\bar{\boldsymbol{I}}_i$ referred to $\boldsymbol{p}_{pi}$
- According to Steiner theorem, the second moment of inertia can be expressed as
$${}^i\bar{\boldsymbol{I}}_i = {}^i\boldsymbol{I}_i + m_i\, \boldsymbol{S}^T({}^i\boldsymbol{r}_{pi})\boldsymbol{S}({}^i\boldsymbol{r}_{pi})$$
$$= \begin{pmatrix} I_{ixx} + m_i(r_{di,y}^2 + r_{di,z}^2) & I_{ixy} - m_i r_{di,x} r_{di,y} & I_{ixz} - m_i r_{di,x} r_{di,z} \\ * & I_{iyy} + m_i(r_{di,x}^2 + r_{di,z}^2) & I_{iyz} - m_i r_{di,y} r_{di,z} \\ * & * & I_{izz} + m_i(r_{di,x}^2 + r_{di,y}^2) \end{pmatrix}$$
$$= \begin{pmatrix} \bar{I}_{ixx} & \bar{I}_{ixy} & \bar{I}_{ixz} \\ * & \bar{I}_{iyy} & \bar{I}_{iyz} \\ * & * & \bar{I}_{izz} \end{pmatrix}$$

- Link kinetic energy $\mathcal{T}_i$ is linear in $n_\ell = 10$ standard dynamic parameters
- Link potential energy $\mathcal{U}_i$ is also linear in a subset of these parameters

$$\mathcal{U}_i = -\boldsymbol{g}_0^T \left(m_i \, \boldsymbol{p}_{pi} + \boldsymbol{R}_i \, m_i{}^i \boldsymbol{r}_{pi} \right)$$

- Contributions by motors to kinetic and potential energy keep the same linearity
- Lagrangian in linearly parametrized form

$$\mathcal{L} = \mathcal{T} - \mathcal{U} = \sum_{i=1}^{n} \left(\boldsymbol{\beta}_{\mathcal{T}_i}^T - \boldsymbol{\beta}_{\mathcal{U}_i}^T \right) \boldsymbol{\pi}_i$$

  where $\boldsymbol{\beta}_{\mathcal{T}_i}$ and $\boldsymbol{\beta}_{\mathcal{U}_i}$ are suitable vector functions of $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$
- Vector of $n_\ell + 1 = 11$ dynamic parameters of link and motor i

$$\boldsymbol{\pi}_i = \begin{pmatrix} m_i & m_i r_{di,x} & m_i r_{di,y} & m_i r_{di,z} & \bar{I}_{ixx} & \bar{I}_{ixy} & \bar{I}_{ixz} & \bar{I}_{iyy} & \bar{I}_{iyz} & \bar{I}_{izz} & \bar{I}_{mi} \end{pmatrix}^T$$

  with $\bar{I}_{mi} = I_{mi} n_{ri}^2$

- The i-th dynamic equation can be written as

$$\sum_{j=1}^{n} \boldsymbol{y}_{ij}^T \boldsymbol{\pi}_j = \xi_i$$

$$\boldsymbol{y}_{ij} = \frac{d}{dt}\frac{\partial \boldsymbol{\beta}_{\mathcal{T}j}}{\partial \dot{q}_i} - \frac{\partial \boldsymbol{\beta}_{\mathcal{T}j}}{\partial q_i} + \frac{\partial \boldsymbol{\beta}_{\mathcal{U}j}}{\partial q_i}$$

- Since derivatives vanish for $j < i$, the following upper block-triangular structure is obtained

$$\begin{pmatrix} \boldsymbol{y}_{11}^T & \boldsymbol{y}_{12}^T & \cdots & \boldsymbol{y}_{1n}^T \\ \mathbf{0}^T & \boldsymbol{y}_{22}^T & \cdots & \boldsymbol{y}_{2n}^T \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}^T & \mathbf{0}^T & \cdots & \boldsymbol{y}_{nn}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \\ \vdots \\ \boldsymbol{\pi}_n \end{pmatrix} = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix}$$

- When dissipative terms are included, check linearity in terms of additional parameters
- Summarizing, when link, motor, and dissipative coefficients are all included, the robot dynamic model can be rewritten as

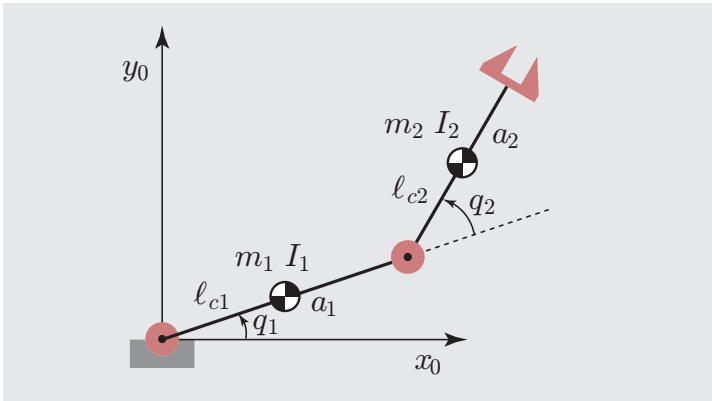
$$\boldsymbol{Y}_{\boldsymbol{\pi}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\,\boldsymbol{\pi} = \boldsymbol{\tau}$$

- The dynamic model is linear with respect to a suitable set of parameters
- $\boldsymbol{\pi} \in \mathbb{R}^p$ collects all robot dynamic parameters
- $\boldsymbol{Y}_{\boldsymbol{\pi}}$ is an $n \times p$ matrix called the regressor of the robot dynamic model
- Some columns of the regressor matrix may be zero

Dynamic Model of Simple Manipulator Structures

- Three cases of derivation of the dynamic model via the EL formalism for simple manipulator structures having two DoFs
  - highlight the physical meaning of all dynamic terms, including inertial coupling
  - in last case, the example introduces also a general recursive computation of the kinetic energy
- For robots with more DoFs, it is advisable to derive the EL model with the aid of symbolic manipulation software

- 2R planar arm



- Derivation using partial Jacobians for the expression of kinetic energy

$$\boldsymbol{J}_p^1 = \begin{pmatrix} -\ell_{c1}s_1 & 0 \\ \ell_{c1}c_1 & 0 \\ 0 & 0 \end{pmatrix} \qquad \boldsymbol{J}_p^2 = \begin{pmatrix} -a_1 s_1 - \ell_{c2}s_{12} & -\ell_{c2}s_{12} \\ a_1 c_1 + \ell_{c2}c_{12} & \ell_{c2}c_{12} \\ 0 & 0 \end{pmatrix}$$

$$\boldsymbol{J}_o^1 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix} \qquad \boldsymbol{J}_o^2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix}$$

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{pmatrix} m_{11}(q_2) & m_{12}(q_2) \\ m_{21}(q_2) & m_{22} \end{pmatrix}$$

$$m_{11} = I_1 + m_1\ell_{c1}^2 + I_{m1}n_{r1}^2 + I_2 + m_2(a_1^2 + \ell_{c2}^2) + 2m_2\ell_{c2}a_1c_2$$
$$m_{12} = m_{21} = I_2 + m_2\ell_{c2}^2 + m_2\ell_{c2}a_1c_2$$
$$m_{22} = I_2 + m_2\ell_{c2}^2 + I_{m2}n_{r2}^2$$

- Christoffel coefficients

$$
\begin{aligned}
c_{111} &= \frac{1}{2}\frac{\partial m_{11}}{\partial q_1} = 0 \\
c_{112} = c_{121} &= \frac{1}{2}\frac{\partial m_{11}}{\partial q_2} = -m_2 \ell_{c2} a_1 s_2 \\
c_{122} &= \frac{\partial m_{12}}{\partial q_2} - \frac{1}{2}\frac{\partial m_{22}}{\partial q_1} = -m_2 \ell_{c2} a_1 s_2 \\
c_{211} &= \frac{\partial m_{21}}{\partial q_1} - \frac{1}{2}\frac{\partial m_{11}}{\partial q_2} = m_2 \ell_{c2} a_1 s_2 \\
c_{212} = c_{221} &= \frac{1}{2}\frac{\partial m_{22}}{\partial q_1} = 0 \\
c_{222} &= \frac{1}{2}\frac{\partial m_{22}}{\partial q_2} = 0
\end{aligned}
$$

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = -m_2\ell_{c2}a_1 s_2 \begin{pmatrix} \dot{q}_2 & \dot{q}_1+\dot{q}_2 \\ -\dot{q}_1 & 0 \end{pmatrix}$$

$$\Downarrow$$

$$\begin{aligned}
\boldsymbol{S}(\boldsymbol{q},\dot{\boldsymbol{q}}) &= \dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\,\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \\
&= -m_2\ell_{c2}a_1 s_2 \begin{pmatrix} 2\dot{q}_2 & \dot{q}_2 \\ \dot{q}_2 & 0 \end{pmatrix} + 2m_2\ell_{c2}a_1 s_2 \begin{pmatrix} \dot{q}_2 & \dot{q}_1+\dot{q}_2 \\ -\dot{q}_1 & 0 \end{pmatrix} \\
&= -m_2\ell_{c2}a_1 s_2 \begin{pmatrix} 0 & -(2\dot{q}_1+\dot{q}_2) \\ 2\dot{q}_1+\dot{q}_2 & 0 \end{pmatrix}
\end{aligned}$$

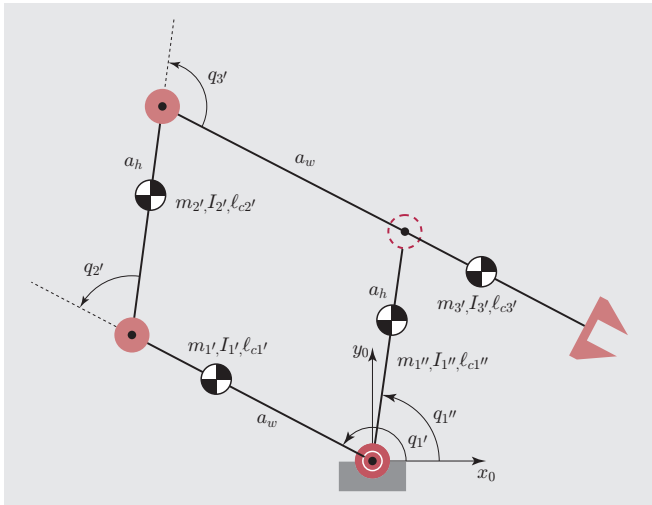
$$\begin{aligned}
g_1 &= (m_1\ell_{c1} + m_2 a_1)gc_1 + m_2\ell_{c2}gc_{12} \\
g_2 &= m_2\ell_{c2}gc_{12}
\end{aligned}$$

- Resulting equations of motion

$$\begin{aligned}
&\left(I_1 + m_1\ell_{c1}^2 + I_{m1}n_{r1}^2 + I_2 + m_2(a_1^2 + \ell_{c2}^2) + 2m_2\ell_{c2}a_1c_2\right)\ddot{q}_1 \\
&+ \left(I_2 + m_2\ell_{c2}^2 + m_2\ell_{c2}a_1c_2\right)\ddot{q}_2 - 2m_2\ell_{c2}a_1s_2\dot{q}_1\dot{q}_2 - m_2\ell_{c2}a_1s_2\dot{q}_2^2 \\
&+ (m_1\ell_{c1} + m_2a_1)gc_1 + m_2\ell_{c2}gc_{12} = \tau_1
\end{aligned}$$

$$\begin{aligned}
&\left(I_2 + m_2\ell_{c2}^2 + m_2\ell_{c2}a_1c_2\right)\ddot{q}_1 + \left(I_2 + m_2\ell_{c2}^2 + I_{m2}n_{r2}^2\right)\ddot{q}_2 \\
&+ m_2\ell_{c2}a_1s_2\dot{q}_1^2 + m_2\ell_{c2}gc_{12} = \tau_2
\end{aligned}$$

- Parallelogram arm



- Derivation using partial Jacobians for the expression of kinetic energy

$$\boldsymbol{J}_p^{1'} = \begin{pmatrix} -\ell_{c1'} s_{1'} & 0 & 0 \\ \ell_{c1'} c_{1'} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \boldsymbol{J}_p^{2'} = \begin{pmatrix} -a_w s_{1'} - \ell_{c2'} s_{1'2'} & -\ell_{c2'} s_{1'2'} & 0 \\ a_w c_{1'} + \ell_{c2'} c_{1'2'} & \ell_{c2'} c_{1'2'} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{J}_p^{3'} = \begin{pmatrix} -a_w s_{1'} - a_h s_{1'2'} - \ell_{c3'} s_{1'2'3'} & -a_h s_{1'2'} - \ell_{c3'} s_{1'2'3'} & -\ell_{c3'} s_{1'2'3'} \\ a_w c_{1'} + a_h c_{1'2'} + \ell_{c3'} c_{1'2'3'} & a_h c_{1'2'} + \ell_{c3'} c_{1'2'3'} & \ell_{c3'} c_{1'2'3'} \\ 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{J}_p^{1''} = \begin{pmatrix} -\ell_{c1''} s_{1''} \\ \ell_{c1''} c_{1''} \\ 0 \end{pmatrix}$$

$$\boldsymbol{J}_o^{1'} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad \boldsymbol{J}_o^{2'} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \quad \boldsymbol{J}_o^{3'} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad \boldsymbol{J}_o^{1''} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$
\begin{aligned}
m_{1'1'} &= I_{1'} + m_{1'}\ell_{c1'}^2 + I_{m1'}n_{r1'}^2 + I_{2'} + m_{2'}(a_w^2 + \ell_{c2'}^2 + 2a_w\ell_{c2'}c_{2'}) + I_{3'} \\
&\quad + m_{3'}(a_w^2 + a_h^2 + \ell_{c3'}^2 + 2a_w a_h c_{2'} + 2a_w\ell_{c3'}c_{2'3'} + 2a_h\ell_{c3'}c_{3'}) \\
m_{1'2'} &= m_{2'1'} = I_{2'} + m_{2'}(\ell_{c2'}^2 + a_w\ell_{c2'}c_{2'}) + I_{3'} \\
&\quad + m_{3'}(a_h^2 + \ell_{c3'}^2 + a_w a_h c_{2'} + a_w\ell_{c3'}c_{2'3'} + 2a_h\ell_{c3'}c_{3'}) \\
m_{1'3'} &= m_{3'1'} = I_{3'} + m_{3'}(\ell_{c3'}^2 + a_w\ell_{c3'}c_{2'3'} + a_h\ell_{c3'}c_{3'}) \\
m_{2'2'} &= I_{2'} + m_{2'}\ell_{c2'}^2 + I_{3'} + m_{3'}(a_h^2 + \ell_{c3'}^2 + 2a_h\ell_{c3'}c_{3'}) \\
m_{2'3'} &= m_{3'2'} = I_{3'} + m_{3'}(\ell_{c3'}^2 + a_h\ell_{c3'}c_{3'}) \\
m_{3'3'} &= I_{3'} + m_{3'}\ell_{c3'}^2 \\
\\
m_{1''1''} &= I_{1''} + m_{1''}\ell_{c1''}^2 + I_{m1''}n_{r1''}^2
\end{aligned}
$$

$$\boldsymbol{\Upsilon} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{\Upsilon}^T \boldsymbol{M}_o(\boldsymbol{q}) \boldsymbol{\Upsilon} + \boldsymbol{M}_m$$

$$\begin{aligned} m_{11} &= I_{1'} + m_{1'} \ell_{c1'}^2 + m_{2'} a_w^2 + I_{3'} + m_{3'} \bar{\ell}_{c3'}^2 + I_{m1'} n_{r1'}^2 \\ m_{12} &= m_{21} = \left(a_w m_{2'} \ell_{c2'} - a_h m_{3'} \bar{\ell}_{c3'} \right) \cos\left(q_{1''} - q_{1'} \right) \\ m_{22} &= I_{1'} + m_{1'} \ell_{c1'}^2 + I_{2'} + m_{2'} \ell_{c2'}^2 + m_{3'} a_h^2 + I_{m1''} n_{r1''}^2 \end{aligned}$$

- A configuration-independent decoupled inertia is obtained when

$$a_w m_{2'} \ell_{c2'} = a_h m_{3'} \bar{\ell}_{c3'}$$

$$
\begin{aligned}
g_{1'} &= (m_{1'}\ell_{c1'} + m_{2'}a_w + m_{3'}a_w)gc_{1'} + (m_{2'}\ell_{c2'} + m_{3'}a_h)gc_{1'2'} \\
&\quad + m_{3'}\ell_{c3'}gc_{1'2'3} \\
g_{2'} &= (m_{2'}\ell_{c2'} + m_{3'}a_h)gc_{1'2'} + m_{3'}\ell_{c3'}gc_{1'2'3} \\
g_{3'} &= m_{3'}\ell_{c3'}gc_{1'2'3}
\end{aligned}
$$

$$
g_{1''} = m_{1''}\ell_{c1''}gc_{1''}
$$

$$
\boldsymbol{g} = \boldsymbol{\Upsilon}^T \boldsymbol{g}_o = \begin{pmatrix} (m_{1'}\ell_{c1'} + m_{2'}a_w - m_{3'}\bar{\ell}_{c3'})gc_{1'} \\ (m_{1''}\ell_{c1''} + m_{2'}\ell_{c2'} + m_{3'}a_h)gc_{1''} \end{pmatrix}
$$

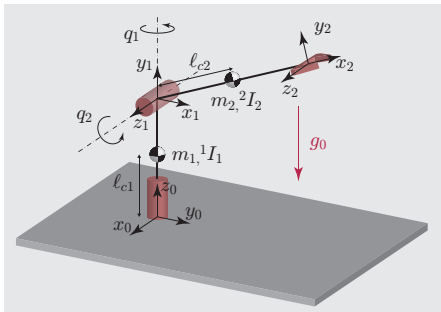
2R Polar Arm

- Kinetic energy

$$\mathcal{T}_i = \frac{1}{2}\, m_i\, {}^i\boldsymbol{v}_{ci}^T\, {}^i\boldsymbol{v}_{ci} + \frac{1}{2}\, {}^i\boldsymbol{\omega}_i^T\, {}^i\boldsymbol{I}_i\, {}^i\boldsymbol{\omega}_i$$

- Joint type

$$\sigma_i = \begin{cases} 1 & \text{if joint } i \text{ is prismatic} \\ 0 & \text{if joint } i \text{ is revolute} \end{cases}$$



Recursive Symbolic Computation of Velocities in Moving Frames

- Derivation using the recursion on velocities expressed in moving frames

$$
\begin{aligned}
{}^{i}\boldsymbol{\omega}_i &= {}^{i-1}\boldsymbol{R}_i^T(q_i)\left({}^{i-1}\boldsymbol{\omega}_{i-1} + (1-\sigma_i)\,\dot{q}_i\,{}^{i-1}\boldsymbol{z}_{i-1}\right)\\
{}^{i}\boldsymbol{v}_i &= {}^{i-1}\boldsymbol{R}_i^T(q_i)\left({}^{i-1}\boldsymbol{v}_{i-1} + {}^{i-1}\boldsymbol{\omega}_i \times {}^{i-1}\boldsymbol{r}_{i-1,i} + \sigma_i\dot{q}_i\,{}^{i-1}\boldsymbol{z}_{i-1}\right)
\end{aligned}
$$

$$
\Downarrow
$$

$$
{}^{i}\boldsymbol{v}_{ci} = {}^{i}\boldsymbol{v}_i + {}^{i}\boldsymbol{\omega}_i \times {}^{i}\boldsymbol{r}_{ci}
$$

- DH parameters

Link	α_i	a_i	d_i	θ_i
1	$\pi/2$	0	d_1	q_1
2	0	a_2	0	q_2

$$
{}^0\boldsymbol{R}_1(q_1) = \begin{pmatrix} c_1 & 0 & s_1 \\ s_1 & 0 & -c_1 \\ 0 & 1 & 0 \end{pmatrix} \qquad {}^1\boldsymbol{R}_2(q_2) = \begin{pmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{pmatrix}
$$

$$
{}^1\boldsymbol{r}_{0,1} = \begin{pmatrix} 0 \\ d_1 \\ 0 \end{pmatrix} \qquad {}^2\boldsymbol{r}_{1,2} = \begin{pmatrix} a_2 \\ 0 \\ 0 \end{pmatrix}
$$

$$
{}^1\boldsymbol{r}_{c1} = \begin{pmatrix} 0 \\ -d_1 + \ell_{c1} \\ 0 \end{pmatrix} \qquad {}^2\boldsymbol{r}_{c2} = \begin{pmatrix} -a_2 + \ell_{c2} \\ 0 \\ 0 \end{pmatrix} \qquad {}^2\boldsymbol{I}_2 = \begin{pmatrix} I_{2xx} & 0 & 0 \\ 0 & I_{2yy} & 0 \\ 0 & 0 & I_{2zz} \end{pmatrix}
$$

$$
{}^1\boldsymbol{\omega}_1 = \begin{pmatrix} 0 \\ \dot{q}_1 \\ 0 \end{pmatrix} \qquad {}^1\boldsymbol{v}_1 = \mathbf{0} \qquad {}^1\boldsymbol{v}_{c1} = \mathbf{0}
$$

$$
{}^2\boldsymbol{\omega}_2 = \begin{pmatrix} s_2\,\dot{q}_1 \\ c_2\,\dot{q}_1 \\ \dot{q}_2 \end{pmatrix} \qquad {}^2\boldsymbol{v}_2 = \begin{pmatrix} 0 \\ a_2\,\dot{q}_2 \\ -a_2 c_2\,\dot{q}_1 \end{pmatrix} \qquad {}^2\boldsymbol{v}_{c2} = \begin{pmatrix} 0 \\ \ell_{c2}\,\dot{q}_2 \\ -\ell_{c2} c_2\,\dot{q}_1 \end{pmatrix}
$$

$$
\mathcal{T}_1 = \frac{1}{2}\,{}^1\boldsymbol{\omega}_1^T\,{}^i\boldsymbol{I}_i\,{}^1\boldsymbol{\omega}_1 = \frac{1}{2}\,I_{1,yy}\,\dot{q}_1^2
$$

$$
\mathcal{T}_2 = \frac{1}{2}\,m_2\ell_{c2}^2\left(c_2^2\,\dot{q}_1^2 + \dot{q}_2^2\right) + \frac{1}{2}\left(I_{2,xx}\,s_2^2 + I_{2,yy}\,c_2^2\right)\dot{q}_1^2 + \frac{1}{2}\,I_{2,zz}\,\dot{q}_2^2
$$

$$
\mathcal{T} = \mathcal{T}_1 + \mathcal{T}_2 = \frac{1}{2}\,\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} = \frac{1}{2}\,\dot{\boldsymbol{q}}^T\begin{pmatrix} \rho_1 + \rho_2\,c_2^2 & 0 \\ 0 & \rho_3 \end{pmatrix}\dot{\boldsymbol{q}}
$$

$$
\rho_1 = I_{1,yy} + I_{2,xx} \qquad \rho_2 = I_{2,yy} + m_2\ell_{c2}^2 - I_{2,xx} \qquad \rho_3 = I_{2,zz} + m_2\ell_{c2}^2
$$

$$\boldsymbol{C}_1(\boldsymbol{q}) = \begin{pmatrix} 0 & -\rho_2\, s_2 c_2 \\ -\rho_2\, s_2 c_2 & 0 \end{pmatrix}$$

$$\boldsymbol{C}_2(\boldsymbol{q}) = \begin{pmatrix} \rho_2\, s_2 c_2 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\Downarrow$$

$$\begin{aligned} \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \begin{pmatrix} -2\rho_2\, s_2 c_2\, \dot{q}_1 \dot{q}_2 \\ \rho_2\, s_2 c_2\, \dot{q}_1^2 \end{pmatrix} \\ &= \begin{pmatrix} -\rho_2\, s_2 c_2\, \dot{q}_2 & -\rho_2\, s_2 c_2\, \dot{q}_1 \\ \rho_2\, s_2 c_2\, \dot{q}_1 & 0 \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \end{aligned}$$

- $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is skew-symmetric

Potential Energy and Gravity Terms

$$\mathcal{U}_1 = \text{constant}$$

$$\mathcal{U}_2 = g m_2 \ell_{c2}\, s_2 = \rho_4 g\, s_2$$

$$\mathcal{U} = \mathcal{U}_1 + \mathcal{U}_2$$

$$\boldsymbol{g}(\boldsymbol{q}) = \left(\frac{\partial \mathcal{U}}{\partial \boldsymbol{q}}\right)^T = \begin{pmatrix} 0 \\ \rho_4 g\, c_2 \end{pmatrix}$$

- Summing up yields

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

Identification of Dynamic Parameters

- The use of the dynamic model for simulation and control purposes requires the knowledge of the numerical values of the dynamic parameters of the robot
- Obtaining such parameters from the design data of the mechanical structure may not be so easy for an end-user
- CAD modeling techniques allow the computation of inertial parameters from geometry and materials, but these estimates are typically inaccurate
- Other dynamic effects, such as joint friction, are not included in the CAD model

- Accurate estimates of the robot dynamic parameters can be obtained using identification techniques that exploit the linearity property
- These techniques compute the parameter vector $\boldsymbol{\pi}$ from the knowledge of the applied joint torques $\boldsymbol{\tau}$ and the regressor matrix $\boldsymbol{Y}_{\boldsymbol{\pi}}$ along suitably chosen trajectories
- Commanded torques may differ from the torques actually realized by the servomotors, and thus it is preferable to use measurements
- When joint torque sensors are available, they directly measure the torques transferred to the robot links
- More often, torque is obtained from current measurements, provided the current-to-torque conversion factor is known
- With the kinematic parameters being known, only joint positions $\boldsymbol{q}$, velocities $\dot{\boldsymbol{q}}$ and accelerations $\ddot{\boldsymbol{q}}$ are required
  - joint positions are measured with good accuracy by encoders
  - velocities and accelerations are reconstructed numerically from recorded position data
  - offline processing allows the use of anti-causal filters incorporating future data samples

- The $n \times p$ regressor matrix $\boldsymbol{Y}_{\boldsymbol{\pi}}$ has several zero columns, corresponding to parameters that do not contribute to the equations of motion
  - removing them leaves $p_r < p$ columns
- If the p_r columns are structurally dependent, only some r linear combinations with fixed coefficients $\boldsymbol{\delta}_j \in I\!R^r$, $j = 1, \ldots, p_r - r$, can be identified

$$\boldsymbol{Y}_{\mathrm{dep}} = \boldsymbol{Y}_{\mathrm{ind}} \begin{pmatrix} \boldsymbol{\delta}_1 & \ldots & \boldsymbol{\delta}_{p_r - r} \end{pmatrix} = \boldsymbol{Y}_{\mathrm{ind}} \boldsymbol{\Delta}$$

$$\begin{aligned} \boldsymbol{Y}_{\boldsymbol{\pi}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \boldsymbol{\pi} &= \begin{pmatrix} \boldsymbol{Y}_{\mathrm{ind}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) & \boldsymbol{Y}_{\mathrm{dep}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \end{pmatrix} \begin{pmatrix} \boldsymbol{\pi}_{\mathrm{ind}} \\ \boldsymbol{\pi}_{\mathrm{dep}} \end{pmatrix} \\ &= \boldsymbol{Y}_{\mathrm{ind}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \left(\boldsymbol{\pi}_{\mathrm{ind}} + \boldsymbol{\Delta} \boldsymbol{\pi}_{\mathrm{dep}} \right) = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \, \boldsymbol{\rho}, \end{aligned}$$

- $\boldsymbol{\rho} = \boldsymbol{\pi}_{\mathrm{ind}} + \boldsymbol{\Delta} \boldsymbol{\pi}_{\mathrm{dep}}$ is the r-dimensional vector of dynamic coefficients

Example

- Reduced parameters for the 2R planar arm

$$\boldsymbol{\pi} = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ \pi_6 \\ \pi_7 \end{pmatrix} = \begin{pmatrix} m_1 \ell_{c1} \\ I_1 + m_1 \ell_{c1}^2 \\ I_{m1} n_{r1}^2 \\ m_2 \\ I_2 + m_2 \ell_{c2}^2 \\ m_2 \ell_{c2} \\ I_{m2} n_{r2}^2 \end{pmatrix}$$

- Dynamic model linear in $\boldsymbol{\pi}$

$$\begin{aligned} &(\pi_2 + \pi_3 + a_1^2 \pi_4 + \pi_5 + 2a_1 c_2 \pi_6)\, \ddot{q}_1 + (\pi_5 + a_1 c_2 \pi_6)\, \ddot{q}_2 \\ &-2a_1 s_2 \pi_6 \dot{q}_1 \dot{q}_2 - a_1 s_2 \pi_6 \dot{q}_2^2 + (\pi_1 + a_1 \pi_4)\, g c_1 + \pi_6 g c_{12} \;=\; \tau_1 \\ &(\pi_5 + a_1 c_2 \pi_6)\, \ddot{q}_1 + (\pi_5 + \pi_7)\, \ddot{q}_2 + a_1 s_2 \pi_6 \dot{q}_1^2 + \pi_6 g c_{12} \;=\; \tau_2 \end{aligned}$$

$$\boldsymbol{Y_\pi} = \begin{pmatrix} \boldsymbol{y_{\pi 1}} & \boldsymbol{y_{\pi 2}} & \boldsymbol{y_{\pi 3}} & \boldsymbol{y_{\pi 4}} & \boldsymbol{y_{\pi 5}} & \boldsymbol{y_{\pi 6}} & \boldsymbol{y_{\pi 7}} \end{pmatrix}$$

$$\boldsymbol{y_{\pi 1}} = \begin{pmatrix} gc_1 \\ 0 \end{pmatrix} \qquad \boldsymbol{y_{\pi 2}} = \begin{pmatrix} \ddot{q}_1 \\ 0 \end{pmatrix} \qquad \boldsymbol{y_{\pi 3}} = \begin{pmatrix} \ddot{q}_1 \\ 0 \end{pmatrix}$$

$$\boldsymbol{y_{\pi 4}} = \begin{pmatrix} a_1^2\ddot{q}_1 + ga_1c_1 \\ 0 \end{pmatrix} \qquad \boldsymbol{y_{\pi 5}} = \begin{pmatrix} \ddot{q}_1 + \ddot{q}_2 \\ \ddot{q}_1 + \ddot{q}_2 \end{pmatrix}$$

$$\boldsymbol{y_{\pi 6}} = \begin{pmatrix} 2a_1c_2\ddot{q}_1 + a_1c_2\ddot{q}_2 - 2a_1s_2\dot{q}_1\dot{q}_2 - a_1s_2\dot{q}_2^2 + gc_{12} \\ a_1c_2\ddot{q}_1 + a_1s_2\dot{q}_1^2 + gc_{12} \end{pmatrix} \qquad \boldsymbol{y_{\pi 7}} = \begin{pmatrix} 0 \\ \ddot{q}_2 \end{pmatrix}$$

$$\boldsymbol{y}_{\boldsymbol{\pi}3} = \boldsymbol{y}_{\boldsymbol{\pi}2}$$
$$\boldsymbol{y}_{\boldsymbol{\pi}4} = a_1 \boldsymbol{y}_{\boldsymbol{\pi}1} + a_1^2 \boldsymbol{y}_{\boldsymbol{\pi}2}$$

$$\boldsymbol{Y}_{\text{ind}} = \begin{pmatrix} \boldsymbol{y}_{\boldsymbol{\pi}1} & \boldsymbol{y}_{\boldsymbol{\pi}2} & \boldsymbol{y}_{\boldsymbol{\pi}5} & \boldsymbol{y}_{\boldsymbol{\pi}6} & \boldsymbol{y}_{\boldsymbol{\pi}7} \end{pmatrix} \qquad \boldsymbol{Y}_{\text{dep}} = \begin{pmatrix} \boldsymbol{y}_{\boldsymbol{\pi}3} & \boldsymbol{y}_{\boldsymbol{\pi}4} \end{pmatrix} = \boldsymbol{Y}_{\text{ind}} \boldsymbol{\Delta}$$

$$\boldsymbol{\Delta} = \begin{pmatrix} 0 & a_1 \\ 1 & a_1^2 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\boldsymbol{\pi}_{\text{ind}} = \begin{pmatrix} \pi_1 & \pi_2 & \pi_5 & \pi_6 & \pi_7 \end{pmatrix}^T \qquad \boldsymbol{\pi}_{\text{dep}} = \begin{pmatrix} \pi_3 & \pi_4 \end{pmatrix}^T$$

$$\boldsymbol{\rho} = \boldsymbol{\pi}_{\text{ind}} + \boldsymbol{\Delta\pi}_{\text{dep}} = \begin{pmatrix} m_1\ell_{c1} + m_2 a_1 \\ I_1 + m_1\ell_{c1}^2 + I_{m1} n_{r1}^2 + m_2 a_1^2 \\ I_2 + m_2\ell_{c2}^2 \\ m_2\ell_{c2} \\ I_{m2} n_{r2}^2 \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_5 \end{pmatrix}$$

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

- New regressor matrix $\boldsymbol{Y}$ with structural independent columns

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 \end{pmatrix}$$

$$\boldsymbol{y}_1 = \begin{pmatrix} gc_1 \\ 0 \end{pmatrix} \qquad \boldsymbol{y}_2 = \begin{pmatrix} \ddot{q}_1 \\ 0 \end{pmatrix} \qquad \boldsymbol{y}_3 = \begin{pmatrix} \ddot{q}_1 + \ddot{q}_2 \\ \ddot{q}_1 + \ddot{q}_2 \end{pmatrix}$$

$$\boldsymbol{y}_4 = \begin{pmatrix} 2a_1c_2\ddot{q}_1 + a_1c_2\ddot{q}_2 - 2a_1s_2\dot{q}_1\dot{q}_2 - a_1s_2\dot{q}_2^2 + gc_{12} \\ a_1c_2\ddot{q}_1 + a_1s_2\dot{q}_1^2 + gc_{12} \end{pmatrix} \qquad \boldsymbol{y}_5 = \begin{pmatrix} 0 \\ \ddot{q}_2 \end{pmatrix}$$

- Joint data are collected at N time instants $t_1, \ldots, t_N$ along a given motion trajectory
- The equations are stacked as

$$\bar{\boldsymbol{\tau}} = \begin{pmatrix} \boldsymbol{\tau}(t_1) \\ \vdots \\ \boldsymbol{\tau}(t_N) \end{pmatrix} = \begin{pmatrix} \boldsymbol{Y}(t_1) \\ \vdots \\ \boldsymbol{Y}(t_N) \end{pmatrix} \boldsymbol{\rho} = \bar{\boldsymbol{Y}} \boldsymbol{\rho}$$

- With $\bar{\boldsymbol{Y}}$ full column rank

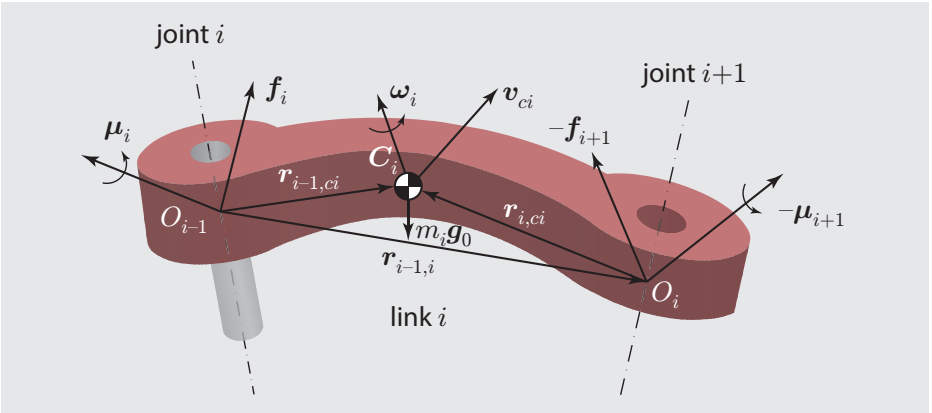
$$\hat{\boldsymbol{\rho}} = \bar{\boldsymbol{Y}}^{\dagger} \bar{\boldsymbol{\tau}} = (\bar{\boldsymbol{Y}}^T \bar{\boldsymbol{Y}})^{-1} \bar{\boldsymbol{Y}}^T \bar{\boldsymbol{\tau}}$$

- Choice of motions for identification
  - number of data should be large enough so to obtain a matrix $\bar{\boldsymbol{Y}}$ with full column rank
  - use of sufficiently rich periodic trajectories is advisable
  - trajectories should contain multiple harmonics at different frequencies and adequately cover the configuration space
  - a large range of velocities and accelerations is needed to obtain a suitable condition number κ for $\bar{\boldsymbol{Y}}$
  - trajectories should not excite unmodeled dynamic effects, such as joint elasticity
  - in the presence of an unknown payload at the end-effector, one may lump its mass and inertia into the standard parameter vector of the last link

Newton–Euler Formulation

- The EL formulation derives the equations of motion in a symbolic form suitable for studying model properties and for control design
- The NE formulation computes the dynamic model in numeric form, efficiently exploiting the serial nature of the kinematic chain
- The NE formulation proceeds from balance of forces and moments acting on each link
- The resulting NE equations lend themselves to a recursive computation
  - forward recursion propagates link velocities and accelerations from base to end-effector
  - backward recursion propagates forces and moments from end-effector to base
- From these forces and moments, joint torques can be computed

- Forces and moments acting on link i



Forward Recursion of Velocities and Accelerations

- In the following, forward recursive formulas are derived by exploiting the serial nature of the kinematic chain from link $i-1$ to link i
  - link velocities
  - link accelerations
- The corresponding formulas are mainly intended for numeric evaluation

Link Velocities

- Composition of vectors in a serial chain

$$\boldsymbol{r}_{0,i} = \boldsymbol{r}_{0,i-1} + \boldsymbol{r}_{i-1,i} = \boldsymbol{r}_{0,i-1} + \boldsymbol{R}_{i-1}{}^{i-1}\boldsymbol{r}_{i-1,i}$$

$$\boldsymbol{v}_i = \boldsymbol{v}_{i-1} + \boldsymbol{\omega}_{i-1} \times \boldsymbol{r}_{i-1,i} + \boldsymbol{v}_{i-1,i}$$

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + \boldsymbol{\omega}_{i-1,i}$$

- Relative velocities between successive links
  - if joint i is prismatic

$$\boldsymbol{\omega}_{i-1,i} = \mathbf{0} \qquad \boldsymbol{v}_{i-1,i} = \dot{q}_i\, \boldsymbol{z}_{i-1}$$

  - if joint i is revolute

$$\boldsymbol{\omega}_{i-1,i} = \dot{q}_i\, \boldsymbol{z}_{i-1} \qquad \boldsymbol{v}_{i-1,i} = \boldsymbol{\omega}_{i-1,i} \times \boldsymbol{r}_{i-1,i} = \dot{q}_i\, \boldsymbol{z}_{i-1} \times \boldsymbol{r}_{i-1,i}$$

- Forward recursion on velocities (with $\sigma_i = 1$ for prismatic, $\sigma_i = 0$ for revolute joints)

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_{i-1} + (1 - \sigma_i)\, \dot{q}_i \boldsymbol{z}_{i-1}$$

$$\boldsymbol{v}_i = \boldsymbol{v}_{i-1} + \boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i} + \sigma_i \dot{q}_i \boldsymbol{z}_{i-1}$$

Link Accelerations

- Forward recursion on angular accelerations

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1} + (1-\sigma_i)\left(\ddot{q}_i \boldsymbol{z}_{i-1} + \dot{q}_i \dot{\boldsymbol{z}}_{i-1}\right)$$

$$\dot{\boldsymbol{z}}_{i-1} = \boldsymbol{\omega}_{i-1} \times \boldsymbol{z}_{i-1}$$

$$\dot{\boldsymbol{\omega}}_i = \dot{\boldsymbol{\omega}}_{i-1} + (1-\sigma_i)\left(\ddot{q}_i \boldsymbol{z}_{i-1} + \boldsymbol{\omega}_{i-1} \times \dot{q}_i \boldsymbol{z}_{i-1}\right)$$

- Forward recursion on linear accelerations

$$\dot{\boldsymbol{v}}_i = \dot{\boldsymbol{v}}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} + \boldsymbol{\omega}_i \times \dot{\boldsymbol{r}}_{i-1,i} + \sigma_i(\ddot{q}_i \boldsymbol{z}_{i-1} + \dot{q}_i \dot{\boldsymbol{z}}_{i-1})$$

$$\dot{\boldsymbol{r}}_{i-1,i} = \boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i} + \sigma_i \dot{q}_i \boldsymbol{z}_{i-1}$$

$$\dot{\boldsymbol{v}}_i = \dot{\boldsymbol{v}}_{i-1} + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i-1,i} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i-1,i}) + \sigma_i(\ddot{q}_i \boldsymbol{z}_{i-1} + 2\boldsymbol{\omega}_i \times \dot{q}_i \boldsymbol{z}_{i-1})$$

Backward Recursion of Forces and Moments

- Newton and Euler equations

$$\boldsymbol{f}_i - \boldsymbol{f}_{i+1} + m_i \boldsymbol{g}_0 = \frac{d}{dt}\left(m_i \boldsymbol{v}_{ci}\right) = m_i \dot{\boldsymbol{v}}_{ci}$$

$$\boldsymbol{\mu}_i + \boldsymbol{f}_i \times \boldsymbol{r}_{i-1,ci} - \boldsymbol{\mu}_{i+1} - \boldsymbol{f}_{i+1} \times \boldsymbol{r}_{i,ci} = \frac{d}{dt}\left(\boldsymbol{I}_i \boldsymbol{\omega}_i\right)$$

$$\begin{aligned}
\frac{d}{dt}\left(\boldsymbol{I}_i \boldsymbol{\omega}_i\right) &= \dot{\boldsymbol{R}}_i{}^i \boldsymbol{I}_i \boldsymbol{R}_i^T \boldsymbol{\omega}_i + \boldsymbol{R}_i{}^i \boldsymbol{I}_i \dot{\boldsymbol{R}}_i^T \boldsymbol{\omega}_i + \boldsymbol{R}_i{}^i \boldsymbol{I}_i \boldsymbol{R}_i^T \dot{\boldsymbol{\omega}}_i \\
&= \boldsymbol{S}(\boldsymbol{\omega}_i) \boldsymbol{R}_i{}^i \boldsymbol{I}_i \boldsymbol{R}_i^T \boldsymbol{\omega}_i + \boldsymbol{R}_i{}^i \boldsymbol{I}_i \boldsymbol{R}_i^T \boldsymbol{S}^T(\boldsymbol{\omega}_i) \boldsymbol{\omega}_i + \boldsymbol{R}_i{}^i \boldsymbol{I}_i \boldsymbol{R}_i^T \dot{\boldsymbol{\omega}}_i \\
&= \boldsymbol{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \boldsymbol{I}_i \boldsymbol{\omega}_i
\end{aligned}$$

- Linear acceleration of the link center of mass (CoM)

$$\dot{\boldsymbol{v}}_{ci} = \dot{\boldsymbol{v}}_i + \dot{\boldsymbol{\omega}}_i \times \boldsymbol{r}_{i,ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \boldsymbol{r}_{i,ci})$$

- Backward recursion on forces and moments

$$\boldsymbol{f}_i = \boldsymbol{f}_{i+1} + m_i(\dot{\boldsymbol{v}}_i - \boldsymbol{g}_0) + \dot{\boldsymbol{\omega}}_i \times m_i \boldsymbol{r}_{i,ci} + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times m_i \boldsymbol{r}_{i,ci})$$

$$\boldsymbol{\mu}_i = -\boldsymbol{f}_i \times \boldsymbol{r}_{i-1,ci} + \boldsymbol{\mu}_{i+1} + \boldsymbol{f}_{i+1} \times \boldsymbol{r}_{i,ci} + \boldsymbol{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \boldsymbol{I}_i \boldsymbol{\omega}_i$$

- Projection on joint i axis

$$\tau_{qi} = \begin{cases} \boldsymbol{\mu}_i^T \boldsymbol{z}_{i-1} & \text{for a revolute joint} \\ \boldsymbol{f}_i^T \boldsymbol{z}_{i-1} & \text{for a prismatic joint} \end{cases}$$

- Addition of motor inertia and joint dissipation to the actuation torque τ_i at joint i

$$\tau_{qi} + I_{mi} n_{ri}^2 \ddot{q}_i = \tau_i + \tau_{Bi}$$

Recursive Algorithm in Moving Frames

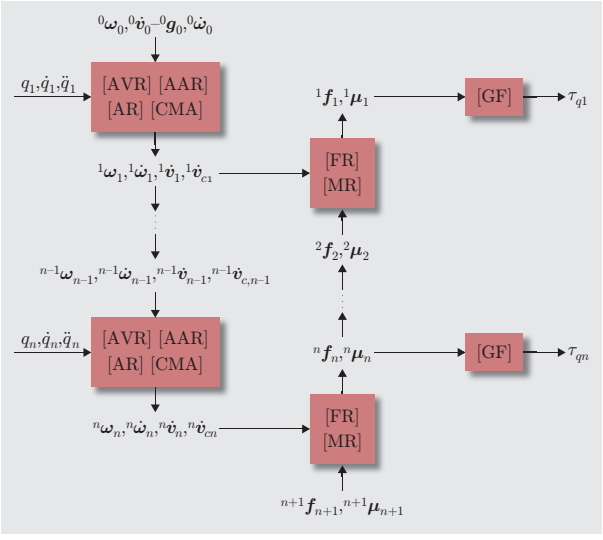
- Forward recursion in moving frames

$$
\begin{aligned}
&\text{[AVR]} = \text{Angular Velocity Recursion} \\
&{}^{i}\boldsymbol{\omega}_i = {}^{i-1}\boldsymbol{R}_i^T \left({}^{i-1}\boldsymbol{\omega}_{i-1} + (1-\sigma_i)\,\dot{\theta}_i \boldsymbol{z}_0 \right) \\
&\text{[AAR]} = \text{Angular Acceleration Recursion} \\
&{}^{i}\dot{\boldsymbol{\omega}}_i = {}^{i-1}\boldsymbol{R}_i^T \left({}^{i-1}\dot{\boldsymbol{\omega}}_{i-1} + (1-\sigma_i) \left(\ddot{\theta}_i \boldsymbol{z}_0 + \dot{\theta}_i\, {}^{i-1}\boldsymbol{\omega}_{i-1} \times \boldsymbol{z}_0 \right) \right) \\
&\text{[AR]} = \text{Acceleration Recursion} \\
&{}^{i}\dot{\boldsymbol{v}}_i = {}^{i-1}\boldsymbol{R}_i^T \left({}^{i-1}\dot{\boldsymbol{v}}_{i-1} + \sigma_i \ddot{d}_i \boldsymbol{z}_0 \right) + 2\,\sigma_i \dot{d}_i\, {}^{i}\boldsymbol{\omega}_i \times {}^{i-1}\boldsymbol{R}_i^T \boldsymbol{z}_0 \\
&\qquad = + {}^{i}\dot{\boldsymbol{\omega}}_i \times {}^{i}\boldsymbol{r}_{i-1,i} + {}^{i}\boldsymbol{\omega}_i \times ({}^{i}\boldsymbol{\omega}_i \times {}^{i}\boldsymbol{r}_{i-1,i}) \\
&\text{[CMA]} = \text{Center of Mass Acceleration} \\
&{}^{i}\dot{\boldsymbol{v}}_{ci} = {}^{i}\dot{\boldsymbol{v}}_i + {}^{i}\dot{\boldsymbol{\omega}}_i \times {}^{i}\boldsymbol{r}_{i,ci} + {}^{i}\boldsymbol{\omega}_i \times ({}^{i}\boldsymbol{\omega}_i \times {}^{i}\boldsymbol{r}_{i,ci})
\end{aligned}
$$

- Backward recursion and generalized force in moving frames

$$
\begin{aligned}
[\text{FR}] &= \text{Force Recursion} \\
{}^{i}\boldsymbol{f}_i &= {}^{i}\boldsymbol{R}_{i+1}\,{}^{i+1}\boldsymbol{f}_{i+1} + m_i\,{}^{i}\dot{\boldsymbol{v}}_{ci} \\
[\text{MR}] &= \text{Moment Recursion} \\
{}^{i}\boldsymbol{\mu}_i &= -\,{}^{i}\boldsymbol{f}_i \times {}^{i}\boldsymbol{r}_{i-1,ci} + {}^{i}\boldsymbol{R}_{i+1}\,{}^{i+1}\boldsymbol{\mu}_{i+1} \\
&= +\,{}^{i}\boldsymbol{R}_{i+1}\,{}^{i+1}\boldsymbol{f}_{i+1} \times {}^{i}\boldsymbol{r}_{i,ci} + {}^{i}\boldsymbol{I}_i\,{}^{i}\dot{\boldsymbol{\omega}}_i + {}^{i}\boldsymbol{\omega}_i \times {}^{i}\boldsymbol{I}_i\,{}^{i}\boldsymbol{\omega}_i \\
[\text{GF}] &= \text{Generalized Force} \\
\tau_{qi} &= \begin{cases} {}^{i}\boldsymbol{\mu}_i^T\,{}^{i-1}\boldsymbol{R}_i^T\boldsymbol{z}_0 & \text{(revolute joint)} \\ {}^{i}\boldsymbol{f}_i^T\,{}^{i-1}\boldsymbol{R}_i^T\boldsymbol{z}_0 & \text{(prismatic joint)} \end{cases}
\end{aligned}
$$

- NE recursive algorithm for inverse dynamics



Closed Kinematic Chains

- Use forward recursion of velocities and accelerations to the links of the equivalent tree-structured open-chain arm
- Compute $\boldsymbol{q}_o$, $\dot{\boldsymbol{q}}_o$, $\ddot{\boldsymbol{q}}_o$ from $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ using the constraint relations and their time derivatives
- Use backward recursion to the links of the equivalent tree-structured open-chain arm
- For a link connected to branches, replace $\boldsymbol{f}_{i+1}$ by $\boldsymbol{f}_{i+1'} + \boldsymbol{f}_{i+1''}$ and $\boldsymbol{\mu}_{i+1}$ by $\boldsymbol{\mu}_{i+1'} + \boldsymbol{\mu}_{i+1''}$
- Compute τ_{oi} as output of the NE algorithm, then compute τ_{qi} at the actuated joints (using virtual works)
- Include motor inertias (at the actuated joints) and friction (local to any joint) as in standard serial chains

Summary of Dynamic Formulations

- Comparison of the two presented formulations
  - both EL and NE formulations compute the relationship between joint torques and motion of the structure
  - each formulation has its pros and cons
- Computational issues illustrated using the NE algorithm
  - full inverse dynamics computation and beyond
  - use also for direct dynamics

Comparison of EL and NE

- EL formulation characteristics
  - looks at the multibody robot as a whole, based on kinetic and potential energy contributions
  - provides equations of motion in compact analytical form
  - explicit expressions of inertia matrix, centrifugal and Coriolis vector (and its factorization), gravitational vector and structural properties
  - advantageous for mechanical design assessment and for development and analysis of motion control laws
  - applicable to serial and parallel branches once generalized coordinates are identified
  - easy to add effects beyond rigidity, such as joint elasticity or link deformation
  - constraint forces at joints are automatically eliminated
  - lends itself naturally to symbolic manipulation languages

- NE formulation characteristics
  - derives dynamic equations from balance of forces and moments on each link or body
  - modular structure
  - recursive structure for serial and branched robots
  - computationally efficient numerical method for inverse and direct dynamics
  - allows explicit computation of constraint forces for joint stress evaluation
  - best tailored for real-time model-based control schemes
  - control torques can be computed numerically without the need of building explicitly the complete dynamic model

Computation of Dynamic Terms with the NE Algorithm

- Consider a serial manipulator with n joints and assume that
  - only joint torques $\boldsymbol{\tau} \in \mathbb{R}^n$ are applied
  - dissipative effects are neglected
  - no end-effector interaction wrenches are present (unless specified)
- NE algorithm evaluation can be represented as an algebraic function

$$\boldsymbol{u} = \mathrm{NEA}_{\boldsymbol{\alpha}}(\mathbf{arg}_1, \mathbf{arg}_2, \mathbf{arg}_3)$$

with 3 arguments and a parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^3$ that accounts for gravity

- $\boldsymbol{\alpha} = {}^0\boldsymbol{g}_0$ presence of gravity
- $\boldsymbol{\alpha} = \mathbf{0}$ absence of gravity

- NEA needs
  - number n of joints
  - joint type information (e.g., indices $\sigma_i = \{0, 1\}$)
  - table of DH kinematic parameters
  - list of dynamic parameters of links and motors
- A call of NEA requires a forward and a backward recursion, each of n steps
- Computational complexity grows linearly with n, i.e., $O(n)$
- Addition of motor inertias and of friction terms is local to each joint and does not affect computational complexity
- Several examples of use of NEA are presented next

- Inverse dynamics by a single call of NEA

$$\boldsymbol{\tau}_d = \boldsymbol{M}(\boldsymbol{q}_d)\ddot{\boldsymbol{q}}_d + \boldsymbol{c}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d) + \boldsymbol{g}(\boldsymbol{q}_d) = \mathsf{NEA}^{0}_{\boldsymbol{g}_0}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d)$$

  where the desired motion is specified by $\boldsymbol{q}_d$, $\dot{\boldsymbol{q}}_d$, $\ddot{\boldsymbol{q}}_d$

- NE computation is typically more efficient than evaluating all symbolic terms in EL model, especially for large n

- Gravity vector

$$\boldsymbol{g}(\boldsymbol{q}) = \mathsf{NEA}^{0}_{\boldsymbol{g}_0}(\boldsymbol{q}, \boldsymbol{0}, \boldsymbol{0})$$

- Coriolis and centrifugal vector

$$\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathsf{NEA}^{\boldsymbol{0}}_{\boldsymbol{0}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{0})$$

- i-th column of inertia matrix

$$\boldsymbol{m}_i(\boldsymbol{q}) = \mathsf{NEA}^{\boldsymbol{0}}_{\boldsymbol{0}}(\boldsymbol{q}, \boldsymbol{0}, \boldsymbol{e}_i)$$

  where $\boldsymbol{e}_i$ is the i-th column of the $n \times n$ identity matrix

- Generalized momentum

$$\boldsymbol{p} = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} = \mathsf{NEA}_{\mathbf{0}}(\boldsymbol{q}, \mathbf{0}, \dot{\boldsymbol{q}})$$

- Feedback linearizing control law

$$\boldsymbol{\tau}_{\mathrm{FBL}} = \mathsf{NEA}_{0_{\boldsymbol{g}_0}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{a})$$

$$\boldsymbol{a} = \ddot{\boldsymbol{q}}_d + \boldsymbol{K}_D\dot{\boldsymbol{e}} + \boldsymbol{K}_P\boldsymbol{e}, \qquad \boldsymbol{e} = \boldsymbol{q}_d - \boldsymbol{q}$$

- In case of end-effector interaction, joint torques to apply an end-effector wrench on the environment

$$\boldsymbol{\tau}_{\mathrm{env}} = -\boldsymbol{J}_e^T(\boldsymbol{q})\boldsymbol{w} = \mathsf{NEA}_{\mathbf{0}}(\boldsymbol{q}, \mathbf{0}, \mathbf{0})$$

where $\boldsymbol{w} = (\boldsymbol{f}, \boldsymbol{\mu})$ is used in the initialization of the backward NE recursion

- Mixed velocity term $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\, \dot{\boldsymbol{q}}'$ with $\dot{\boldsymbol{q}}' \neq \dot{\boldsymbol{q}}$
  - needed in some control laws, where $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ should be skew-symmetric
  - use the exchange identity valid for Christoffel factorization matrices

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}')(\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}') = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}')\dot{\boldsymbol{q}}' + 2\,\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}'$$

- Three calls of NEA give

$$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}' = \frac{1}{2}\left(\mathsf{NEA}_{\mathbf{0}}(\boldsymbol{q}, \dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}', \mathbf{0}) - \mathsf{NEA}_{\mathbf{0}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \mathbf{0}) - \mathsf{NEA}_{\mathbf{0}}(\boldsymbol{q}, \dot{\boldsymbol{q}}', \mathbf{0})\right)$$

- The i-th column $\boldsymbol{c}_i$ of $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ can be computed as

$$\boldsymbol{c}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \frac{1}{2}\left(\mathsf{NEA}_{\mathbf{0}}(\boldsymbol{q}, \dot{\boldsymbol{q}} + \boldsymbol{e}_i, \mathbf{0}) - \mathsf{NEA}_{\mathbf{0}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \mathbf{0}) - \mathsf{NEA}_{\mathbf{0}}(\boldsymbol{q}, \boldsymbol{e}_i, \mathbf{0})\right)$$

- Direct dynamics at time t_k with state $(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and motor torque $\boldsymbol{\tau}$ as input
- EL formulation provides the acceleration as

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q}) \left(\boldsymbol{\tau} - \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}) \right)$$

  to be used in a numerical integration step Δt to obtain $\boldsymbol{q}(t_{k+1})$ and $\dot{\boldsymbol{q}}(t_{k+1})$
- Using instead NEA, evaluate quadratic velocity and gravity terms

$$\boldsymbol{n} = \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \mathsf{NEA}_{\boldsymbol{g}_0}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{0})$$

- Evaluate the columns of inertia matrix $\boldsymbol{M}$ using n calls
  - overall complexity $O(n^2)$
- Invert numerical matrix $\boldsymbol{M}$ to get $\boldsymbol{M}^{-1}$
  - inversion has complexity $O(n^3)$, with low coefficient
- Compute $\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1} (\boldsymbol{\tau} - \boldsymbol{n})$ —equivalently, solve $\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{\tau} - \boldsymbol{n}$ for $\ddot{\boldsymbol{q}}$
- NE formulation provides an $O(n^3)$ algorithm for direct dynamics
- Articulated-body algorithms compute $\ddot{\boldsymbol{q}}$ without building $\boldsymbol{M}$, with complexity $O(n)$

Dynamic Scaling of Trajectories

- A trajectory planned only from geometric or kinematic aspects may be dynamically unfeasible or overly conservative
- The torque profile $\boldsymbol{\tau}(t)$ needed to move the robot along a trajectory $\boldsymbol{q}(t)$, $t \in [0, T]$, planned in configuration space is computed using inverse dynamics

$$\boldsymbol{\tau}(t) = \boldsymbol{M}(\boldsymbol{q}(t))\, \ddot{\boldsymbol{q}}(t) + \boldsymbol{c}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) + \boldsymbol{g}(\boldsymbol{q}(t))$$

- The trajectory $\boldsymbol{q}(t)$ is dynamically feasible if the actuator torque limits are satisfied

$$|\tau_i(t)| \leq \tau_{i,\max} \qquad t \in [0, T] \quad i = 1, \ldots, n$$

- Apply now a uniform time scaling with $r = k\, t \in [0, T_s]$
  - k is a positive constant and $T_s = k\, T$
- The scaled trajectory is denoted by $\boldsymbol{q}_s(r)$ and its torque by $\boldsymbol{\tau}_s(r)$

$$\boldsymbol{q}(t) = \boldsymbol{q}_s(r) \qquad \dot{\boldsymbol{q}}(t) = k\, \dot{\boldsymbol{q}}_s(r) \qquad \ddot{\boldsymbol{q}}(t) = k^2 \ddot{\boldsymbol{q}}(r)$$

- Substituting the time-warp relationships in inverse dynamics yields

$$\boldsymbol{\tau}(t) = k^2 \left(\boldsymbol{M}(\boldsymbol{q}_s(r))\, \ddot{\boldsymbol{q}}_s(r) + \boldsymbol{c}(\boldsymbol{q}_s(r), \dot{\boldsymbol{q}}_s(r)) \right) + \boldsymbol{g}(\boldsymbol{q}(t))$$

  - the inertial and velocity terms scale with k^2
  - the gravitational term is unaffected since it depends only on the configuration

- Torque of the scaled trajectory

$$\boldsymbol{\tau}_s(r) = \boldsymbol{M}(\boldsymbol{q}_s(r))\, \ddot{\boldsymbol{q}}_s(r) + \boldsymbol{c}(\boldsymbol{q}_s(r), \dot{\boldsymbol{q}}_s(r)) + \boldsymbol{g}(\boldsymbol{q}_s(r))$$

- Comparing the two torque expressions gives the general relationship

$$\boldsymbol{\tau}(t) - \boldsymbol{g}(\boldsymbol{q}(t)) = k^2 \left(\boldsymbol{\tau}_s(r) - \boldsymbol{g}(\boldsymbol{q}_s(r)) \right)$$

- Using $r = k\,t$, one obtains

$$\boldsymbol{\tau}_s(kt) = \frac{1}{k^2}\left(\boldsymbol{\tau}(t) - \boldsymbol{g}(\boldsymbol{q}(t))\right) + \boldsymbol{g}(\boldsymbol{q}(t))$$

- Uniform time scaling by k causes a scaling by $1/k^2$ of the torque associated with inertial and velocity terms only, while the gravitational torque is unchanged
- In practice, actuator limits dominate gravitational load at all configurations

$$\tau_{i,\max} > |g_i(\boldsymbol{q})| \qquad \text{for all } \boldsymbol{q} \quad i = 1, \ldots, n$$

- Under this assumption, feasibility can be recovered by sufficiently slowing down the motion, i.e., using a $k > 1$ that is large enough
- When $\boldsymbol{q}(t)$ is dynamically unfeasible, one is interested in the smallest $k^\star > 1$ such that $\boldsymbol{q}_s(k^\star t)$ is feasible

- Compute first, for $i = 1, \ldots, n$

$$k_i = \max\left(1, \max_{t \in [0,T]} \left(\frac{\tau_i(t) - g_i(\boldsymbol{q}(t))}{\tau_{i,\max} - g_i(\boldsymbol{q}(t))}, \frac{g_i(\boldsymbol{q}(t)) - \tau_i(t)}{\tau_{i,\max} + g_i(\boldsymbol{q}(t))} \right)\right) \geq 1$$

- The optimal factor is

$$k^\star = \max_{i=1,\ldots,n} \sqrt{k_i} > 1$$

- The scaled trajectory $\boldsymbol{q}_s(k^\star t)$ is dynamically feasible and (at least) one torque saturates its limit at some time instant
- When the trajectory $\boldsymbol{q}(t)$ is conservative, a similar procedure can be applied to uniformly speed up motion using the largest factor $k^\star < 1$ that preserves feasibility
- Uniform time scaling may be severely suboptimal, motivating nonuniform time scaling for minimum-time motion along a given geometric path under torque limits

Task-Space Dynamic Model

- The configuration-space dynamic model is naturally expressed in terms of the n-dimensional generalized coordinates $\boldsymbol{q}$
- An alternative is to write the equations of motion in task space, relating generalized forces to the m-dimensional task variables $\boldsymbol{y}$
- Redundant DoFs ($n > m$) and task singularities must be carefully handled
- A task-space model derived via EL represents a complete description only in the nonredundant case
- For redundant robots, internal motions are not reflected at the task level

- Consider for simplicity the configuration-space dynamic model without interaction and friction forces, and solve for the joint accelerations

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q}) \left(\boldsymbol{\tau} - \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}) \right)$$

- Second-order differential kinematics relates $\ddot{\boldsymbol{q}}$ to task acceleration

$$\ddot{\boldsymbol{y}} = \boldsymbol{J}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q}) \dot{\boldsymbol{q}}$$

- Task acceleration in terms of joint torques

$$\ddot{\boldsymbol{y}} = \boldsymbol{J}(\boldsymbol{q}) \boldsymbol{M}^{-1}(\boldsymbol{q}) \left(\boldsymbol{\tau} - \boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}) \right) + \dot{\boldsymbol{J}}(\boldsymbol{q}) \dot{\boldsymbol{q}}$$

- In the nonredundant case ($m = n$) and far from singularities, any joint torque can be written as

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q}) \, \boldsymbol{\gamma}$$

  a one-to-one mapping between $\boldsymbol{\tau}$ and the generalized forces $\boldsymbol{\gamma}$ performing work on $\boldsymbol{y}$

- Task-space inertia matrix

$$\boldsymbol{\Lambda}(\boldsymbol{q}) = (\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}^{T}(\boldsymbol{q}))^{-1} = \boldsymbol{J}^{-T}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}^{-1}(\boldsymbol{q})$$

- Task-space dynamic model

$$\boldsymbol{\Lambda}(\boldsymbol{q})\ddot{\boldsymbol{y}} + \boldsymbol{\psi}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\eta}(\boldsymbol{q}) = \boldsymbol{\gamma}$$

with velocity and gravity terms in task space

$$\boldsymbol{\psi}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\Lambda}(\boldsymbol{q}) \left(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}} \right)$$
$$\boldsymbol{\eta}(\boldsymbol{q}) = \boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{g}(\boldsymbol{q})$$

- For nonredundant robots, task variables $\boldsymbol{y}$ form a set of generalized coordinates far from singularities

- In the redundant case ($m < n$), infinitely many joint torques yield the same task-space generalized force $\boldsymbol{\gamma}$
- Decompose a generic torque vector as

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\gamma} + (\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{H}(\boldsymbol{q}))\boldsymbol{\tau}_N$$

  where $\boldsymbol{H}$ is a generalized inverse of $\boldsymbol{J}^T$, so that $\boldsymbol{J}^T\boldsymbol{H}\boldsymbol{J}^T = \boldsymbol{J}^T$, and $\boldsymbol{\tau}_N$ is an arbitrary n-vector
- With $\boldsymbol{J}(\boldsymbol{q})$ full row rank, substituting the decomposition yields

$$\ddot{\boldsymbol{y}} = \boldsymbol{J}\boldsymbol{M}^{-1}(\boldsymbol{J}^T\boldsymbol{\gamma} - \boldsymbol{c} - \boldsymbol{g}) + \boldsymbol{J}\boldsymbol{M}^{-1}(\boldsymbol{I} - \boldsymbol{J}^T\boldsymbol{H})\boldsymbol{\tau}_N + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}$$

- Dynamic consistency requires that the $\boldsymbol{\tau}_N$ component does not contribute to $\ddot{\boldsymbol{y}}$

$$\boldsymbol{J}\boldsymbol{M}^{-1}(\boldsymbol{I} - \boldsymbol{J}^T\boldsymbol{H}) = \boldsymbol{O}$$

- Dynamically consistent choice of $\boldsymbol{H}$

$$\boldsymbol{H} = (\boldsymbol{J}^T)^\dagger_{\boldsymbol{M}} = (\boldsymbol{J}^\dagger_{\boldsymbol{M}})^T = (\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)^{-1}\boldsymbol{J}\boldsymbol{M}^{-1} = \boldsymbol{\Lambda}\,\boldsymbol{J}\boldsymbol{M}^{-1}$$

$$\boldsymbol{J}\boldsymbol{M}^{-1}(\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{J}^\dagger_{\boldsymbol{M}})^T) = \boldsymbol{J}\boldsymbol{M}^{-1}(\boldsymbol{I} - \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{M}^{-1}\boldsymbol{J}^T)^{-1}\boldsymbol{J}\boldsymbol{M}^{-1}) = \boldsymbol{O}$$

  so that the torque decomposition becomes

$$\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{\gamma} + (\boldsymbol{I} - \boldsymbol{J}^\dagger_{\boldsymbol{M}}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q}))^T\boldsymbol{\tau}_N$$

- Setting $\boldsymbol{\tau}_N = \boldsymbol{0}$ gives $\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{q})\gamma$
- $\boldsymbol{J}^\dagger_{\boldsymbol{M}}$ is the dynamically consistent pseudoinverse of $\boldsymbol{J}$

$$\boldsymbol{\tau} = \boldsymbol{\tau} + \boldsymbol{J}^T(\boldsymbol{J}^\dagger_{\boldsymbol{M}})^T\boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{J}^\dagger_{\boldsymbol{M}})^T\boldsymbol{\tau} = \boldsymbol{J}^T(\boldsymbol{J}^\dagger_{\boldsymbol{M}})^T\boldsymbol{\tau} + (\boldsymbol{I} - \boldsymbol{J}^\dagger_{\boldsymbol{M}}\boldsymbol{J})^T\boldsymbol{\tau}$$

$$\boldsymbol{\gamma} = (\boldsymbol{J}^\dagger_{\boldsymbol{M}}(\boldsymbol{q}))^T\boldsymbol{\tau}$$

- For nonredundant robots, $\boldsymbol{J}^\dagger_{\boldsymbol{M}} = \boldsymbol{J}^{-1}$ and this reduces to the inverse of $\boldsymbol{\tau} = \boldsymbol{J}^T\boldsymbol{\gamma}$

- Task-space model for nonredundant and redundant robots

$$\boldsymbol{\Lambda}\,\ddot{\boldsymbol{y}} = \boldsymbol{\gamma} - (\boldsymbol{J}_M^{\dagger})^T(\boldsymbol{c}+\boldsymbol{g}) + \boldsymbol{\Lambda}\dot{\boldsymbol{J}}\dot{\boldsymbol{q}}$$

- Reordering terms gives again the task-space dynamic model

$$\boldsymbol{\Lambda}(\boldsymbol{q})\ddot{\boldsymbol{y}} + \boldsymbol{\psi}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{\eta}(\boldsymbol{q}) = \boldsymbol{\gamma}$$

- In the task-space model the velocity terms can be factorized as $\boldsymbol{\psi}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\,\dot{\boldsymbol{y}}$ so that $\dot{\boldsymbol{\Lambda}}(\boldsymbol{q}) - 2\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is skew-symmetric
- A valid choice is

$$\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}}) = (\boldsymbol{J}_M^{\dagger}(\boldsymbol{q}))^T\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{J}_M^{\dagger}(\boldsymbol{q}) - \boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{J}}(\boldsymbol{q})\boldsymbol{J}_M^{\dagger}(\boldsymbol{q})$$

- As task Jacobian one may choose the geometric Jacobian $\boldsymbol{J}_g$, yielding

$$\boldsymbol{\Lambda}_g(\boldsymbol{q})\dot{\boldsymbol{t}}_e + \boldsymbol{\Gamma}_g(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{t}_e + \boldsymbol{\eta}_g(\boldsymbol{q}) = \boldsymbol{w}_e$$

with the control wrench $\boldsymbol{w}_e$ coming from the decomposition

$$\boldsymbol{\tau} = \boldsymbol{J}_g^T\boldsymbol{w}_e + (\boldsymbol{I} - \boldsymbol{J}_g^T\boldsymbol{H}_g)\boldsymbol{\tau}_N$$

- Task-space direct dynamics: compute $\ddot{\boldsymbol{y}}$ from joint torques $\boldsymbol{\tau}$ at state $(\boldsymbol{q}, \dot{\boldsymbol{q}})$
  - nonredundant case: obtain $\boldsymbol{\gamma}$ by inverting $\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{\gamma}$
  - redundant case: obtain $\boldsymbol{\gamma}$ from $\boldsymbol{\tau}$ via $\boldsymbol{\gamma} = (\boldsymbol{J}_{\boldsymbol{M}}^{\dagger})^T \boldsymbol{\tau}$
- For redundant robots, the task-space model does not capture internal motions for simulation purposes
- Task-space inverse dynamics: compute $\boldsymbol{\gamma}$ from desired $\ddot{\boldsymbol{y}}$, then compute joint torques
  - redundant case: all torques realizing $\boldsymbol{\gamma}$ are given by the dynamically consistent decomposition, with choices of $\boldsymbol{\tau}_N$ generating internal motions

Constrained Dynamic Model

- The robot in free motion has been considered as a fully actuated mechanical system described by the EL model in terms of generalized coordinates $\boldsymbol{q}$
- Suppose now that the robot motion is subject to a set of k physical constraints imposed on its mechanical parts
- Constraints on the end-effector pose variables $\boldsymbol{y}_e$ can be written as $\boldsymbol{f}(\boldsymbol{y}_e) = \boldsymbol{0}$ and, from the direct kinematics $\boldsymbol{y}_e = \boldsymbol{k}_e(\boldsymbol{q})$, as geometric constraints on $\boldsymbol{q}$

$$\boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{f}(\boldsymbol{k}_e(\boldsymbol{q})) = \boldsymbol{0}$$

- The constraint Jacobian is

$$\boldsymbol{A}(\boldsymbol{q}) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{q}} \in I\!R^{k \times n}$$

- Differentiating the constraint gives the kinematic constraints ($\boldsymbol{A}$ full row rank k)

$$\boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}$$

Example

- 2R planar arm with end-effector position $\boldsymbol{p}_e = (p_x, p_y) = \boldsymbol{k}_e(\boldsymbol{q})$ and Jacobian $\boldsymbol{J}(\boldsymbol{q})$
- End-effector position is constrained to move on a circle with center at (x_0, y_0) and radius R —a single scalar constraint ($k = 1$)

$$f(\boldsymbol{p}_e) = (p_x - x_0)^2 + (p_y - y_0)^2 - R^2 = 0$$

$$h(\boldsymbol{q}) = f(\boldsymbol{k}_e(\boldsymbol{q})) = (a_1 c_1 + a_2 c_{12} - x_0)^2 + (a_1 s_1 + a_2 s_{12} - y_0)^2 - R^2 = 0$$

$$\begin{aligned}
\boldsymbol{A}(\boldsymbol{q}) = \frac{\partial h}{\partial \boldsymbol{q}} &= \frac{\partial f}{\partial \boldsymbol{p}_e}\frac{\partial \boldsymbol{p}_e}{\partial \boldsymbol{q}} = \begin{pmatrix} 2(p_x - x_0) & 2(p_y - y_0) \end{pmatrix} \boldsymbol{J}(\boldsymbol{q}) \\
&= \begin{pmatrix} 2(a_1 c_1 + a_2 c_{12} - x_0) & 2(a_1 s_1 + a_2 s_{12} - y_0) \end{pmatrix} \boldsymbol{J}(\boldsymbol{q}) \\
&= \begin{pmatrix} 2x_0\,(a_1 s_1 + a_2 s_{12}) - 2y_0\,(a_1 c_1 + a_2 c_{12}) & 2a_2\,(x_0 s_{12} - y_0 c_{12}) - 2a_1 a_2 s_2 \end{pmatrix}
\end{aligned}$$

- The augmented Lagrangian is

$$\mathcal{L}_a(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{\lambda}^T \boldsymbol{h}(\boldsymbol{q}) = \mathcal{T}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{U}(\boldsymbol{q}) + \boldsymbol{\lambda}^T \boldsymbol{h}(\boldsymbol{q})$$

  where $\boldsymbol{\lambda} \in I\!R^k$ collects the Lagrange multipliers associated with the constraints

- Constrained EL equations

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}_a}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial \mathcal{L}_a}{\partial \boldsymbol{q}}\right)^T = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{q}}\right)^T - \boldsymbol{A}^T(\boldsymbol{q})\boldsymbol{\lambda} = \boldsymbol{\tau}$$

$$\left(\frac{\partial \mathcal{L}_a}{\partial \boldsymbol{\lambda}}\right)^T = \boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{0}$$

  or mixed differential–algebraic equations containing the unknown multipliers

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} + \boldsymbol{A}^T(\boldsymbol{q})\boldsymbol{\lambda}$$
$$\boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{0}$$

Projected Robot Dynamics

- Differentiating $\boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}$ yields

$$\boldsymbol{A}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{A}}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}$$

- A closed-form expression for $\boldsymbol{\lambda}$ is obtained as

$$\boldsymbol{\lambda} = (\boldsymbol{A}_{\boldsymbol{M}}^{\dagger}(\boldsymbol{q}))^{T}\left(\boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{\tau}\right) - (\boldsymbol{A}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{A}^{T}(\boldsymbol{q}))^{-1}\dot{\boldsymbol{A}}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

  where $\boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{c}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})$

- Projected dynamic model

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \left(\boldsymbol{I} - \boldsymbol{A}^{T}(\boldsymbol{q})(\boldsymbol{A}_{\boldsymbol{M}}^{\dagger}(\boldsymbol{q}))^{T}\right)\left(\boldsymbol{\tau} - \boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}})\right) - \boldsymbol{M}(\boldsymbol{q})\boldsymbol{A}_{\boldsymbol{M}}^{\dagger}(\boldsymbol{q})\dot{\boldsymbol{A}}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- $\boldsymbol{I} - \boldsymbol{A}^{T}(\boldsymbol{A}_{\boldsymbol{M}}^{\dagger}(\boldsymbol{q}))^{T}$ is a (dynamic) projection matrix ensuring that, regardless of $\boldsymbol{\tau}$, the solution to these n 2nd-order differential equations complies with the constraints

Reduced Robot Dynamics

- Complete $\boldsymbol{A}(\boldsymbol{q})$ to a square, nonsingular matrix by adding a $(n-k) \times n$ matrix $\boldsymbol{D}(\boldsymbol{q})$

$$\begin{pmatrix} \boldsymbol{A}(\boldsymbol{q}) \\ \boldsymbol{D}(\boldsymbol{q}) \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{E}(\boldsymbol{q}) & \boldsymbol{G}(\boldsymbol{q}) \end{pmatrix}$$

  with $\boldsymbol{E}(\boldsymbol{q}) \in I\!R^{n \times k}$, $\boldsymbol{G}(\boldsymbol{q}) \in I\!R^{n \times (n-k)}$

- The columns of $\boldsymbol{G}$ are a basis of the null space of $\boldsymbol{A}$
- Reduced velocities and accelerations

$$\boldsymbol{u} = \boldsymbol{D}(\boldsymbol{q})\dot{\boldsymbol{q}} \quad \Rightarrow \quad \dot{\boldsymbol{u}} = \boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{D}}(\boldsymbol{q})\dot{\boldsymbol{q}}.$$

- Inverse relationships (exploiting identities in the product of a matrix times its inverse)

$$\dot{\boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{u}$$
$$\ddot{\boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{u}} + \dot{\boldsymbol{G}}(\boldsymbol{q})\boldsymbol{u} = \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{u}} - (\boldsymbol{E}(\boldsymbol{q})\dot{\boldsymbol{A}}(\boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{D}}(\boldsymbol{q}))\boldsymbol{G}(\boldsymbol{q})\boldsymbol{u}$$

Example

- 2R planar arm with end-effector constrained on a circle [reprise]

$$\boldsymbol{D}(\boldsymbol{q}) = \begin{pmatrix} -\frac{1}{2}(p_y - y_0) & \frac{1}{2}(p_x - x_0) \end{pmatrix} \boldsymbol{J}(\boldsymbol{q})$$

$$\det \begin{pmatrix} \boldsymbol{A}(\boldsymbol{q}) \\ \boldsymbol{D}(\boldsymbol{q}) \end{pmatrix} = \det \left(\begin{pmatrix} 2(p_x - x_0) & 2(p_y - y_0) \\ -\frac{1}{2}(p_y - y_0) & \frac{1}{2}(p_x - x_0) \end{pmatrix} \boldsymbol{J}(\boldsymbol{q}) \right) = R^2 \cdot \det \boldsymbol{J}(\boldsymbol{q})$$

- The reduced velocity $u = \boldsymbol{D}(\boldsymbol{q})\,\dot{\boldsymbol{q}} \in \mathbb{R}$ is simply the speed (with sign) of the end-effector along the tangent to the circle
- Admissible joint velocities

$$\dot{\boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{q})\,u = \boldsymbol{J}^{-1}(\boldsymbol{q}) \begin{pmatrix} -2(p_y - y_0) \\ 2(p_x - x_0) \end{pmatrix} \frac{u}{R^2}$$

- Premultiplying the dynamics (being $\boldsymbol{AE} = \boldsymbol{I}$ and $\boldsymbol{AG} = \boldsymbol{O}$)

$$\boldsymbol{E}^T(\boldsymbol{q})\left(\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{\tau}\right) = \boldsymbol{\lambda}$$

$$\boldsymbol{G}^T(\boldsymbol{q})\left(\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right) = \boldsymbol{G}^T(\boldsymbol{q})\,\boldsymbol{\tau}$$

- Substituting in terms of reduced velocities and accelerations

$$\dot{\boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{u}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{u}} + \dot{\boldsymbol{G}}(\boldsymbol{q})\boldsymbol{u} = \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{u}} - \left(\boldsymbol{E}(\boldsymbol{q})\dot{\boldsymbol{A}}(\boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{D}}(\boldsymbol{q})\right)\boldsymbol{G}(\boldsymbol{q})\boldsymbol{u}$$

$$\Downarrow$$

$$\boldsymbol{M}_{\mathrm{red}}(\boldsymbol{q})\dot{\boldsymbol{u}} + \boldsymbol{n}_{\mathrm{red}}(\boldsymbol{q}, \boldsymbol{u}) = \boldsymbol{G}^T(\boldsymbol{q})\,\boldsymbol{\tau}$$

$$\boldsymbol{M}_{\mathrm{red}}(\boldsymbol{q}) = \boldsymbol{G}^T(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})\boldsymbol{G}(\boldsymbol{q})$$

$$\boldsymbol{n}_{\mathrm{red}}(\boldsymbol{q}, \boldsymbol{u}) = \boldsymbol{G}^T(\boldsymbol{q})\left(\boldsymbol{n}(\boldsymbol{q}, \boldsymbol{G}(\boldsymbol{q})\boldsymbol{u}) + \boldsymbol{M}(\boldsymbol{q})\,\dot{\boldsymbol{G}}(\boldsymbol{q})\boldsymbol{u}\right)$$

$$\boldsymbol{\lambda} = \boldsymbol{E}^T(\boldsymbol{q})\Big(\boldsymbol{M}(\boldsymbol{q})\boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{u}} + \boldsymbol{M}(\boldsymbol{q})\big(\boldsymbol{E}(\boldsymbol{q})\dot{\boldsymbol{A}}(\boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{D}}(\boldsymbol{q})\big)\dot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{\tau}\Big)$$

- The reduced dynamic model is lower-dimensional ($n-k$) than the projected dynamic model (n), but computationally more complex (the latter is preferred for simulation purposes)
- The reduced model is very useful for inverse dynamics computation as well as for control design based on feedback linearization

$$\boldsymbol{\tau} = \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{M}(\boldsymbol{q}) \left(\boldsymbol{E}(\boldsymbol{q}) \dot{\boldsymbol{A}}(\boldsymbol{q}) + \boldsymbol{G}(\boldsymbol{q}) \dot{\boldsymbol{D}}(\boldsymbol{q}) \right) \dot{\boldsymbol{q}} + \boldsymbol{M}(\boldsymbol{q}) \boldsymbol{G}(\boldsymbol{q}) \, \boldsymbol{a} - \boldsymbol{A}^T(\boldsymbol{q}) \boldsymbol{f}$$

$$\Downarrow$$

$$\dot{\boldsymbol{u}} = \boldsymbol{a} \qquad \boldsymbol{\lambda} = \boldsymbol{f}$$

i.e., a desired reduced acceleration $\boldsymbol{a}$ and a desired constraint force $\boldsymbol{f}$ have been imposed to the closed-loop system