Modelo Dinâmico de Manipuladores

Formulação de Newton-Euler

Walter Fetter Lages

fetter@ece.ufrgs.br

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Sistemas Elétricos de Automação e Energia

ENG04479 Robótica A

Introdução

- O formalismo de Newton-Euler é baseado nas equações de Newton e de Euler:

$$F = m\dot{v}_c$$

$$N = {}^cI\dot{\omega} + \omega \times {}^cI\omega$$

- Procedimento recursivo
  - Adequado para implementação computacional
  - Não se tem uma expressão em forma fechada
  - Não permite explorar a estrutura do modelo

Dedução da Equação de Euler

$$ {}^{0}N_i = \frac{d}{dt}\left({}^{C_0}I\omega_i\right) $$

onde

${}^{C_0}I$**:** tensor de inércia em relação ao sistema $\{0\}$

ω_i**:** velocidade angular do sistema $\{i\}$ descrita no sistema $\{0\}$

${}^{0}N_i$**:** torque total no centro de massa do elo $\{i\}$ expresso no sistema $\{0\}$

Dedução da Equação de Euler

$$^{C_0}I = {}^0R_i \, {}^{C_i}I \, {}^0R_i^T$$

$$\omega_i = {}^0\omega_i = {}^0R_i \, {}^i\omega_i$$

$${}^0N_i = {}^0R_i \, {}^iN_i$$

$${}^0R_i \, {}^iN_i = \frac{d}{dt}\left({}^0R_i \, {}^{C_i}I \, {}^0R_i^T \, {}^0R_i \, {}^i\omega_i\right)$$

$${}^0R_i {}^iN_i = {}^0R_i \left({}^{C_i}I \, {}^i\dot{\omega}_i\right) + {}^0\omega_i \times {}^0R_i \left({}^{C_i}I \, {}^i\omega_i\right)$$

$${}^iN_i = {}^{C_i}I \, {}^i\dot{\omega}_i + {}^0\omega_i \times {}^{C_i}I \, {}^i\omega_i$$

Copyright (c) Walter Fetter Lages – p.4

Cálculo Aceleração Linear

$$ {}^{A}V_{Q} = {}^{A}R_{B}\,{}^{B}V_{Q} + {}^{A}\Omega_{B} \times {}^{A}R_{B}\,{}^{B}Q $$

Portanto a aceleração será,

$$ {}^{A}\dot{V}_{Q} = \frac{d}{dt}\left({}^{A}R_{B}\,{}^{B}V_{Q}\right) + {}^{A}\dot{\Omega}_{B} \times {}^{A}R_{B}\,{}^{B}Q + {}^{A}\Omega_{B} \times \frac{d}{dt}\left({}^{A}R_{B}\,{}^{B}Q\right) $$

Por outro lado,

$$ {}^{A}V_{Q} = \frac{d}{dt}\left({}^{A}R_{B}\,{}^{B}Q\right) = {}^{A}R_{B}\,{}^{B}V_{Q} + {}^{A}\Omega_{B} \times {}^{A}R_{B}\,{}^{B}Q $$

Cálculo da Aceleração Linear

Similarmente, pode-se obter

$$\frac{d}{dt}\left({}^{A}R_{B}{}^{B}V_{Q}\right) = {}^{A}R_{B}{}^{B}\dot{V}_{Q} + {}^{A}\Omega_{B} \times {}^{A}R_{B}{}^{B}V_{Q}$$

Substituindo-se:

$$\begin{aligned}{}^{A}\dot{V}_{Q} = {}^{A}R_{B}{}^{B}\dot{V}_{Q} + {}^{A}\Omega_{B} \times {}^{A}R_{B}{}^{B}V_{Q} + {}^{A}\dot{\Omega}_{B} \times {}^{A}R_{B}{}^{B}Q + \\ {}^{A}\Omega_{B} \times \left({}^{A}R_{B}{}^{B}V_{Q} + {}^{A}\Omega_{B} \times {}^{A}R_{B}{}^{B}Q\right)\end{aligned}$$

ou

$$\begin{aligned}{}^{A}\dot{V}_{Q} = {}^{A}R_{B}{}^{B}\dot{V}_{Q} + 2{}^{A}\Omega_{B} \times {}^{A}R_{B}{}^{B}V_{Q} + {}^{A}\dot{\Omega}_{B} \times {}^{A}R_{B}{}^{B}Q + \\ {}^{A}\Omega_{B} \times \left({}^{A}\Omega_{B} \times {}^{A}R_{B}{}^{B}Q\right)\end{aligned}$$

Cálculo da Aceleração Linear

Para o caso genérico, quanto o sistema {B} está se movendo em relação ao sistema {A}:

$$
\begin{aligned}
{}^A\dot{V}_Q = {}^A\dot{V}_{Borg} + {}^AR_B\,{}^B\dot{V}_Q + 2\,{}^A\Omega_B \times {}^AR_B\,{}^BV_Q + \\
{}^A\dot{\Omega}_B \times {}^AR_B\,{}^BQ + {}^A\Omega_B \times \left({}^A\Omega_B \times {}^AR_B\,{}^BQ\right)
\end{aligned}
$$

Quando ${}^BQ$ é constante em relação a {B}, ou seja: ${}^BV_Q = {}^B\dot{V}_Q = 0$:

$$
\begin{aligned}
{}^A\dot{V}_Q = {}^A\dot{V}_{Borg} + {}^A\dot{\Omega}_B \times {}^AR_B\,{}^BQ + \\
{}^A\Omega_B \times \left({}^A\Omega_B \times {}^AR_B\,{}^BQ\right)
\end{aligned}
$$

que é usada para calcular a aceleração linear dos elos.

Aceleração Angular

$$^{A}\Omega_C = {}^{A}\Omega_B + {}^{A}R_B\,{}^{B}\Omega_C$$

Diferenciando:

$$^{A}\dot{\Omega}_C = {}^{A}\dot{\Omega}_B + \frac{d}{dt}\left({}^{A}R_B\,{}^{B}\Omega_C\right)$$

$$\frac{d}{dt}\left({}^{A}R_B\,{}^{B}\Omega_C\right) = {}^{A}R_B\,{}^{B}\dot{\Omega}_C + {}^{A}\Omega_B \times {}^{A}R_B\,{}^{B}\Omega_C$$

tem-se

$$^{A}\dot{\Omega}_C = {}^{A}\dot{\Omega}_B + {}^{A}R_B\,{}^{B}\dot{\Omega}_C + {}^{A}\Omega_B \times {}^{A}R_B\,{}^{B}\Omega_C$$

que é usada para calcular a aceleração angular de elos.

Formalismo de Newton-Euler

- procedimento iterativo para calcular os torques nas juntas do robô
- iterações diretas e iterações reversas
- iterações diretas calculam as acelerações e velocidades dos elos do robô a partir da base em direção à garra. As
- iterações reversas calculam os torques da garra para a base do robô

Iterações Diretas

- Velocidades e Acelerações Angulares dos Elos com Juntas Rotacionais

$$^{i-1}\omega_i = {}^{i-1}\omega_{i-1} + {}^{i-1}R_{i-1}\dot{\theta}_i{}^{i-i}\hat{Z}_{i-i}$$

$$^{i-1}\dot{\omega}_i = {}^{i-1}\dot{\omega}_{i-1} + {}^{i-1}R_{i-1}\ddot{\theta}_i{}^{i-i}\hat{Z}_{i-i} + {}^{i-1}\omega_{i-1} \times {}^{i-1}R_{i-1}\dot{\theta}_i{}^{i-i}\hat{Z}_{i-i}$$

ou

$$^{i-1}\omega_i = {}^{i-1}\omega_{i-1} + \dot{\theta}_i{}^{i-i}\hat{Z}_{i-i}$$

$$^{i-1}\dot{\omega}_i = {}^{i-1}\dot{\omega}_{i-1} + \ddot{\theta}_i{}^{i-i}\hat{Z}_{i-i} + {}^{i-1}\omega_{i-1} \times \dot{\theta}_i{}^{i-i}\hat{Z}_{i-i}$$

Iterações Diretas

- Velocidades e Acelerações Angulares dos Elos com Juntas Rotacionais

Para obter-se uma expressão que possa ser utilizada iterativamente, é interessante ter-se ω_i e $\dot{\omega}_i$ representadas no sistema associado ao elo i. Assim,

$$ {}^{i}\omega_i = {}^{i}R_{i-1} \left({}^{i-1}\omega_{i-1} + \dot{\theta}_i {}^{i-i}\hat{Z}_{i-i} \right) $$

e

$$ {}^{i}\dot{\omega}_i = {}^{i}R_{i-1} \left({}^{i-1}\dot{\omega}_{i-1} + \ddot{\theta}_i {}^{i-i}\hat{Z}_{i-i} + {}^{i-1}\omega_{i-1} \times \dot{\theta}_i {}^{i-i}\hat{Z}_{i-i} \right) $$

Iterações Diretas

- Acelerações Lineares dos Elos com Juntas Rotacionais

$$
{}^{i-1}\dot{v}_i = {}^{i-1}\dot{v}_{i-1} + {}^{i-1}\dot{\omega}_i \times {}^{i}R_{i-1}{}^{i-1}P_{iorg} + {}^{i-1}\omega_i \times \left({}^{i-1}\omega_i \times {}^{i}R_{i-1}{}^{i-1}P_{iorg}\right)
$$

portanto,

$$
{}^{i}\dot{v}_i = {}^{i}R_{i-1}{}^{i-1}\dot{v}_{i-1} + {}^{i}\dot{\omega}_i \times {}^{i}R_{i-1}{}^{i-1}P_{iorg} + {}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}R_{i-1}{}^{i-1}P_{iorg}\right)
$$

Iterações Diretas

- Para juntas prismáticas:

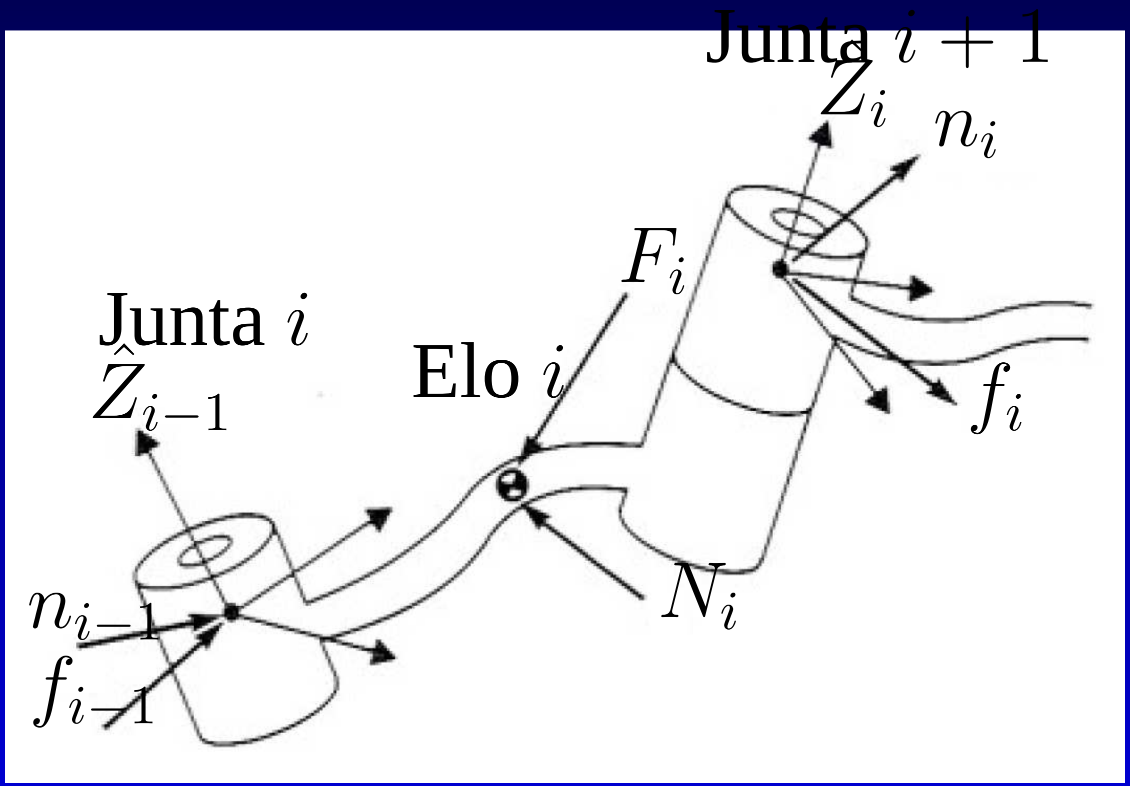
$$ {}^{i}\omega_i = {}^{i}R_{i-1}{}^{i-1}\omega_{i-1} $$

$$ {}^{i}\dot{\omega}_i = {}^{i}R_{i-1}{}^{i-1}\dot{\omega}_{i-1} $$

$$ \begin{aligned} {}^{i}\dot{v}_i = {}^{i}R_{i-1}{}^{i-1}\dot{v}_{i-1} + {}^{i}\dot{\omega}_i \times {}^{i}R_{i-1}{}^{i-1}P_{iorg} + \\ {}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}R_{i-1}{}^{i-1}P_{iorg}\right) + \\ 2{}^{i}\omega_i \times {}^{i}R_{i-1}\dot{d}_i{}^{i-1}\hat{Z}_{i-1} + {}^{i}R_{i-1}\ddot{d}_i{}^{i-1}\hat{Z}_{i-1} \end{aligned} $$

Iterações Diretas

- é necessário ter-se a aceleração no centro de massa do elo, e não na ponta do elo



$$^{i}\dot{v}_{ci} = {}^{i}\dot{v}_i + {}^{i}\dot{\omega}_i \times {}^{i}P_{ci} + {}^{i}\omega_i \times \left({}^{i}\omega_i \times {}^{i}P_{ci}\right)$$

Iterações Reversas

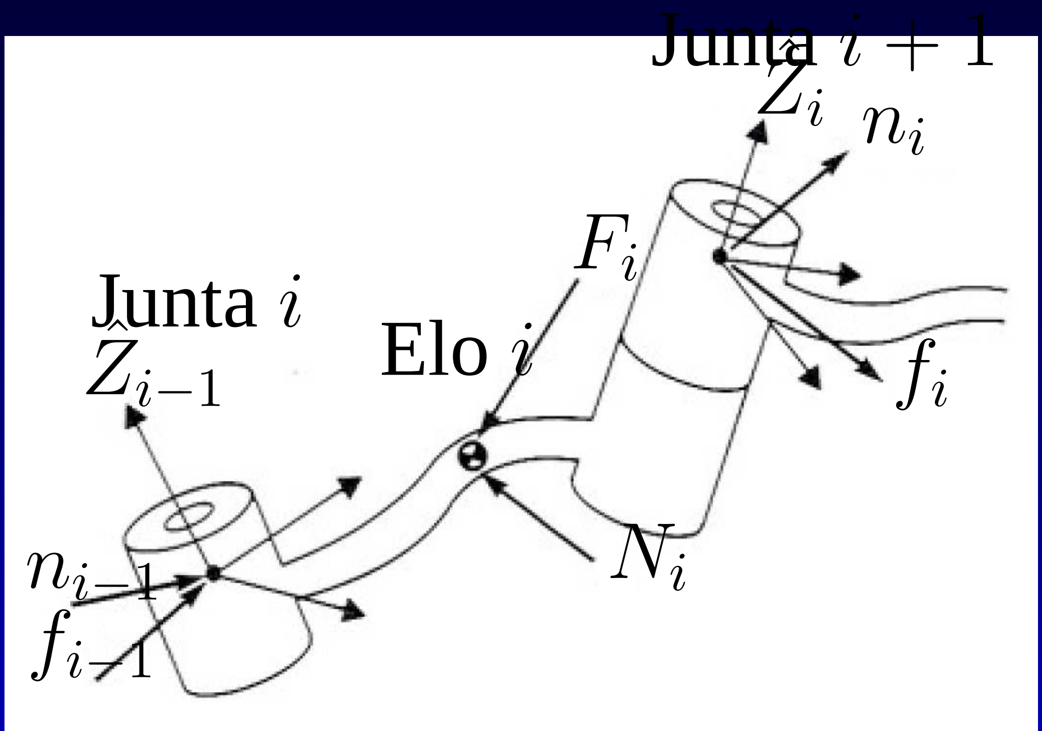
- Torques e forças nos elos

$$ {}^{i}F_i = m_i {}^{i}\dot{v}_{ci} $$

$$ {}^{i}N_i = {}^{ci}I_i {}^{i}\dot{\omega}_i + {}^{i}\omega_i \times {}^{ci}I_i {}^{i}\omega_i $$

Iterações Reversas

- Balanço de forças e momentos



$$ {}^iF_i = {}^iR_{i-1}\,{}^{i-1}f_{i-1} - {}^if_i $$

$$ {}^iN_i = {}^iR_{i-1}\,{}^{i-1}n_{i-1} - {}^in_i + \left({}^iP_{i-1} - {}^iP_{ci}\right) \times {}^iR_{i-1}\,{}^{i-1}f_{i-1} + {}^iP_{ci} \times {}^if_i $$

Iterações Reversas

- Portanto as forças e torques na junta i, serão:

$$ {}^{i-1}f_{i-1} = {}^{i-1}R_i \left({}^iF_i + {}^if_i \right) $$

$$ \begin{aligned} {}^{i-1}n_{i-1} &= {}^{i-1}R_i \left[{}^iN_i + {}^in_i + \right. \\ &+ \left. \left({}^iP_{ci} - {}^iP_{i-1} \right) \times {}^iR_{i-1} {}^{i-1}f_{i-1} - {}^iP_{ci} \times {}^if_i \right] \end{aligned} $$

$\tau_i = {}^{i-1}n_{i-1}^T \, {}^{i-1}\hat{Z}_{i-1}$, se a junta for rotacional, ou

$\mathcal{F}_i = {}^{i-1}f_{i-1}^T \, {}^{i-1}\hat{Z}_{i-1}$, se a junta for prismática.

Procedimento de Newton-Euler

- Inicialização

$$^0\dot{v}_0 = {}^0\omega_0 = {}^0\dot{\omega}_0 = 0$$

$$^{n+1}f_{n+1} = \text{forças externas agindo na garra}$$

$$^{n+1}n_{n+1} = \text{torques externos agindo na garra}$$

Procedimento de Newton-Euler

- Iterações diretas: $i = 1, 2, \ldots, n$

$$
{}^{i}\omega_i = \begin{cases} {}^{i}R_{i-1}\left({}^{i-1}\omega_{i-1} + \dot{\theta}_i\,{}^{i-i}\hat{Z}_{i-i}\right) & \text{, junta rotacional} \\ \\ {}^{i}R_{i-1}\,{}^{i-1}\omega_{i-1} & \text{, junta prismática} \end{cases}
$$

$$
{}^{i}\dot{\omega}_i = \begin{cases} {}^{i}R_{i-1}\left({}^{i-1}\dot{\omega}_{i-1} + \ddot{\theta}_i\,{}^{i-i}\hat{Z}_{i-i} + {}^{i-1}\omega_{i-1} \times \dot{\theta}_i\,{}^{i-i}\hat{Z}_{i-i}\right) & \text{, junta rotacion} \\ \\ {}^{i}R_{i-1}\,{}^{i-1}\dot{\omega}_{i-1} & \text{, junta prismáti} \end{cases}
$$

Interações Diretas

$$
{}^i\dot{v}_i = \begin{cases} {}^iR_{i-1}{}^{i-1}\dot{v}_{i-1} + {}^i\dot{\omega}_i \times {}^iR_{i-1}{}^{i-1}P_{iorg} + \\ {}^i\omega_i \times \left({}^i\omega_i \times {}^iR_{i-1}{}^{i-1}P_{iorg}\right) & \text{, junta rotacional} \\ \\ {}^iR_{i-1}{}^{i-1}\dot{v}_{i-1} + {}^i\dot{\omega}_i \times {}^iR_{i-1}{}^{i-1}P_{iorg} + \\ {}^i\omega_i \times \left({}^i\omega_i \times {}^iR_{i-1}{}^{i-1}P_{iorg}\right) \\ +2{}^i\omega_i \times {}^iR_{i-1}\dot{d}_i{}^{i-1}\hat{Z}_{i-1} + {}^iR_{i-1}\ddot{d}_i{}^{i-1}\hat{Z}_{i-1} & \text{, junta prismática} \end{cases}
$$

$$
{}^i\dot{v}_{ci} = {}^i\dot{v}_i + {}^i\dot{\omega}_i \times {}^iP_{ci} + {}^i\omega_i \times \left({}^i\omega_i \times {}^iP_{ci}\right)
$$

Procedimento de Newton-Euler

- Iterações reversas: $i = n, n-1, \dots, 1$

$$^{i}F_i = m_i\,{}^{i}\dot{v}_{ci}$$

$$^{i}N_i = {}^{ci}I_i\,{}^{i}\dot{\omega}_i + {}^{i}\omega_i \times {}^{ci}I_i\,{}^{i}\omega_i$$

$$^{i-1}f_{i-1} = {}^{i-1}R_i\left({}^{i}F_i + {}^{i}f_i\right)$$

$$^{i-1}n_{i-1} = {}^{i-1}R_i\left[{}^{i}N_i + {}^{i}n_i + \left({}^{i}P_{ci} - {}^{i}P_{i-1}\right) \times {}^{i}R_{i-1}\,{}^{i-1}f_{i-1} - {}^{i}P_{ci} \times {}^{i}f_i\right]$$

$$\tau_i = \begin{cases} {}^{i-1}n_{i-1}^{T}\,{}^{i-1}\hat{Z}_{i-1} & \text{, se a junta for rotacional, ou} \\ {}^{i-1}f_{i-1}^{T}\,{}^{i-1}\hat{Z}_{i-1} & \text{, se a junta for prismática.} \end{cases}$$

Implementação no ROS

- Biblioteca KDL, importada do OROCOS
- `chainidsolver_recursive_newton_euler.hpp`

```
class KDL::ChainIdSolver_RNE
{
  public:
  ChainIdSolver_RNE(const Chain & chain,Vector grav);
  int CartToJnt(const JntArray &q,const JntArray &q_dot,
    const JntArray &q_dotdot,const Wrenches &f_ext,
    JntArray &torques);
  .
  .
};
```