$$ {}^{i}n_{i} = {}^{i}N_{i} + {}_{i+1}^{\ i}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_{i} + {}^{i}P_{i+1} \times {}_{i+1}^{\ i}R\,{}^{i+1}f_{i+1} \ . \tag{6.42} $$

Essas equações são avaliadas elo a elo, começando do elo $n$ e prosseguindo "para dentro" (em inglês, *inward*) em direção à base do robô. Essas *iterações "para dentro" da força* são análogas às iterações de força estática apresentadas no Capítulo 5, exceto que as forças inerciais e torques são agora considerados a cada elo.

Como no caso estático, os torques de junta necessários são encontrados tomando-se o componente $\hat{Z}$ do torque aplicado por um elo ao seu vizinho:

$$ \tau_{i} = {}^{i}n_{i}^{T}\,{}^{i}\hat{Z}_{i} \ . \tag{6.43} $$

Para uma junta $i$ prismática, usamos

$$ \tau_{i} = {}^{i}f_{i}^{T}\,{}^{i}\hat{Z}_{i} \ , \tag{6.44} $$

onde utilizamos o símbolo $\tau$ para uma força linear do atuador.

Observe que, para um robô que se movimenta no espaço livre, ${}^{N+1}f_{N+1}$ e ${}^{N+1}n_{N+1}$ são igualados a zero, de forma que a primeira aplicação das equações para o elo $n$ é muito simples. Se o robô está em contato com o ambiente, as forças e torques devido a esse contato podem ser incluídos no equilíbrio de força através de ${}^{N+1}f_{N+1}$ e ${}^{N+1}n_{N+1}$ diferentes de zero.

## > O algoritmo dinâmico iterativo de Newton-Euler

O algoritmo completo para computar torques de juntas a partir do movimento das juntas é composto por duas partes. Primeiro, velocidades de elos e acelerações são iterativamente computadas, saindo do elo 1 ao elo $n$, e as equações de Newton-Euler são aplicadas a cada elo. Segundo, forças e torques de interação e torques de atuadores de juntas são computados recursivamente, voltando do elo $n$ ao elo 1. As equações estão resumidas a seguir para o caso em que todas as juntas são rotacionais:

Iterações "para fora": $i$: $0 \rightarrow 5$

$$ {}^{i+1}\omega_{i+1} = {}_{\ i}^{i+1}R\,{}^{i}\omega_{i} + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \ , \tag{6.45} $$

$$ {}^{i+1}\dot{\omega}_{i+1} = {}_{\ i}^{i+1}R\,{}^{i}\dot{\omega}_{i} + {}_{\ i}^{i+1}R\,{}^{i}\omega_{i} \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \ , \tag{6.46} $$

$$ {}^{i+1}\dot{v}_{i+1} = {}_{\ i}^{i+1}R\left({}^{i}\dot{\omega}_{i} \times {}^{i}P_{i+1} + {}^{i}\omega_{i} \times \left({}^{i}\omega_{i} \times {}^{i}P_{i+1}\right) + {}^{i}\dot{v}_{i}\right) , \tag{6.47} $$

$$ \begin{aligned} {}^{i+1}\dot{v}_{C_{i+1}} = {} & {}^{i+1}\dot{\omega}_{i+1} \times {}^{i+1}P_{C_{i+1}} \\ & + {}^{i+1}\omega_{i+1} \times \left({}^{i+1}\omega_{i+1} \times {}^{i+1}P_{C_{i+1}}\right) + {}^{i+1}\dot{v}_{i+1} \ , \end{aligned} \tag{6.48} $$

$$ {}^{i+1}F_{i+1} = m_{i+1}\,{}^{i+1}\dot{v}_{C_{i+1}} \ , \tag{6.49} $$

$$ {}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\omega_{i+1} \ . \tag{6.50} $$

Iterações "para dentro": $i$ $6 \rightarrow 1$

$$ {}^{i}f_i = {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i\,, \tag{6.51} $$

$$ \begin{aligned} {}^{i}n_i = {}&{}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{C_i} \times {}^{i}F_i \\ &+ {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1}\,, \end{aligned} \tag{6.52} $$

$$ \tau_i = {}^{i}n_i^T\,{}^{i}\hat{Z}_i\,. \tag{6.53} $$

### > Inclusão da força da gravidade no algoritmo dinâmico

O efeito da carga da gravidade sobre os elos pode ser incluído simplesmente ajustando-se ${}^{0}\dot{v}_0 = G$, em que $G$ tem a magnitude do vetor gravidade, mas apontando na direção oposta. Isso equivale a dizer que a base do robô está acelerando para cima, com uma aceleração de 1 g. Essa aceleração fictícia para cima tem sobre os elos o mesmo efeito que a gravidade teria. Dessa forma, o efeito da gravidade é calculado sem um esforço computacional maior.

## 6.6 ITERATIVO *VERSUS* FORMA FECHADA

As equações (6.46) a (6.53) fornecem um esquema computacional pelo qual, dadas as posições, velocidades e acelerações das juntas, podemos computar os torques de juntas necessários. Como no desenvolvimento das equações para calcular o Jacobiano no Capítulo 5, essas relações podem ser usadas de duas formas: como algoritmo computacional numérico, ou como algoritmo usado analiticamente para desenvolver equações simbólicas.

O uso das equações como algoritmo computacional numérico é atraente porque elas se aplicam a qualquer robô. Uma vez que os tensores de inércia, massa dos elos, vetores $P_{C_i}$ e matrizes ${}^{i+1}_{i}R$ tenham sido especificados para um determinado manipulador, as equações podem ser aplicadas diretamente para estimar os torques das juntas correspondentes a qualquer movimento.

No entanto, com frequência estamos interessados em obter uma percepção melhor da estrutura das equações. Por exemplo, qual é a forma dos termos de gravidade? Como os efeitos da magnitude da gravidade se comparam com os da inércia? Para investigar essas e outras questões, é útil escrever equações dinâmicas de forma fechada. Tais equações podem ser derivadas aplicando-se as equações recursivas de Newton-Euler simbolicamente a $\Theta$, $\dot{\Theta}$ e $\ddot{\Theta}$. Isso é análogo ao que fizemos no Capítulo 5 para derivar a forma simbólica do Jacobiano.

## 6.7 UM EXEMPLO DE EQUAÇÕES DINÂMICAS DE FORMA FECHADA

Aqui computamos as equações dinâmicas de forma fechada para o manipulador planar de dois elos mostrado na Figura 6.6. Por simplicidade, presumimos que a distribuição de massa é extremamente simples: toda a massa existe como uma massa pontual na extremidade distal de cada elo. Essas massas são $m_1$ e $m_2$.

Primeiro, determinamos os valores de várias quantidades que aparecerão nas equações recursivas de Newton-Euler. Os vetores que localizam o centro de massa para cada elo são