Note per la soluzione dell'equilibrio dinamico con Robotics Toolbox

Per la soluzione dell'equilibrio dinamico del robot a 6 gdl considerando anche le proprietà inerziali del tool e della base fissa, occorre aggiungere all'oggetto robot (denominato ad esempio *rob6R*) creato con il comando SerialLink, due link aggiuntivi rappresentativi di tool e base, creando così un oggetto (denominato ad esempio *Brob6RT*) tipo SerialLink avente 8 gdl complessivi.
Seguire la seguente struttura di programmazione:

```
% L = vector of Links 1-6
L(1)=Link('alpha',alfa1,'a',a1,'d',d1,'offset',teta10,'modified','qlim',qlim1);     % robot link 1
L(2)=Link('alpha',alfa2,'a',a2,'d',d2,'offset',teta20,'modified','qlim',qlim2);     % robot link 2
L(3)=Link('alpha',alfa3,'a',a3,'d',d3,'offset',teta30,'modified','qlim',qlim3);     % robot link 3
L(4)=Link('alpha',alfa4,'a',a4,'d',d4,'offset',teta40,'modified','qlim',qlim4);     % robot link 4
L(5)=Link('alpha',alfa5,'a',a5,'d',d5,'offset',teta50,'modified','qlim',qlim5);     % robot link 5
L(6)=Link('alpha',alfa6,'a',a6,'d',d6,'offset',teta60,'modified','qlim',qlim6);     % robot link 6

rob6R= SerialLink(L,'name','rob6R');     % to build up a SerialLink object
% add tool TCP reference frame
rob6R.tool = ATCP6o ;

% inertial data
%Link 1
L(1).m = m1;   % mass
L(1).r = b1;   % position COG
L(1).I = I1;   % inertia tensor
%Link 2
…
%Link 6
…

% T = link tool
T(1)=Link('alpha',alfaT,'a',aT,'d',dT,'offset',tetaT0,'modified');  %tool
tool= SerialLink(T,'name','tool');     % to build up a SerialLink object

% inertial data assignment
%tool
T(1).m = mT;   % mass
T(1).r = bT;   % position COG
T(1).I = IT;   % inertia tensor

% B = link base
B(1)=Link('alpha',alfaB,'a',aB,'d',dB,'offset',tetaB0,'modified');  %base
base= SerialLink(B,'name','base');     % to build up a SerialLink object

% inertial data assignment
%base
B(1).m = m0;   % mass
B(1).r = b0;   % position COG
B(1).I = I0;   % inertia tensor

% base + robot + tool
Brob6RT=SerialLink([base rob6R tool],'name','Brob6RT');
% add tool TCP reference frame (TCP wrt dhT = TCP wrt 6)
Brob6RT.tool = ATCP6o ;

% display rob6
rob6R

% display base+rob6+tool
Brob6RT
```

Verificare la creazione degli oggetti SerialLink con i comandi:

```
% display rob6R
rob6R
% plot rob6R
rob6R.plot
% display base+rob6+tool
Brob6RT
%plot base+rob6+tool
Brob6RT.plot
```

Si osserva che le proprietà inerziali (massa, posizione del baricentro, tensore di inerzia) di ciascun link devono essere definite nel sistema di riferimento di Denavit-Hartenberg di ciascun link. Occorre quindi trasformare le relative informazioni ottenute dalla modellazione CAD ai sistemi di riferimento DH per ciascun link.

Algoritmo Recursive Newton Euler (RNE)

Per risolvere l'equilibrio dinamico usare l'algoritmo RNE implementato con la funzione **rne_mdh_POLITO.m**

Note d'uso:

1. Copiare la funzione **rne_mdh_POLITO.m** nella directory **…\rvctools\robot\@SerialLink**
2. Usare comando **help rne_mdh_POLITO** per dettagli su uso della funzione

```
[TAU,WBASE] = R.rne_mdh_POLITO([Q,QD,QDD],GRAV,FEXT);

TAU=azioni motrici richieste nei giunti per l'equilibrio dinamico
WBASE=wrench esercitato sulla base per l'equilibrio dinamico
Q=vettore valori gdl nei giunti (1xN giunti)
QD=vettore velocità gdl nei giunti (1xN giunti)
QDD=vettore accelerazione gdl nei giunti (1xN giunti)
GRAV=vettore accelerazione di gravità da assegnare alla base espresso nel sistema di riferimento di
base per considerare i pesi propri dei link (GRAV=[0 0 9.81]';  %m/s2)
FEXT=vettore di forze generalizzate (1x6) esercitate dal link N sull'esterno, espresso nel sistema
di riferimento secondo Denavit-Hartenberg del link N
```