

## *SymDyn3D* - UN CODICE MULTIBODY PER LA GENERAZIONE DELLE EQUAZIONI DEL MOTO IN FORMA SIMBOLICA

P.P. VALENTINI, E. PENNESTRI, G. D'ALESSIO, G. FERRARI

*Dipartimento di Ingegneria Meccanica, Università degli Studi di Roma "Tor Vergata", Roma*

### SOMMARIO

Le applicazioni della dinamica dei sistemi multibody sono sempre più frequenti e, in taluni casi, richiedono la disponibilità delle equazioni del moto in forma simbolica. Inoltre, un'efficace didattica sulle metodologie multibody presuppone la disponibilità dei codici sorgente. Tali motivazioni hanno spinto gli autori a sviluppare, in linguaggio Maple V, il codice *SymDyn3D* in grado di generare, in forma simbolica, le equazioni del moto per una larga classe di modelli tridimensionali. *SymDyn3D* fornisce tutto il codice sorgente Fortran necessario per l'integrazione numerica delle equazioni. Inoltre, la struttura di un semplice programma di post-processing viene discussa. Le finalità del presente lavoro sono quelle di illustrare alcune delle potenzialità di *SymDyn3D*.

### ABSTRACT

Multibody dynamic analysis may require the availability of the equations of motion in symbolic form. For this purpose the authors developed the code *SymDyn3D*, entirely written in Maple language. The index of the differential-algebraic system of equations generated can be varied. A simple post-processor for the visualization of the motion has been also developed. Purpose of this paper is the discussion of the main capabilities of *SymDyn3D*.

### 1. INTRODUZIONE

Le tecniche multibody costituiscono un potente strumento per l'impostazione di modelli di analisi di sistemi meccanici. *SymDyn3D* è un codice di calcolo, sviluppato presso l'università di Roma "Tor Vergata", per la generazione automatica ed in forma simbolica del sistema di equazioni algebrico-differenziali (DAE) che governano la cinematica e la dinamica di un modello di corpi rigidi spazialmente vincolati. Il codice qui descritto costituisce la naturale estensione di uno similare, limitato al caso piano, e proposto alcuni anni orsono [1, 2].

L'idea della generazione in forma simbolica delle equazioni del moto, non è certamente nuova ed i primi studi sull'argomento sono dovuti a T.R. Kane e D.A. Levinson (AUTOLEV) [3], W.O. Schielen e E.J. Kreuzer (NEWEUL) [4], J. Wittenburg e U. Wolz (MESAVERDE) [5], M.W. Sayers (AUTOSIM) [6], M.G. Hollars e D.E. Rosenthal (SD/FAST) [7]. I software suddetti, pur se si sviluppano sulla base di formalismi dinamici differenti, condividono l'approccio sistematico caratteristico delle tecniche multibody.

*SymDyn3D* non si limita alla semplice generazione simbolica delle equazioni del moto, ma fornisce anche tutto il codice Fortran richiesto per la loro soluzione numerica.

Quest'ultima caratteristica non è presente nei programmi commercialmente disponibili (e.g. SD-FAST) in cui all'utente è comunque richiesta la stesura di un programma principale che richiami le varie procedure generate dal codice multibody.

In particolare, l'output di *SymDyn3D* è stato calibrato per l'impiego della subroutine RADAU5 [15] nell'integrazione numerica del sistema DAE. La scelta del metodo di integrazione è giustificata dalla disamina delle prestazioni computazionali di vari codici di integrazione presentata da Mingardi e Pennestri [8].

Una caratteristica peculiare di `SymDyn3D`, non presente in programmi simili, è la possibilità di scegliere l'indice DAE della formulazione. Ciò oltre ai vantaggi computazionali connessi, consente di adattare il programma di simulazione a particolari esigenze. Infatti, non è raro il caso di dover sviluppare dei modelli in cui siano prescritte le leggi di moto di taluni corpi rilevate sperimentalmente tramite accelerometri.

Quest'ultima caratteristica si è rivelata particolarmente utile nello sviluppo di un manichino che riproduce la dinamica vibrazionale di passeggeri di autovetture [9].

I modelli che `SymDyn3D` è in grado di trattare sono costituiti da corpi rigidi vincolati tra loro da coppie cinematiche standard quali rotoidale, prismatica, cilindrica, etc.

Su tali corpi possono agire forze esterne di varia natura e viscoelastiche.

La formula di Kutzbach viene ritenuta valida per il calcolo dei gradi di libertà del sistema. L'attuale versione del programma funziona in modalità batch, ma è allo studio una versione interattiva.

La topologia del sistema, nonché tutte le altre informazioni relative alle azioni dinamiche e cinematiche imposte vengono trasferite al programma utilizzando matrici.

Il codice è completato dal programma `ViR3D` di post-processing grafico, appositamente scritto in Visual Basic, che consente, dopo aver letto i numerosi files generati nell'esecuzione, di visualizzare il moto dei corpi.

## 2. FORMULAZIONE DINAMICA

La formulazione dinamica implementata in `SymDyn3D` è di tipo lagrangiano e si esplicita per *default* nel seguente sistema DAE di indice 3

$$[M] \{\ddot{q}\} + [\Psi_q]^T \{\lambda\} = \{Q\} , \quad (1a)$$

$$\{\Psi(q,t)\} = \{0\} , \quad (1b)$$

in cui  $[M]$  è la matrice delle masse,  $\{q\}$  il vettore delle coordinate generalizzate,  $\{Q\}$  il vettore delle forze generalizzate,  $\{\lambda\}$  il vettore dei moltiplicatori di Lagrange,  $\{\Psi\}$  il vettore dei vincoli.

Per quanto concerne la generazione delle equazioni scalari dei vincoli dovuti alle coppie cinematiche, `SymDyn3D` si avvale dell'approccio modulare proposto da E.J. Haug [12, 16] e di talune procedure MapleV, già sviluppate da uno degli autori, per l'analisi cinematica di meccanismi spaziali [11]. Com'è noto, tale metodologia si avvale di un set ridondante di coordinate e conduce a matrici sparse di ordine elevato. L'esperienza dimostra che operando con un set minimo di coordinate si perviene a matrici di ordine ridotto, ma dense e con elementi ad elevata complessità algebrica.

Tra le peculiarità delle metodologie impiegate ricordiamo:

- la facilità di implementazione;
- la possibilità di interfacciare immediatamente i risultati della simulazione con un postprocessing grafico.

Si osserva che se i vincoli del meccanismo sono sovrabbondanti, il vettore dei moltiplicatori di Lagrange, talvolta inizialmente richiesto da talune procedure numeriche di integrazione, non può essere univocamente definito. In tal caso, appare conveniente ricorrere a procedure di ortogonalizzazione dei vincoli nel corso dell'integrazione.

Rispetto all'approccio proposto da E.J. Haug [12], fondato sulla partizione delle coordinate e sulla successiva integrazione del sistema di equazioni differenziali con il metodo di Adams-Bashfort, si registrano sostanziali differenze.

In particolare, si preferisce adottare per le eccellenti caratteristiche di stabilità ed accuratezza numerica il metodo implicito di Hairer e Wanner [15]. Ciò implica un'accurata stima dei valori iniziali imposti alle posizioni, alle velocità, alle accelerazioni ed ai moltiplicatori di Lagrange.

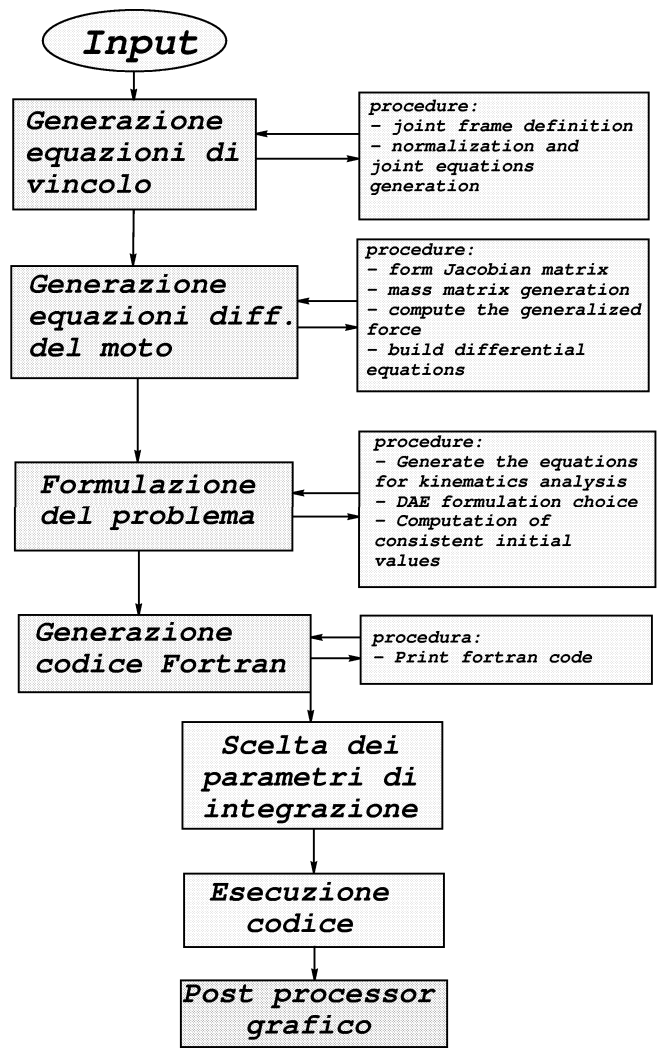


Figura 1: Struttura del programma SymDyn3D

### 3. STRUTTURA DEL PROGRAMMA

Il programma si articola in diverse sezioni secondo lo schema generale presentato in Figura 1. La struttura modulare facilita la leggibilità del programma e ne consente ulteriori ampliamenti.

#### 3.1 Sezione di input

Con riferimento allo schema di Figura 2, la sezione che sovrintende all'input si articola nelle seguenti fasi:

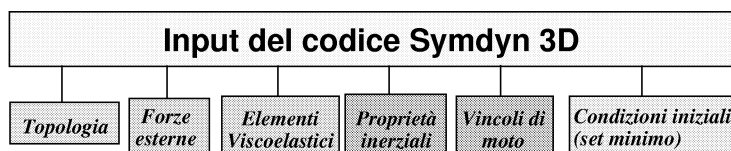


Figura 2: Fasi della procedura di inserimento dei dati

#### 1. Topologia

Vengono innanzitutto definiti il numero dei corpi ( $NB$ ), quello delle coppie cinematiche.

La specifica delle connessioni tra i corpi si realizza attraverso la matrice  $CON$ .

Tale matrice possiede tante righe quante sono le coppie cinematiche presenti e la  $k^{ma}$  riga, relativa al collegamento tra i membri  $i$  e  $j$ , possiede la seguente struttura:

$$\left[ \begin{array}{c} i, j, \text{ tipo di vincolo,} \\ P_{ik}^x, P_{jk}^y, P_{ik}^z, Q_{ik}^x, Q_{jk}^y, Q_{ik}^z, R_{ik}^x, R_{jk}^y, R_{ik}^z, \\ P_{jk}^x, P_{jk}^y, P_{jk}^z, Q_{jk}^x, Q_{jk}^y, Q_{jk}^z, R_{jk}^x, R_{jk}^y, R_{jk}^z \end{array} \right] \quad (2)$$

ove le coordinate dei punti  $P_{mk}$ ,  $Q_{mk}$  ed  $R_{mk}$   $m = i, j$  individuano il riferimento del giunto su ciascun elemento cinematico della coppia, così come specificato in Figura 3.

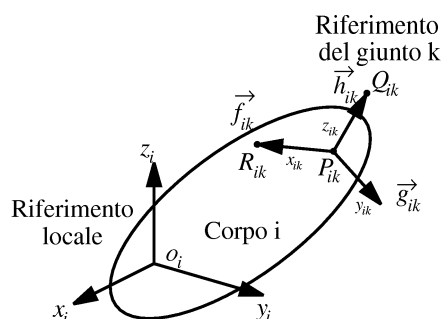


Figura 3: Definizione del riferimento del giunto: Nomenclatura

#### 2. Forze esterne

Indicato con  $NFORCES$  il numero totale di forze e/o coppie esterne agenti sul sistema, viene formata la matrice  $FORCE$ , di dimensione  $NFORCES \times 11$ , la cui generica riga, tramite la seguente sintassi, contiene tutti i dati relativi alla forza esterna:

$$\left[ \begin{array}{c} link, Xforce, Yforce, Zforce, \\ Xtorque, Ytorque, Ztorque, x, y, z, type \end{array} \right] \quad (3)$$

dove  $link$  è il numero identificativo del membro su cui agisce la forza o coppia,  $Xforce$ ,  $Yforce$ ,  $Zforce$  sono le componenti della forza lungo i tre assi coordinati,  $Xtorque$ ,  $Ytorque$ ,  $Ztorque$  le componenti della coppia lungo i tre assi coordinati,  $x$ ,  $y$ ,  $z$  il punto di applicazione nel riferimento locale del corpo,  $type$  specifica se le componenti della forza o coppia sono espresse nel riferimento locale ( $type = Loc$ ) o inerziale ( $type = Glob$ ).

### 3. Elementi viscoelastici

Le caratteristiche relative agli elementi molla-smorzatore viscoso, sono specificate tramite gli elementi della matrice *SPDAMPERS*, la cui generica riga ha la sintassi:

$$[i, j, force, x_1, y_1, z_1, x_2, y_2, z_2, T] \quad (4)$$

nel caso si tratti di un dispositivo traslazionale, e

$$[i, j, torque, 0, 0, 0, 0, 0, 0, R] \quad (5)$$

nel caso di dispositivo rotazionale.

In particolare, *i* e *j* distinguono i due corpi collegati, *force* e *torque* sono le espressioni matematiche della forza e della coppia rispettivamente,  $x_1, y_1, z_1, x_2, y_2, z_2$  i punti di ancoraggio del dispositivo traslazionale sui due corpi, *T* ed *R* distinguono, rispettivamente, la natura traslazionale o rotazionale del dispositivo. Nelle espressioni di *force* e *torque* le distanze (o gli angoli) e le derivate delle distanze (o le derivate degli angoli) vengono indicate con *d* e *dd*, e si provvederà in seguito a sostituire ad esse le espressioni corrette.

### 4. Proprietà inerziali

Massa  $m_i$  e tensore d'inerzia  $[J_i]$  ( $i = 1, 2, \dots, NB$ ), quest'ultimo valutato nel riferimento locale al corpo  $i^{mo}$ , vengono specificati mediante apposite matrici cui il programma di generazione delle equazioni fa riferimento nel corso dell'esecuzione.

### 5. Vincoli di moto

Tali vincoli sono esplicitamente espressi attraverso equazioni che figurano quali elementi del vettore *DRIV*.

### 6. Condizioni iniziali

Com'è noto, l'integrazione dei sistemi DAE richiede l'assegnazione di condizioni di moto iniziali congruenti con tutte le equazioni di vincolo imposte. D'altro canto, l'utente non è in grado di fornire tali informazioni. Per risolvere il problema, si è deciso di richiedere all'utente stesso di specificare il set minimo di condizioni iniziali nei vettori  $q_{in}$  e  $qp_{in}$ . Successivamente, nel programma Fortran principale generato da *SymDyn3D*, si provvederà al calcolo delle rimanenti condizioni iniziali. In particolare, ci si avvale delle procedure messe a disposizione dal package *MINPACK* per la soluzione dei sistemi di equazioni algebrici non lineari.

E' allo studio un'interfaccia grafica interattiva di supporto all'utente nella fase di input dei dati.

## 3.2 Equazioni di vincolo

*SymDyn3D* distingue tra vincoli scleronomi e reonomi. In particolare, le seguenti coppie cinematiche possono essere incluse nel modello:

Rotoidale	Prismatica	Cilindrica
Sferica	Sfera nel cilindro	Giunto di Hooke

Se necessario, altre coppie possono essere create appositamente. Ad esempio, è stata creata una coppia *elbow* che tiene conto dell'angolo tra gli assi dei movimenti consentiti (prono-supinazione e flessione) all'avambraccio nel suo moto relativo al braccio [14].

Per ciascuna coppia vengono generate tante equazioni scalari quanti sono i gradi di vincolo introdotti dalle coppie medesime.

Le suddette equazioni traducono relazioni elementari di vincolo quali quelle di parallelismo (*par1* e *par2*), ortogonalità (*ort1*, *ort2*) tra vettori e di coincidenza tra punti (*sf*) (v. Tabelle 1 e 2).

I vincoli reonomi, se presenti, andranno esplicitamente indicati in un apposito vettore. Una versione di *SymDyn3D* è stata modificata per consentire di imporre leggi di moto costituite da funzioni discrete di valori.

Poiché l'orientamento dei corpi viene specificato utilizzando i parametri di Eulero, il sistema  $\{\Psi\} = \{0\}$  comprende anche le condizioni di normalizzazione tra i parametri

$$q_{7i-3}^2 + q_{7i-2}^2 + q_{7i-1}^2 + q_{7i}^2 = 1 \quad (i = 1, 2, \dots, NB) \quad (6)$$

Tabella 1: Numero di equazioni scalari associate a ciascun vincolo base

Vincolo di base	Num. eq.
Primo vincolo di ortogonalità (ort-1)	1
Secondo vincolo di ortogonalità (ort-2)	1
Primo vincolo di parallelismo (par-1)	2
Secondo vincolo di parallelismo (par-2)	2
Sferico (sf)	3
Sferico-sferico	3

Tabella 2: Coppie cinematiche e vincoli di base

Coppia cinematica	Vincoli di base			
	ort-1	par-1	par-2	sf
Rotoidale (R)	-	X	-	X
Prismatica (P)	X	X	X	-
Cilindrica (C)	-	X	X	-
Sfera-Cil. (G)	-	-	X	-
Sferica (S)	-	-	-	X

Questa sezione del programma produce, infine, la matrice Jacobiana dei vincoli  $[\Psi_q]$  ed i vettori  $\Psi_t$  e  $\gamma$  per la definizione delle velocità e delle accelerazioni attraverso la soluzione dei sistemi:

$$[\Psi_q] \{\dot{q}\} = -\{\Psi_t\} \quad (7)$$

$$[\Psi_q] \{\ddot{q}\} = \{\gamma\} \quad (8)$$

### 3.3 Equazioni differenziali del moto

Questa parte del codice sovrintende alla generazione della matrice  $M$  delle masse e del vettore delle forze esterne generalizzate  $Q$  ed all'assemblaggio di tali array secondo la (1a).

In particolare, il segmento di codice MapleV (R.6) che è adibito alla generazione della matrice globale delle masse a partire dalle sottomatrici delle masse di ciascun corpo è di seguito riportato:

*Assemble the 7x7 mass matrix of the i-th body*

mas\_i:  $3 \times 3$  diagonal mass matrix for the i-th body

J\_i: Local inertia tensor ( $3 \times 3$ ) for the i-th body

e0, e1, e2, e3, e4: Euler's parameters

mass\_i: full mass matrix ( $7 \times 7$ ) of the i-th body

```
> massa:=proc(mas_i,J_i,e0,e1,e2,e3) global m_i:
> local G,zero34,zero43,m,i,j,H;
> G:=matrix([[ -e1,e0,e3,-e2],[ -e2,-e3,e0,e1],[ -e3,e2,-e1,e0]]);
> H:=matrix(4,3,0):H:=evalm(transpose(G)&*J_i&*G);
> for i from 1 to 4 do for j from 1 to 4 do H[i,j]:=4*H[i,j]:od:od:
> zero34:=matrix(3,4,0):zero43:=matrix(4,3,0):m_i:=matrix(7,7,0):
> m_i:=blockmatrix(2,2,[mas_i,zero34,zero43,H]):end:
```

*NB: Number of bodies Assemble the global mass matrix [M] ( $7*NB \times 7*NB$ ):*

```
> MASS:=matrix(7*NB,7*NB,0):
> for i from 1 to NB do
> M||i:=matrix(7,7,0):
> m||i:=massa(mas||i,J||i,q[7*i-3],q[7*i-2],q[7*i-1],q[7*i]):
> M||i:=copy(m||i):
> j:=7*(i-1)+1:
> linalg[copyinto](M||i,MASS,j,j):
> od:
```

Per quanto concerne le forze generalizzate<sup>1</sup> associate agli spostamenti virtuali del generico corpo  $i^{mo}$ , introdotta la matrice di trasformazione dal riferimento locale  $o_i - x_i y_i z_i$  a quello inerziale

$$[A]_i^0 = [E]_i^0 [G^T]_i^0 \quad (9)$$

ed il vettore  $\{s\}^{(i)}$  che individua le coordinate cartesiane del punto di applicazione della forza distinguiamo i seguenti casi:

1. Forza  $\{f_i\}^{(0)}$  e coppia  $\{\tau_i\}^{(0)}$  esterne con componenti associate nel riferimento inerziale  $O - XYZ$

$$\{Q_i\}_{7 \times 1} = \left\{ \begin{array}{c} \{f_i\}^{(0)} \\ 2 [E^T]_i^0 \left[ \begin{array}{c} s_i^{(0)} \\ \{f_i\}^{(0)} \end{array} \right] + 2 [E^T]_i^0 \{\tau_i\}^{(0)} \end{array} \right\} \quad (10)$$

2. Forza  $\{f_i\}^{(i)}$  e coppia  $\{\tau_i\}^{(i)}$  esterne con componenti associate nel riferimento locale

$$\{Q_i\} = \left\{ \begin{array}{c} [A]_i^0 \{f_i\}^{(i)} \\ 2 [G^T]_i^0 \left[ \begin{array}{c} s_i^{(i)} \\ \{f_i\}^{(i)} \end{array} \right] + 2 [G^T]_i^0 \{\tau_i\}^{(i)} \end{array} \right\} \quad (11)$$

3. Azioni dinamiche esercitate da un elemento molla-smorzatore viscoso traslazionale

Se  $\{d_{ij}\}$  è il vettore, di modulo  $\ell$ , che congiunge i punti  $P_i$  e  $P_j$  di attacco, rispettivamente sul corpo  $i$ -esimo e  $j$ -esimo, i vettori forze generalizzate per i corpi  $i$  e  $j$  saranno:

$$\{Q_i\} = \frac{\mathcal{F}}{\ell} \left\{ \begin{array}{c} \{d_{ij}\}^{(0)} \\ 2 [G^T]_i^0 \left[ \begin{array}{c} s_{P_i}^{(i)} \\ \{d_{ij}\}^{(0)} \end{array} \right] [A^T]_i^0 \{d_{ij}\}^{(0)} \end{array} \right\} \quad (12)$$

$$\{Q_j\} = -\frac{\mathcal{F}}{\ell} \left\{ \begin{array}{c} \{d_{ij}\}^{(0)} \\ 2 [G^T]_j^0 \left[ \begin{array}{c} s_{P_j}^{(j)} \\ \{d_{ij}\}^{(0)} \end{array} \right] [A^T]_j^0 \{d_{ij}\}^{(0)} \end{array} \right\} \quad (13)$$

dove:

$$\mathcal{F} = k(\ell - \ell_0) + c\dot{\ell} + \mathcal{F}_a(\ell, \dot{\ell}, t) \quad (14)$$

è la forza generata dal dispositivo,  $\mathcal{F}_a$  la forza generata da un'eventuale attuatore posto in serie ad esso,  $\{s_{P_i}\}^{(i)}$  il vettore  $o_i \vec{P}_i$ ,  $k$  e  $c$  i coefficienti di rigidità e smorzamento viscoso.

4. Azioni dinamiche esercitate da un elemento molla-smorzatore viscoso rotazionale

Se la rotazione relativa tra i membri  $i$  e  $j$  avviene attorno ad un asse parallelo ai versori  $\{h_i\}$  ed  $\{h_j\}$  tra loro sovrapposti, i vettori delle forze generalizzate agenti sui corpi suddetti saranno, rispettivamente:

$$\{Q_i\} = \mathcal{M} \left\{ \begin{array}{c} \{0\}_{3 \times 1} \\ 2 [G^T]_i^0 \{h_i\}^{(i)} \end{array} \right\} \quad (15)$$

$$\{Q_j\} = -\mathcal{M} \left\{ \begin{array}{c} \{0\}_{3 \times 1} \\ 2 [G^T]_j^0 [A^T]_j^0 [A]_i^0 \{h_i\}^{(i)} \end{array} \right\} \quad (16)$$

dove:

$$\mathcal{M} = k_t(\vartheta - \vartheta_0) + c_t \dot{\vartheta} + \mathcal{M}_a(\vartheta, \dot{\vartheta}, t) \quad (17)$$

è il modulo della coppia generata dal dispositivo,  $k_t$  e  $c_t$  i coefficienti di rigidità e smorzamento viscoso torsionale,  $\mathcal{M}_a$  il modulo della coppia generata da un'eventuale attuatore posto in serie ad esso;  $\vartheta$  è l'angolo di rotazione relativa, misurabile ad es. tra gli assi  $\{f_i\}$  ed  $\{f_j\}$ , dato dalla formula:

$$\vartheta = \text{ATAN2} \left( f_j^{(ik)} [2], f_j^{(ik)} [1] \right) \quad (18)$$

<sup>1</sup>Nel prosieguo con la notazione  $\{\cdot\}^{(i)}$   $\{\cdot\}^{(0)}$  si individua il medesimo vettore, ma con componenti espresse nel riferimento cartesiano locale del corpo  $i^{mo}$  e, rispettivamente, in quello inerziale.

ove  $\{f_j\}^{(i_k)}$ , vettore  $\{f_j\}$  con le componenti espresse nel riferimento del giunto  $k$  del membro  $i$ , è espresso da:

$$\{f_j\}^{(i_k)} = [C]_i^{i_k} [A^T]_i^0 [A]_j^0 \{f_j\}^{(0)} \quad (19)$$

e  $[C]_i^{i_k}$  la matrice di trasformazione dal riferimento  $o_i - x_i y_i z_i$  a quello del giunto  $k$ , sempre solidale ad  $i$ , data da

$$[C]_i^{i_k} = [ \{f_i\}^i \quad \{g_i\}^i \quad \{h_i\}^i ] . \quad (20)$$

Il modulo della velocità angolare relativa tra i corpi  $i$  e  $j$  si dimostra essere pari a:

$$\dot{\vartheta} = \left( 2 [A^T]_i^0 [A]_j^0 [G]_j^0 \{\dot{p}_j\}^{(j)} - 2 [G]_i^0 \{\dot{p}_i\}^{(i)} \right)^T \{h_i\}^{(i)} \quad (21)$$

### 3.4 Generazione codice Fortran e postprocessing grafico

Il programma SymDyn3D è stato integrato, al fine di facilitare l'utilizzo delle soluzioni da esso generato, con un modulo per la postprocessing grafico dei risultati. Detto modulo, denominato ViR3D, è stato sviluppato interamente utilizzando Visual Basic a cui sono state collegate le librerie grafiche OpenGL. La logica di programmazione utilizzata è totalmente *object oriented* e compatibile con il modello COM. La struttura definita dalle classi è rappresentata in Figura 4 e rappresenta una soluzione intermedia fra modularità e performance, elemento quest'ultimo non più critico nelle applicazioni grafiche scientifiche con l'avvento delle nuove schede grafiche a basso costo dotate di GPU OpenGL compatibile. La modu-

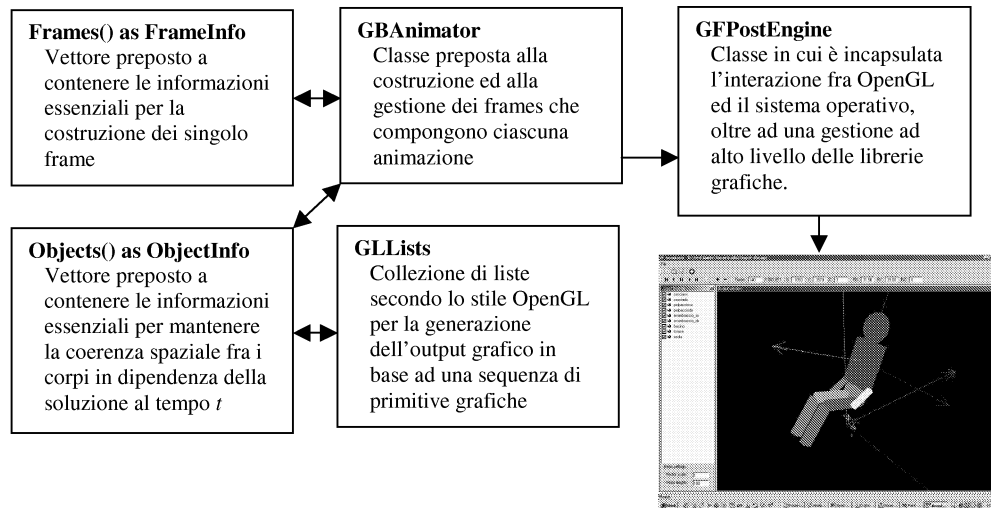


Figura 4: Struttura del programma ViR3D per il post-processing grafico

larità del modello adottato è l'elemento principale su cui si basa lo sviluppo, nella seconda fase, di detto modulo e rivolta all'integrazione di una terza classe, oltre a GBAimator e GFPostEngine, in grado di gestire le fasi di preprocessing del modello 3D completo attraverso un input interattivo e a comando. Tra le caratteristiche del modulo di postprocessing, particolare rilievo hanno quelle che permettono, in tempo reale, di definire dei punti chiave nel modello e eseguire un *tracking* delle posizioni nel tempo; inoltre, è possibile richiedere al modulo di integrare nella soluzione l'evoluzione del vettore velocità istantanea di rotazione e velocità lineare del baricentro di un generico corpo. Detta integrazione avviene, oggi, solamente rappresentando i vettori dotati di un'etichetta sulla punta, con l'indicazione del valore numerico della grandezza rappresentata, durante l'animazione ed è in fase di sviluppo una integrazione in linea che fa uso di grafici 2D sovrapposti.

#### 4. ESEMPIO

L'esempio che viene riportato a titolo di esempio di impiego del codice consiste nell'analisi dinamica di un meccanismo costituito da due pendoli. Il primo di questi, collegato al telaio mediante una coppia rotoidale, ha l'asse di rotazione relativa coincidente con l'asse  $z$  (v. Figura 5), mentre il secondo è collegato al primo tramite una coppia rotoidale avente asse di rotazione relativa nella direzione della dimensione maggiore del primo pendolo. I dati geometrici e inerziali caratteristici di ciascun pendolo (parallelepipedo a base quadrata) sono:

- Lunghezza: 1 m
- Massa: 1 kg
- Momento di inerzia maggiore:  $0.1 \text{ kg}\cdot\text{m}^2$
- Momento di inerzia minore:  $0.01 \text{ kg}\cdot\text{m}^2$

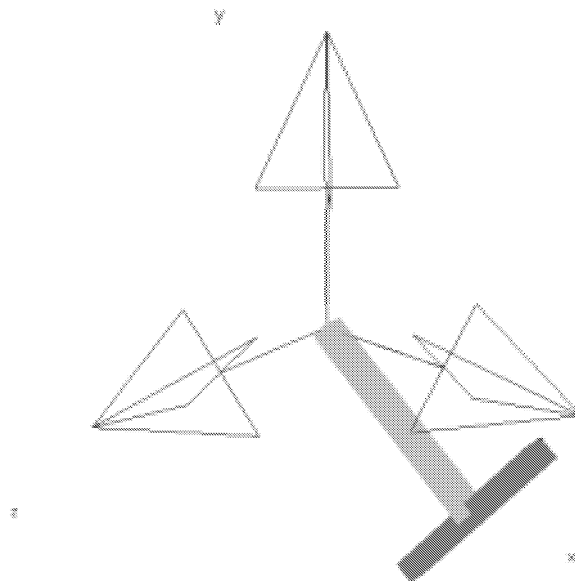


Figura 5: Doppio pendolo: visualizzazione in ViR3D

E' stata adottata una posizione iniziale con il primo pendolo disposto parallelamente all'asse  $x$ . Si è scelto di simulare il movimento del sistema per 5 s (*i.e.* 500 step di output). Alcuni dei risultati ottenuti sono riassunti nelle Figure 6 e 7. In particolare, nella prima è stata riportata la traiettoria spaziale del baricentro del primo pendolo, mentre nella seconda l'andamento della coordinata assoluta  $z$  per entrambi i baricentri dei corpi.

#### Riferimenti bibliografici

- [1] Pennestri, E., Belfiore, N.P., *An Application of Computer Algebra to Multibody Dynamics*, Proc. Fifth National Applied Mechanisms and Robotics Conference, Cincinnati (OH), October 12-15, 1997, vol.2, Paper AMR97-058.
- [2] Pennestri, E., *SymDyn- A Maple Program for Multibody Dynamics Simulation*, MapleTech, vol.5, No.1, 1998, pp.28-32.
- [3] Kane, T.R., Levinson, D.A., *Dynamics, Theory and Applications*, McGraw-Hill Series in Mechanical Engineering, 1985, McGraw-Hill Book Company.
- [4] Schielen, W.O., Kreuzer, E.J., *Symbolic Computerized Derivation of Equations of Motion*, *Dynamics of Multibody Systems*, IUTAM, Munich, Springer-Verlag, 1977.
- [5] Wittenburg, J., Wolz, U., *MESA VERDE: A Symbolic Program for Nonlinear Articulated-Rigid-Body Dynamics*, *Proceedings of the 10th Design Engineering Conference on Mechanical Vibration and Noise*, Cincinnati, 1985.
- [6] Sayers, M.W., *A Symbolic Computer Language for Multibody Systems*, *Journal of Guidance, Control, and Dynamics*, vol.14, No.6, Nov/dec.1991, pp.1153,1163.

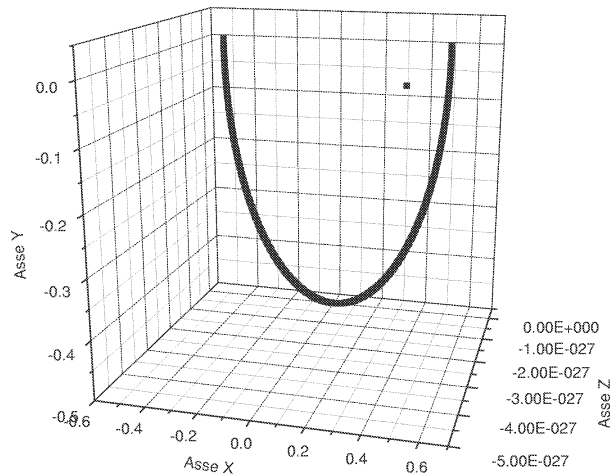


Figura 6: Traiettoria baricentro primo pendolo

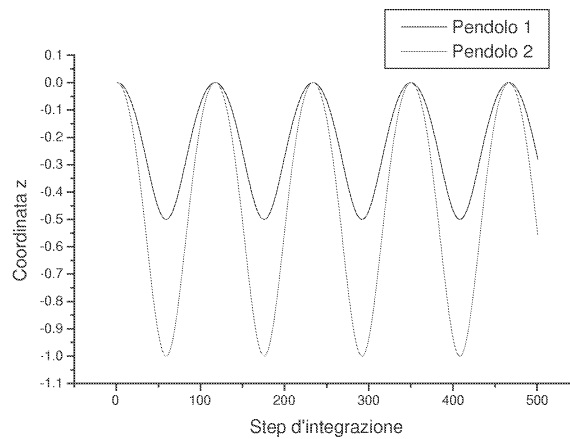


Figura 7: Coordinate assolute z dei baricentri

- [7] Hollars, M.G., Rosenthal, D.E., Sherman, M.A., *SD/FAST User's Manual*, Symbolic Dynamics, Inc., Version B.2, September 1996.
- [8] Mingardi, S., Pennestrì, E., A Short Review of Numerical Methods and Software Tools for Solving Differential-Algebraic Systems of Equations, *Proc. Sixth Applied Mechanisms and Robotics Conference*, December 12-15 1999, Cincinnati, OH, Paper No. AMR99-053.
- [9] Campanile, P., *et al.*, Implementazione mediante tecniche multibody di un modello per la simulazione delle dinamiche vibrazionali del sistema uomo-sedile, *AIMETA, XV Congresso Nazionale*, 2001, Taormina.
- [10] Unda, J., Garcia de Jalon, J., Losantos, F., Enparantza, R., A Comparative Study on Some Different Formulations of the Dynamic Equations of Constrained Mechanical Systems, *ASME 86-DET-138*.
- [11] Brutti, C., Pennestrì, E., Urbinati, F., *Kinematic Analysis of Spatial Mechanisms by Means of Maple*, MapleTech, vol.5, No.2 & 3, pp.49-57.
- [12] Haugh, E.,J., *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, vol.I, Allyn and Bacon, 1989, pp.48-104.
- [13] Kecskeméthy, A., Krupp, T., Hiller, M., Symbolic Processing of Multiloop Mechanism Dynamics Using Closed-Form Kinematics Solutions, *Multibody System Dynamics*, vol.1, 1997, pp.23-45.
- [14] Pennestrì, E., Renzi, A., Santonocito, P., *Dynamic modeling of the human arm with video-based experimental analysis*, In corso di pubblicazione su *Multibody System Dynamics*.
- [15] Hairer, E., Wanner, G., *Solving Ordinary and Differential Equations II, Stiff and Differential-Algebraic Problems*, Springer Verlag, 1991.
- [16] Pennestrì, E., *Dinamica Tecnica e Computazionale*, vol.II, Casa Editrice Ambrosiana, Milano, 2001, in corso di pubblicazione