3D Kinematics and Inverse Dynamics
Matlab Toolbox

User Guide

Version 7 of July 2016

Author

Raphaël Dumas
Univ Lyon, Université Claude Bernard Lyon 1, IFSTTAR, UMR_T9406, LBMC, F69622, Lyon, France
raphael.dumas@ifsttar.fr

Description

This toolbox contains all the Matlab functions for 3D kinematics and inverse dynamics computation applied to the lower and upper limb. The right lower (upper) limb is considered for sign conventions.

For instance, this toolbox includes functions for multibody optimisation with different joint models (noted N, S, U, and H for no model, spherical, universal and hinge models) and functions for inverse dynamics with different methods: vector and Euler angle (noted VE), homogenous matrix (noted HM), wrench and quaternion (noted WQ), generalized (i.e., natural) coordinate (noted GC).
The computations are performed avoiding loops on the sampled instants of time and, thus, require a set of custom-made functions (*_array3.m) for matrix/vector basic operations.

This toolbox also includes an example dataset (*.mat files to be loaded) for gait and manual wheelchair propulsion) and a series of demonstrations (Main_Question_*.m).

Licence

The toolbox is available at www.mathworks.com/matlabcentral/fileexchange

Toolbox content (alphabetic order)

Derive_array3.m	Computation of time derivative of matrix or vector
Dynamics_HM.m	Recursive computations of joint moments and forces by homogenous matrix method
Dynamics_VE.m	Recursive computations of joint moments and forces by vector method
Dynamics_WQ.m	Recursive computations of joint moments and forces by wrench method
Extend_Segment_Fields.m	Computation of all possible segment parameters from natural coordinates
Inverse_Dynamics_GC.m	Computation of joint moments and forces by generalized coordinates (i.e., natural coordinates) method
Inverse_Dynamics_HM.m	Computation of joint moments and forces by homogenous matrix method
Inverse_Dynamics_VE.m	Computation of joint moments and forces by vector and Euler angles method
Inverse_Dynamics_WQ.m	Computation of joint moments and forces by wrench and quaternion method
Joint_Kinematics.m	Computation of joint angles and displacements
Mfilt_array3.m	Filtering of matrix elements
Minv_array3.m	Computation of matrix inverse
Mprod_array3.m	Computation of matrix product
Multibody_Optimisation_NNN.m	Computation of Q by minimisation under constraints (the number of DoFs are 6 at the ankle/wrist, knee/elbow and hip/shoulder)
Multibody_Optimisation_SSS.m	Computation of Q by minimisation under constraints (the number of DoFs are 3 at the ankle/wrist, knee/elbow and hip/shoulder)
Multibody_Optimisation_UHS.m	Computation of Q by minimisation under constraints (the number of DoFs are 2 at the ankle, 1 at the knee and 3 at the hip)
Multibody_Optimisation_UUS.m	Computation of Q by minimisation under constraints (the number of DoFs are 2 at the wrist, 2 at the elbow and 3 at the shoulder)
q2R_array3.m	Computation of rotation matrix from quaternion
Q2Tuv_array3.m	Computation of homogenous matrix from natural coordinates (with X = u and Z = (u×v)/||u×v||)
Q2Tuw_array3.m	Computation of homogenous matrix from natural coordinates (with X = u and Y = (w×u)/||w×u||)
Q2Twu_array3.m	Computation of homogenous matrix from natural coordinates (with Z = w and Y = (w×u)/||w×u||)
qinv_array3.m	Computation of quaternion inverse
qlog_array3.m	Computation of quaternion logarithm
qprod_array3.m	Computation of quaternion product
R2fixedZYX_array3.m	Computation of Euler angles from rotation matrix (with ZYX fixed (i.e., space) sequence for segment kinematics)
R2mobileXZY_array3.m	Computation of Euler angles from rotation matrix (with XZY mobile (i.e., body) sequence for segment kinematics)
R2mobileZXY_array3.m	Computation of Euler angles from rotation matrix (with ZXY mobile (i.e., body) sequence for joint kinematics)
R2mobileZYX_array3.m	Computation of Euler angles from rotation matrix (with ZYX mobile (i.e., body) sequence for joint kinematics)
R2q_array3.m	Computation of quaternion from rotation matrix
SARA_array3.m	Functional method for axis of rotation (AoR)
SCoRE_array3.m	Functional method for centre of rotation (CoR)
Segment_Kinematics_HM.m	Computation of segment kinematics by homogenous matrix method
Segment_Kinematics_VE.m	Computation of segment kinematics by Euler angles method
Segment_Kinematics_WQ.m	Computation of segment kinematics by quaternion method
Tinv_array3.m	Computation of homogenous matrix inverse
Vexp_array3.m	Computation of vector exponential
Vfilt_array3.m	Filtering of vector
Vnop_array3.m	Non-orthogonal projection on three basis vectors
Vnorm_array3.m	Computation of vector norm
Vskew_array3.m	Computation of skew matrix of vector

After download, the toolbox must be unzipped in a directory to be specified in the path.

Example datasets and demonstrations

The datasets are provided as two files (Segment.mat, Joint.mat) following the data format presented pages 45-51.

Lower limb – gait

The subject (male, 50 y.o., 1.85 m, 90 kg, asymptomatic) performed one gait cycle on level ground at comfortable speed. He signed an inform consent and the experimental procedures were approved by the local ethic's committee.

Trajectories of skin markers were recorded with a 12-cameras Motion Analysis System (Santa Rosa, CA, USA) at 100 Hz (filtered using a 4th-order Butterworth with a cut-off frequency of 6 Hz). Ground reaction forces were recorded using an AMTI forceplate (Watertown, MA, USA) at 100 Hz (filtered using a 2nd-order Butterworth with a cut-off frequency of 20 Hz).

Upper limb – manual wheelchair propulsion

The subject (male, 39 y.o., 1.80 m, 82 kg, spinal cord injury at T5 level) performed one cycle of manual wheelchair propulsion on level ground at comfortable speed. He signed an inform consent and the experimental procedures were approved by the local ethic's committee.

Trajectories of skin markers were recorded with a 12-cameras Motion Analysis System (Santa Rosa, CA, USA) at 100 Hz (filtered using a 4th-order Butterworth with a cut-off frequency of 5 Hz). Propulsion kinetics were recorded using an instrumented wheel (TSR, Merignac, France) at 500 Hz (filtered using a 2nd-order Butterworth with a cut-off frequency of 20 Hz and resampled at 100 Hz).

Demonstrations

The demonstrations for the 3D kinematics and inverse dynamics Matlab toolbox potential applications are the following main programs:

Main_Question_0.m	Which data format?
Main_Question_1.m	What is the impact of SCS and JCS axes on joint angles and displacements?
Main_Question_2.m	What is the impact of multibody optimisation on joint angles and displacements?
Main_Question_3.m	What is the impact of the inverse dynamic method on joint moments and forces?
Main_Question_4.m	What is the impact of joint forces and moment expression on joint moments and forces?
Main_Question_5.m	What is the impact body segment inertial parameters on joint moments and forces?
Main_Question_6.m	What is the impact of multibody optimisation on joint moments and forces?

Derive_array3.m

Purpose:
Computation of time derivative of matrix or vector

Synopsis:
dVdt = Derive_array3(V,dt)
dMdt = Derive_array3(M,dt)
Inputs: V (i.e., vector) or M (i.e., matrix), dt (i.e., sampling time that is to say dt = 1/f)
Outputs: dVdt or dMdt (i.e., vector or matrix)

Description:
Gradient approximation in the 3rd dimension of matrix or vector (i.e., all sampled instants of time, *cf.* data structure)

See also:
Segment_Kinematics_VE.m
Segment_Kinematics_HM.m
Segment_Kinematics_WQ.m
Inverse_Dynamics_GC.m
Mfilt_array3.m
Vfilt_array3.m

Dynamics_HM.m

Purpose:
Recursive computations of joint moments and forces by homogenous matrix method

Synopsis:
[Joint,Segment] = Dynamics_HM(Joint,Segment,n)
Inputs: Joint, Segment (*cf.* data structure), n (i.e., number of sampled instants of time)
Outputs: Joint, Segment (*cf.* data structure)

Description:
Computation of joint force and moment (**F**, **M**) at proximal endpoint of segment ($\mathbf{r}_P$) expressed in ICS, as a function of:

- homogenous matrix of pseudo-inertia (**J**)
- homogenous matrix of acceleration (**H**)
- homogenous matrix of ground reaction force and moment ($\mathbf{\Phi}_1$)

Theoretical background:

$$\mathbf{\Phi}_i = \mathbf{\Phi}_{i-1} + \left[\mathbf{H}_i - \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{g} \\ 000 & 0 \end{bmatrix}\right].\mathbf{J}_i - \mathbf{J}_i \cdot \left[\mathbf{H}_i - \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{g} \\ 000 & 0 \end{bmatrix}\right]^T$$

References:
N Doriot, L Cheze. A 3D Kinematic and dynamic study of the lower limb during the stance phase of gait using an homogeneous matrix approach. IEEE Transactions on Biomedical Engineering 2004;51(1):21-7
G Legnani, F Casolo, P Righettini, B Zappa, A homogeneous matrix approach to 3D kinematics and dynamics - I. Theory. Mechanisms and Machine Theory 1996; 31(5): 573–87
R Dumas, E Nicol, L Cheze. Influence of the 3D inverse dynamic method on the joint forces and moments during gait. Journal of Biomechanical Engineering 2007; 129(5): 786-90

See also:
Inverse_Dynamics_HM.m
Segment_Kinematics_HM.m

Dynamics_VE.m

Purpose:
Recursive computations of joint moments and forces by vector method

Synopsis:
[Joint,Segment] = Dynamics_VE(Joint,Segment,n)
Inputs: Joint, Segment (*cf.* data structure), n (i.e., number of sampled instants of time)
Outputs: Joint, Segment (*cf.* data structure)

Description:
Computation of joint force and moment (**F**, **M**) at proximal endpoint of segment ($\mathbf{r}_P$) expressed in ICS, as a function of:

- segment mass (m)
- position of centre of mass ($\mathbf{r}_C^s$) expressed in SCS
- inertia tensor at centre of mass ($\mathbf{I}^s$) expressed in SCS
- segment angular velocity and acceleration ($\boldsymbol{\omega}$, $\boldsymbol{\alpha}$) and linear acceleration of centre of mass (**a**) expressed in ICS
- position of proximal endpoint ($\mathbf{r}_P$) expressed in ICS
- position of centre of pressure ($\mathbf{r}_{P1}$) expressed in ICS
- ground reaction force and moment ($\mathbf{F}_1$, $\mathbf{M}_1$) expressed in ICS

Theoretical background:

$$\mathbf{F}_i = m_i\mathbf{a}_i - m_i\mathbf{g} - \mathbf{F}_{i-1}$$

$$\mathbf{M}_i^s = \mathbf{I}_i^s \cdot \alpha_i^s + \omega_i^s \times \left(\mathbf{I}_i^s \cdot \omega_i^s\right) - \mathbf{M}_{i-1}^s - \mathbf{r}_{C_i}^s \times \mathbf{F}_i^s - \left(\mathbf{r}_{P_{i-1}}^s - \mathbf{r}_{P_i}^s - \mathbf{r}_{C_i}^s\right) \times \mathbf{F}_{i-1}^s$$

References:

MP Kadaba, HK Ramakrishnan, ME Wootten, J Gainey, G Gorton, GV Cochran. Repeatability of kinematic, kinetic, and electromyographic data in normal adult gait. Journal of Orthopaedic Research 1989;7(6):849-60
RB Davis, S Ounpuu, D Tyburski, JR Gage. A gait analysis data collection and reduction technique. Human Movement Science 1991; 10: 575-87
CL Vaughan, BL Davis, JC O'Connor. Dynamics of human gait (2d edition). Human Kinetics, Champaign, Illinois, 1999
R Dumas, E Nicol, L Cheze. Influence of the 3D inverse dynamic method on the joint forces and moments during gait. Journal of Biomechanical Engineering 2007; 129(5): 786-90

See also:
Inverse_Dynamics_VE.m
Segment_Kinematics_VE.m

Dynamics_WQ.m

Purpose:
Recursive computation of joint moments and forces by wrench method

Synopsis:
[Joint,Segment] = Dynamics_WQ(Joint,Segment,n)
Inputs: Joint, Segment (*cf.* data structure), n (i.e., number of sampled instants of time)
Outputs: Joint, Segment (*cf.* data structure)

Description:
Computation of joint force and moment (**F**, **M**) at proximal endpoint of segment ($\mathbf{r}_P$) expressed in ICS, as a function of:

- segment mass (m)
- position of centre of mass ($\mathbf{r}_C^s$) expressed in SCS
- inertia tensor at centre of mass ($\mathbf{I}^s$) expressed in SCS
- segment angular velocity and acceleration ($\boldsymbol{\omega}$, $\boldsymbol{\alpha}$) and linear acceleration of centre of mass (**a**) expressed in ICS
- position of proximal endpoint ($\mathbf{r}_P$) expressed in ICS
- position of centre of pressure ($\mathbf{r}_{P1}$) expressed in ICS
- ground reaction force and moment ($\mathbf{F}_1$, $\mathbf{M}_1$) expressed in ICS

Theoretical background:

$$\begin{pmatrix} \mathbf{F}_i \\ \mathbf{M}_i \end{pmatrix} = \begin{bmatrix} m_i \mathbf{E}_{3\times3} & \mathbf{0}_{3\times3} \\ m_i \widetilde{\left(\mathbf{r}_{C_i} - \mathbf{r}_{P_i}\right)} & \mathbf{I}_i \end{bmatrix} \begin{pmatrix} \mathbf{a}_i - \mathbf{g} \\ \boldsymbol{\alpha}_i \end{pmatrix} + \begin{pmatrix} \mathbf{0}_{3\times1} \\ \boldsymbol{\omega}_i \times \left(\mathbf{I}_i \boldsymbol{\omega}_i\right) \end{pmatrix} + \begin{bmatrix} \mathbf{E}_{3\times3} & \mathbf{0}_{3\times3} \\ \widetilde{\left(\mathbf{r}_{P_{i-1}} - \mathbf{r}_{P_i}\right)} & \mathbf{E}_{3\times3} \end{bmatrix} \begin{pmatrix} \mathbf{F}_{i-1} \\ \mathbf{M}_{i-1} \end{pmatrix}$$

with $\begin{pmatrix} 0 \\ \mathbf{r}_{C_i} \end{pmatrix} = \mathbf{q}_i \otimes \begin{pmatrix} 0 \\ \mathbf{r}_{C_i}^s \end{pmatrix} \otimes \overline{\mathbf{q}}_i$ and $\mathbf{I}_i = \mathbf{R}_i \mathbf{I}_i^s \mathbf{R}_i^T$

References:
R Dumas, R Aissaoui, J A de Guise. A 3D generic inverse dynamic method using wrench notation and quaternion algebra. Computer Methods in Biomechanics and Biomedical Engineering 2004;7(3):159-166
R Dumas, E Nicol, L Cheze. Influence of the 3D inverse dynamic method on the joint forces and moments during gait. Journal of Biomechanical Engineering 2007; 129(5): 786-90

See also:
Inverse_Dynamics_WQ.m
Segment_Kinematics_WQ.m
Vskew_array3.m

Extend_Segment_Fields.m

Purpose:
Computation of all possible segment parameters from natural coordinates

Synopsis:
Segment = Extend_Segment_Fields(Segment)
Inputs: Segment (*cf.* data structure)
Outputs: Segment (*cf.* data structure)

Description:
Computation of homogenous matrix (**T**), position of proximal endpoints ($\mathbf{r}_P$), quaternion (**q**), rotation matrix (**R**) and Euler angles (φ, θ, ψ) from natural coordinates (**Q**) (*cf.* data structure)

See also:
Inverse_Dynamics_VE.m
Inverse_Dynamics_HM.m
Inverse_Dynamics_WQ.m
Inverse_Dynamics_GC.m
Mprod_array3.m
Vskew_array3.m
Q2Tuv_array3.m
R2q_array3.m
R2fixedZYX_array3.m
q2R_array3.m

Inverse_Dynamics_GC.m

Purpose:
Computation of joint moments and forces by generalized coordinates (i.e., natural coordinates) method

Synopsis:
[Joint,Segment] = Inverse_Dynamics_GC(Joint,Segment,f,fc,n)
Inputs: Joint, Segment (*cf.* data structure), f (i.e., sampling frequency), fc (i.e., cut off frequency), n (i.e., number of sampled instants of time)
Outputs: Joint, Segment (*cf.* data structure)

Description:
Computation of joint force and moment (**F**, **M**) at proximal endpoint of segment ($\mathbf{r}_P$) expressed in ICS, as a function of:

- generalized mass matrix (**G**)
- second time derivatives of natural coordinates (**Q**)
- generalized ground reaction force and moment (through $\mathbf{N}_1^{P_1}$ and $\mathbf{N}_1^{*}$)

Theoretical background:

$$\begin{pmatrix} \mathbf{F}_i \\ \mathbf{M}_i \\ \boldsymbol{\lambda}_i^r \end{pmatrix} = \left[\left[\mathbf{N}_i^{P_i}\right]^T \quad \left[\mathbf{N}_i^{*}\right]^T \quad -\left[\mathbf{K}_i^{r}\right]^T \right]^{-1} \cdot$$

$$\left([\mathbf{G}_i]\ddot{\mathbf{Q}}_i - \left[\mathbf{N}_i^{C_i}\right]^T m_i \mathbf{g} - \left[\mathbf{N}_i^{P_i}\right]^T (-\mathbf{F}_{i-1}) - \left[\mathbf{N}_i^{*}\right]^T \left(-\mathbf{M}_{i-1} + \left(\mathbf{r}_{P_{i-1}} - \mathbf{r}_{P_i}\right) \times (-\mathbf{F}_{i-1})\right) \right)$$

with $\left[\mathbf{N}_i^{P_i}\right]^T = \begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{E}_{3\times3} \\ \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} \end{bmatrix}$

with $\left[\mathbf{N}_i^{*}\right]^T = \begin{bmatrix} \mathbf{0}_{3\times1} & \left(\mathbf{r}_{P_i} - \mathbf{r}_{D_i}\right) & \mathbf{0}_{3\times1} \\ -\mathbf{u}_i & -\left(\mathbf{r}_{P_i} - \mathbf{r}_{D_i}\right) & -\mathbf{w}_i \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{w}_i \\ \mathbf{u}_i & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} \left[\mathbf{B}_i^{*}\right]^{-1}$

where $\left[\mathbf{B}_i^{*}\right] = \left[\mathbf{w}_i \times \mathbf{u}_i \quad \mathbf{u}_i \times \left(\mathbf{r}_{P_i} - \mathbf{r}_{D_i}\right) \quad -\left(\mathbf{r}_{P_i} - \mathbf{r}_{D_i}\right) \times \mathbf{w}_i \right]$

with $\left[\mathbf{K}_i^{r}\right]^T = \begin{bmatrix} 2\mathbf{u}_i & \left(\mathbf{r}_{P_i} - \mathbf{r}_{D_i}\right) & \mathbf{w}_i & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{u}_i & \mathbf{0}_{3\times1} & 2\left(\mathbf{r}_{P_i} - \mathbf{r}_{D_i}\right) & \mathbf{w}_i & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & -\mathbf{u}_i & \mathbf{0}_{3\times1} & -2\left(\mathbf{r}_{P_i} - \mathbf{r}_{D_i}\right) & -\mathbf{w}_i & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{u}_i & \mathbf{0}_{3\times1} & \left(\mathbf{r}_{P_i} - \mathbf{r}_{D_i}\right) & 2\mathbf{w}_i \end{bmatrix}$

with

$$\mathbf{G}_{i\,(12\times12)} = \begin{bmatrix} (J_i)_{uu}\mathbf{E}_{3\times3} & \left(m_i\left(n_i^{C_i}\right)_u + (J_i)_{uv}\right)\mathbf{E}_{3\times3} & -(J_i)_{uv}\mathbf{E}_{3\times3} & (J_i)_{uw}\mathbf{E}_{3\times3} \\ & \left(m_i + 2m_i\left(n_i^{C_i}\right)_v + (J_i)_{vv}\right)\mathbf{E}_{3\times3} & \left(-m_i\left(n_i^{C_i}\right)_v - (J_i)_{vv}\right)\mathbf{E}_{3\times3} & \left(m_i\left(n_i^{C_i}\right)_w + (J_i)_{vw}\right)\mathbf{E}_{3\times3} \\ & & (J_i)_{vv}\mathbf{E}_{3\times3} & -(J_i)_{vw}\mathbf{E}_{3\times3} \\ \text{symetrical} & & & (J_i)_{ww}\mathbf{E}_{3\times3} \end{bmatrix}$$

where $\mathbf{n}_i^{C_i} = \begin{pmatrix} \left(n_i^{C_i}\right)_u \\ \left(n_i^{C_i}\right)_v \\ \left(n_i^{C_i}\right)_w \end{pmatrix} = \left[\mathbf{B}_i^{\mathbf{u}}\right]^{-1} \cdot \mathbf{r}_{C_i}^s$

and where $\mathbf{J}_i = \left[\mathbf{B}_i^{\mathbf{uv}}\right]^{-1}\left[\mathbf{I}_i^s + m_i\left(\left(\left(\mathbf{r}_{C_i}^s\right)^T \mathbf{r}_{C_i}^s\right)\mathbf{E}_{3\times3} - \mathbf{r}_{C_i}^s\left(\mathbf{r}_{C_i}^s\right)^T\right)\right]\left(\left[\mathbf{B}_i^{\mathbf{uv}}\right]^{-1}\right)^T$

and with $\left[\mathbf{N}_i^{C_i}\right]^T = \begin{bmatrix} \left(n_i^{C_i}\right)_u \mathbf{E}_{3\times3} \\ \left(1 + \left(n_i^{C_i}\right)_v\right)\mathbf{E}_{3\times3} \\ -\left(n_i^{C_i}\right)_v \mathbf{E}_{3\times3} \\ \left(n_i^{C_i}\right)_w \mathbf{E}_{3\times3} \end{bmatrix}$

Alternatively,

$$\begin{pmatrix} \mathbf{F}_i \\ \mathbf{M}_i \end{pmatrix} = \left[\left[\mathbf{Z}_i^r\right]^T\left[\mathbf{N}_i^{P_i}\right]^T \quad \left[\mathbf{Z}_i^r\right]^T\left[\mathbf{N}_i^*\right]^T\right]^{-1} \cdot$$

$$\left[\mathbf{Z}_i^r\right]^T\left([\mathbf{G}_i]\ddot{\mathbf{Q}}_i - \left[\mathbf{N}_i^{C_i}\right]^T m_i\mathbf{g} - \left[\mathbf{N}_i^{P_i}\right]^T(-\mathbf{F}_{i-1}) - \left[\mathbf{N}_i^*\right]^T\left(-\mathbf{M}_{i-1} + \left(\mathbf{r}_{P_{i-1}} - \mathbf{r}_{P_i}\right)\times(-\mathbf{F}_{i-1})\right)\right)$$

with $\mathbf{Z}_i^r$ the matrix composed of the eigenvectors of $\left[\mathbf{K}_i^r\right]^T\left[\mathbf{K}_i^r\right]$ corresponding to the null eigenvalues

References:

R Dumas, L Cheze. 3D inverse dynamics in non-orthonormal segment coordinate system. Medical & Biological Engineering & Computing 2007; 45(3): 315-22
R Dumas, E Nicol, L Cheze. Influence of the 3D inverse dynamic method on the joint forces and moments during gait. Journal of Biomechanical Engineering 2007; 129(5): 786-90
JW Kamman, RL Huston. Constrained multibody system dynamics - an automated approach. Computers & Structures 1984; 18(6): 999–1003

See also:

Derrive_array3.m
Vnorm_array3.m
Minv_array3.m
Mprod_array3.m

Inverse_Dynamics_HM.m

Purpose:
Computation of joint moments and forces by homogenous matrix method

Synopsis:
[Joint,Segment] = Inverse_Dynamics_HM(Joint,Segment,f,fc,n)
Inputs: Joint, Segment (*cf.* data structure), f (i.e., sampling frequency), fc (i.e., cut off frequency), n (i.e., number of sampled instants of time)
Outputs: Joint, Segment (*cf.* data structure)

Description:
Data formatting and call of functions Kinematics_HM.m and Dynamics_HM.m

Theoretical background:

$$\mathbf{\Phi}_1 = \mathbf{T}_1 \cdot \begin{bmatrix} \tilde{\mathbf{M}}_1 & \mathbf{F}_1 \\ -\left[\mathbf{F}_1\right]^T & 0 \end{bmatrix} \cdot \left[\mathbf{T}_1\right]^T$$

$$\mathbf{J}^s = \begin{bmatrix} \dfrac{I_{xx} + I_{yy} + I_{zz}}{2}\mathbf{E}_{3\times 3} - \mathbf{I}^s & m_i \mathbf{r}_C^s \\ m_i \left(\mathbf{r}_C^s\right)^T & m_i \end{bmatrix}$$

References:
N Doriot, L Cheze. A 3D Kinematic and dynamic study of the lower limb during the stance phase of gait using an homogeneous matrix approach. IEEE Transactions on Biomedical Engineering 2004; 51(1): 21-7
G Legnani, F Casolo, P Righettini, B Zappa, A homogeneous matrix approach to 3D kinematics and dynamics - I. Theory. Mechanisms and Machine Theory 1996; 31(5): 573–87

See also:
Extend_Segment_Fields.m
Mprod_array3.m
Vskew_array3.m
Segment_Kinematics_HM.m
Dynamics_HM.m

Inverse_Dynamics_VE.m

Purpose:
Computation of joint moments and forces by vector and Euler angles method

Synopsis:
[Joint,Segment] = Inverse_Dynamics_VE(Joint,Segment,f,fc,n)
Inputs: Joint, Segment (*cf.* data structure), f (i.e., sampling frequency), fc (i.e., cut off frequency), n (i.e., number of sampled instants of time)
Outputs: Joint, Segment (*cf.* data structure)

Description:
Data formatting and call of functions Kinematics_VE.m and Dynamics_VE.m

See also:
Extend_Segment_Fields.m
Segment_Kinematics_VE.m
Dynamics_VE.m

Inverse_Dynamics_WQ.m

Purpose:
Computation of joint moments and forces by wrench and quaternion method

Synopsis:
[Joint,Segment] = Inverse_Dynamics_WQ(Joint,Segment,f,fc,n)
Inputs: Joint, Segment, (*cf.* data structure), f (i.e., sampling frequency), fc (i.e., cut off frequency), n (i.e., number of sampled instants of time)
Outputs: Joint, Segment (*cf.* data structure)

Description:
Data formatting and call of functions Kinematics_WQ.m and Dynamics_WQ.m

See also:
Extend_Segment_Fields.m
Segment_Kinematics_WQ.m
Dynamics_WQ.m

Joint_Kinematics.m

Purpose:
Computation of joint angles and displacements

Synopsis:
Joint = Joint_Kinematics(Segment)
Inputs: Segment (*cf.* data structure)
Outputs: Joint (*cf.* data structure)

Description:
Definition of JCS axes ($\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$) from generalized coordinates (**Q**), and computation of the joint angles and displacement about these axes

Theoretical background:
For ZXY sequence:

$$\mathbf{R}_i\left(\theta_1,\theta_2,\theta_3\right)=\mathbf{B}_{i+1}^{\mathbf{wu}}.\left[\mathbf{u}_{i+1}\quad \mathbf{r}_{P_{i+1}}-\mathbf{r}_{D_{i+1}}\quad \mathbf{w}_{i+1}\right]^{-1}.\left[\mathbf{u}_i\quad \mathbf{r}_{P_i}-\mathbf{r}_{D_i}\quad \mathbf{w}_i\right]\left[\mathbf{B}_i^{\mathbf{uv}}\right]^{-1}$$

For ZYX sequence:

$$\mathbf{R}_i\left(\theta_1,\theta_2,\theta_3\right)=\mathbf{B}_{i+1}^{\mathbf{wu}}.\left[\mathbf{u}_{i+1}\quad \mathbf{r}_{P_{i+1}}-\mathbf{r}_{D_{i+1}}\quad \mathbf{w}_{i+1}\right]^{-1}.\left[\mathbf{u}_i\quad \mathbf{r}_{P_i}-\mathbf{r}_{D_i}\quad \mathbf{w}_i\right]\left[\mathbf{B}_i^{\mathbf{uw}}\right]^{-1}$$

$$\mathbf{r}_{P_i}-\mathbf{r}_{D_{i+1}} = \underbrace{\frac{\left(\mathbf{e}_2\times\mathbf{e}_3\right)\bullet\left(\mathbf{r}_{P_i}-\mathbf{r}_{D_{i+1}}\right)}{\left(\mathbf{e}_1\times\mathbf{e}_2\right)\bullet\mathbf{e}_3}}_{d_1}\mathbf{e}_1 + \underbrace{\frac{\left(\mathbf{e}_3\times\mathbf{e}_1\right)\bullet\left(\mathbf{r}_{P_i}-\mathbf{r}_{D_{i+1}}\right)}{\left(\mathbf{e}_1\times\mathbf{e}_2\right)\bullet\mathbf{e}_3}}_{d_2}\mathbf{e}_2 + \underbrace{\frac{\left(\mathbf{e}_1\times\mathbf{e}_2\right)\bullet\left(\mathbf{r}_{P_i}-\mathbf{r}_{D_{i+1}}\right)}{\left(\mathbf{e}_1\times\mathbf{e}_2\right)\bullet\mathbf{e}_3}}_{d_3}\mathbf{e}_3$$

Reference:
R Dumas, T Robert, V Pomero, L Cheze. Joint and segment coordinate systems revisited. Computer Methods in Biomechanics and Biomedical Engineering 2012; 15(Suppl 1): 183-5

See also:
Mprod_array3.m
Tinv_array3.m
Q2Tuv_array3.m
Q2Tuw_array3.m
Q2Twu_array3.m
R2mobileZXY_array3.m
R2mobileZYX_array3.m
Vnop_array3.m

Mfilt_array3.m

Purpose:
Filtering of matrix

Synopsis:
Mf = Mfilt_array3(M,f,fc)
Inputs: M (i.e., matrix), f (i.e., sampling frequency), fc (i.e., cut off frequency)
Outputs: Mf (i.e., matrix)

Description:
Filtering, along with the 3rd dimension (i.e., all sampled instants of time, *cf.* data structure), of the matrix elements by a 4th order Butterworth

See also:
Vfilt_array3.m

Minv_array3.m

Purpose:
Computation of matrix inverse

Synopsis:
B = Minv_array3(A)
Inputs: A (i.e., matrix)
Outputs: B (i.e., matrix)

Description:
Computation of the inverse of a square matrix made for all sampled instants of time (i.e., in 3^{rd} dimension, *cf.* data structure) in the case of rotation matrix (**R**)

See also:
Mprod_array3.m
Tinv_array3.m

Mprod_array3.m

Purpose:
Computation of matrix product

Synopsis:
C = Mprod_array3(A,B)
Inputs: A, B (i.e., matrices)
Outputs: C (i.e., matrix)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the product of two matrices of compatible sizes

See also:
Minv_array3.m

Multibody_Optimisation_NNN.m

Purpose:
Multibody optimisation: minimisation of the sum of the square distances between measured and model-determined marker positions subject to kinematic and rigid body constraints (Gauss-Newton algorithm)

Synopsis:
Segment = Multibody_Optimisation_NNN(Segment)
Inputs: Segment (*cf.* data structure)
Outputs: Segment (*cf.* data structure)

Description:
Computation of **Q** by minimisation under constraints (the number of DoFs are 6 at the ankle/wrist, knee/elbow and hip/shoulder)

Reference:
S Duprey, L Cheze, R Dumas. Influence of joint constraints on lower limb kinematics estimation from skin markers using global optimization. Journal of Biomechanics 2010; 43(14): 2858-62

See also:
Multibody_Optimisation_SSS.m
Multibody_Optimisation_UHS.m
Multibody_Optimisation_UUS.m

Multibody_Optimisation_SSS.m

Purpose:
Multibody optimisation: minimisation of the sum of the square distances between measured and model-determined marker positions subject to kinematic and rigid body constraints (Gauss-Newton algorithm)

Synopsis:
Segment = Multibody_Optimisation_SSS(Segment)
Inputs: Segment (*cf.* data structure)
Outputs: Segment (*cf.* data structure)

Description:
Computation of **Q** by minimisation under constraints (the number of DoFs are 3 at the ankle/wrist, knee/elbow and hip/shoulder)

Reference:
S Duprey, L Cheze, R Dumas. Influence of joint constraints on lower limb kinematics estimation from skin markers using global optimization. Journal of Biomechanics 2010; 43(14): 2858-62

See also:
Multibody_Optimisation_NNN.m
Multibody_Optimisation_UHS.m
Multibody_Optimisation_UUS.m

Multibody_Optimisation_UHS.m

Purpose:
Multibody optimisation: minimisation of the sum of the square distances between measured and model-determined marker positions subject to kinematic and rigid body constraints (Gauss-Newton algorithm)

Synopsis:
Segment = Multibody_Optimisation_UHS(Segment)
Inputs: Segment (*cf.* data structure)
Outputs: Segment (*cf.* data structure)

Description:
Computation of **Q** by minimisation under constraints (the number of DoFs are 2 at the ankle, 1 at the knee and 3 at the hip)

Reference:
S Duprey, L Cheze, R Dumas. Influence of joint constraints on lower limb kinematics estimation from skin markers using global optimization. Journal of Biomechanics 2010; 43(14): 2858-62

See also:
Multibody_Optimisation_NNN.m
Multibody_Optimisation_SSS.m
Multibody_Optimisation_UUS.m

Multibody_Optimisation_UUS.m

Purpose:
Multibody optimisation: minimisation of the sum of the square distances between measured and model-determined marker positions subject to kinematic and rigid body constraints (Gauss-Newton algorithm)

Synopsis:
Segment = Multibody_Optimisation_UHS(Segment)
Inputs: Segment (*cf.* data structure)
Outputs: Segment (*cf.* data structure)

Description:
Computation of **Q** by minimisation under constraints (the number of DoFs are 2 at the wrist, 2 at the elbow and 3 at the shoulder)

Reference:
S Duprey, L Cheze, R Dumas. Influence of joint constraints on lower limb kinematics estimation from skin markers using global optimization. Journal of Biomechanics 2010; 43(14): 2858-62

See also:
Multibody_Optimisation_NNN.m
Multibody_Optimisation_SSS.m
Multibody_Optimisation_UHS.m

q2R_array3.m

Purpose:
Computation of rotation matrix from quaternion

Synopsis:
q = q2R_array3(R)
Inputs: q (i.e., quaternion)
Outputs: R (i.e., rotation matrix)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the rotation matrix (**R**) from the quaternion (**q**)

Theoretical background:

$$\mathbf{R} = \left[\left(q_0{}^2 + \mathbf{q}_v{}^T \mathbf{q}_v \right) \mathbf{E}_{\mathbf{3 \times 3}} + 2\mathbf{q}_v \mathbf{q}_v{}^T + 2 q_0 \tilde{\mathbf{q}}_v \right] \text{ with } \mathbf{q} = \begin{pmatrix} q_0 \\ \mathbf{q}_v \end{pmatrix}$$

Reference:
R Dumas, R Aissaoui, J A de Guise. A 3D generic inverse dynanmic method using wrench notation and quaternion algebra. Computer Methods in Biomechanics and Biomedical Engineering 2004; 7(3): 159-66

See also:
Extend_Segment_Fields.m
q2R_array3.m
Vskew_array3.m

Q2Tuv_array3.m

Purpose:
Computation of homogenous matrix from natural coordinates

Synopsis:
T = Q2Tuv_array3(Q)
Inputs: Q (i.e., natural coordinates)
Outputs: T (i.e., homogenous matrix)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the homogenous matrix (**T**) from natural coordinates (**Q**) with origin at endpoint *P* and axis correspondence $\mathbf{X} = \mathbf{u}$ and $\mathbf{Z} = \dfrac{\mathbf{u} \times \mathbf{v}}{\left\| \mathbf{u} \times \mathbf{v} \right\|}$

Theoretical background:

$$\mathbf{T}^{\mathbf{uv}} = \begin{bmatrix} \begin{bmatrix} \mathbf{u} & (\mathbf{r}_P - \mathbf{r}_D) & \mathbf{w} \end{bmatrix} \begin{bmatrix} \mathbf{B}^{\mathbf{uv}} \end{bmatrix}^{-1} & \mathbf{r}_P \\ 0 \; 0 \; 0 & 1 \end{bmatrix} \text{ with}$$

$$\mathbf{B}^{\mathbf{uv}} = \begin{bmatrix} 1 & L\cos\gamma & \cos\beta \\ 0 & L\sin\gamma & \dfrac{\cos\alpha - \cos\beta\cos\gamma}{\sin\gamma} \\ 0 & 0 & \sqrt{1 - (\cos\beta)^2 - \left(\dfrac{\cos\alpha - \cos\beta\cos\gamma}{\sin\gamma}\right)^2} \end{bmatrix}$$

$$L = \sqrt{(\mathbf{r}_P - \mathbf{r}_D)^2}$$

$$\alpha = \cos^{-1}\left(\frac{(\mathbf{r}_P - \mathbf{r}_D) \bullet \mathbf{w}}{L}\right)$$

$$\beta = \cos^{-1}(\mathbf{u} \bullet \mathbf{w})$$

$$\gamma = \cos^{-1}\left(\frac{\mathbf{u} \bullet (\mathbf{r}_P - \mathbf{r}_D)}{L}\right)$$

Reference:
R Dumas, L Cheze. 3D inverse dynamics in non-orthonormal segment coordinate system. Medical & Biological Engineering & Computing 2007; 45(3): 315-22
R Dumas, T Robert, V Pomero, L Cheze. Joint and segment coordinate systems revisited. Computer Methods in Biomechanics and Biomedical Engineering 2012; 15(Suppl 1): 183-5

See also:
Joint_Kinematics.m
Extend_Segment_Fields.m

Q2Tuw_array3.m

Purpose:
Computation of homogenous matrix from natural coordinates

Synopsis:
T = Q2Tuw_array3(Q)
Inputs: Q (i.e., natural coordinates)
Outputs: T (i.e., homogenous matrix)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the homogenous matrix (**T**) from natural coordinates (**Q**) with origin at endpoint P and axis correspondence $\mathbf{X} = \mathbf{u}$ and $\mathbf{Y} = \frac{\mathbf{w}\times\mathbf{u}}{\left\|\mathbf{w}\times\mathbf{u}\right\|}$

Theoretical background:

$$\mathbf{T}^{\mathbf{uw}} = \begin{bmatrix} \begin{bmatrix}\mathbf{u} & (\mathbf{r}_P - \mathbf{r}_D) & \mathbf{w}\end{bmatrix}\begin{bmatrix}\mathbf{B}^{\mathbf{uw}}\end{bmatrix}^{-1} & \mathbf{r}_P \\ 0\ 0\ 0 & 1 \end{bmatrix} \text{ with}$$

$$\mathbf{B}^{\mathbf{uw}} = \begin{bmatrix} 1 & L\cos\gamma & \cos\beta \\ 0 & L\sqrt{1-(\cos\gamma)^2 - \left(\frac{\cos\alpha - \cos\gamma\cos\beta}{\sin\beta}\right)^2} & 0 \\ 0 & L\frac{\cos\alpha - \cos\gamma\cos\beta}{\sin\beta} & \sin\beta \end{bmatrix}$$

$$L = \sqrt{(\mathbf{r}_P - \mathbf{r}_D)^2}$$

$$\alpha = \cos^{-1}\left(\frac{(\mathbf{r}_P - \mathbf{r}_D)\bullet\mathbf{w}}{L}\right)$$

$$\beta = \cos^{-1}(\mathbf{u}\bullet\mathbf{w})$$

$$\gamma = \cos^{-1}\left(\frac{\mathbf{u}\bullet(\mathbf{r}_P - \mathbf{r}_D)}{L}\right)$$

See also:
Joint_Kinematics.m
Q2Tuv_array3.m
Q2Twu_array3.m

Q2Twu_array3.m

Purpose:
Computation of homogenous matrix from natural coordinates

Synopsis:
T = Q2Twu_array3(Q)
Inputs: Q (i.e., natural coordinates)
Outputs: T (i.e., homogenous matrix)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the homogenous matrix (**T**) from natural coordinates (**Q**) with origin at endpoint D and axis correspondence $\mathbf{Z} = \mathbf{w}$ and $\mathbf{Y} = \dfrac{\mathbf{w} \times \mathbf{u}}{\left\| \mathbf{w} \times \mathbf{u} \right\|}$

Theoretical background:

$$\mathbf{T}^{\mathbf{wu}} = \begin{bmatrix} \begin{bmatrix} \mathbf{u} & (\mathbf{r}_P - \mathbf{r}_D) & \mathbf{w} \end{bmatrix} \begin{bmatrix} \mathbf{B}^{\mathbf{wu}} \end{bmatrix}^{-1} & \mathbf{r}_D \\ 000 & 1 \end{bmatrix} \text{ with}$$

$$\mathbf{B}^{\mathbf{wu}} = \begin{bmatrix} \sin\beta & L\dfrac{\cos\gamma - \cos\alpha\cos\beta}{\sin\beta} & 0 \\ 0 & L\sqrt{1 - (\cos\alpha)^2 - \left(\dfrac{\cos\gamma - \cos\alpha\cos\beta}{\sin\beta}\right)^2} & 0 \\ \cos\beta & L\cos\alpha & 1 \end{bmatrix}$$

$$L = \sqrt{(\mathbf{r}_P - \mathbf{r}_D)^2}$$

$$\alpha = \cos^{-1}\left(\frac{(\mathbf{r}_P - \mathbf{r}_D) \bullet \mathbf{w}}{L}\right)$$

$$\beta = \cos^{-1}(\mathbf{u} \bullet \mathbf{w})$$

$$\gamma = \cos^{-1}\left(\frac{\mathbf{u} \bullet (\mathbf{r}_P - \mathbf{r}_D)}{L}\right)$$

Reference:
R Dumas, T Robert, V Pomero, L Cheze. Joint and segment coordinate systems revisited. Computer Methods in Biomechanics and Biomedical Engineering 2012; 15(Suppl 1): 183-5

See also:
Joint_Kinematics.m
Q2Tuv_array3.m
Q2Tuw_array3.m

qinv_array3.m

Purpose:
Computation of quaternion inverse

Synopsis:
p = qinv_array3(q)
Inputs: q (i.e., quaternion)
Outputs: p (i.e., quaternion)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the inverse of a quaternion

Theoretical background:

$$\mathbf{q}^{-1} = \frac{\overline{\mathbf{q}}}{\|\mathbf{q}\|} = \frac{1}{\|\mathbf{q}\|}\begin{pmatrix} q_0 \\ -\mathbf{q}_v \end{pmatrix} \text{ with } \mathbf{q} = \begin{pmatrix} q_0 \\ \mathbf{q}_v \end{pmatrix}$$

Reference:
R Dumas, R Aissaoui, J A de Guise. A 3D generic inverse dynanmic method using wrench notation and quaternion algebra. Computer Methods in Biomechanics and Biomedical Engineering 2004; 7(3): 159-66

See also:
qprod_array3.m

qlog_array3.m

Purpose:
Computation of quaternion logarithm

Synopsis:
halfktheta = qlog_array3(q)
Inputs: q (i.e., quaternion)
Outputs: halfktheta (i.e., spatial attitude vector divided by 2)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the haft of spatial attitude vector from a quaternion

Theoretical background:

$$\frac{\mathbf{k}\theta}{2} = \frac{\mathbf{q}_v}{\left\|\mathbf{q}_v\right\|}\cos^{-1}\left(q_0\right) \text{ with } \mathbf{q} = \begin{pmatrix} q_0 \\ \mathbf{q}_v \end{pmatrix} \text{ as } \mathbf{q} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \mathbf{k}\sin\left(\frac{\theta}{2}\right) \end{bmatrix}$$

Reference:
J Lee, SY Shin. General construction of time-domain filters for orientation data. IEEE Transactions on Visualization and Computer Graphics 2002; 8(2): 119-28

See also:
Vexp_array3.m

qprod_array3.m

Purpose:
Computation of quaternion product

Synopsis:
r = Mprod_array3(p,q)
Inputs: p, q (i.e., quaternions)
Outputs: r (i.e., quaternion)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the product of two quaternions

Theoretical background:

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_0 q_0 - \mathbf{p}_v{}^T \mathbf{q}_v \\ p_0 \mathbf{q}_v + q_0 \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} p_0 & -\mathbf{p}_v{}^T \\ \mathbf{p}_v & p_0 \mathbf{E}_{3\times 3} + \tilde{\mathbf{p}}_v \end{bmatrix} \begin{pmatrix} q_0 \\ \mathbf{q}_v \end{pmatrix} \text{ with } \mathbf{p} = \begin{pmatrix} p_0 \\ \mathbf{p}_v \end{pmatrix} \text{ and } \mathbf{q} = \begin{pmatrix} q_0 \\ \mathbf{q}_v \end{pmatrix}$$

Reference:
R Dumas, R Aissaoui, J A de Guise. A 3D generic inverse dynanmic method using wrench notation and quaternion algebra. Computer Methods in Biomechanics and Biomedical Engineering 2004; 7(3): 159-66

See also:
qinv_array3.m

R2fixedZYX_array3.m

Purpose:
Computation of Euler angles from rotation matrix with a ZYX fixed (i.e., space) sequence for segment kinematics)

Synopsis:
Segment_Euler_Angles = R2fixedZYX_array3(R)
Inputs: R (i.e., rotation matrix)
Outputs: Segment_Euler_Angles (i.e., φ, θ, ψ, in line)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the Euler angles (φ, θ, ψ) from the rotation matrix (**R**) using a sequence of fixed axes ZYX

Theoretical background:

$$\mathbf{R} = \begin{bmatrix} \cos\varphi\cos\theta & -\sin\varphi\cos\theta & \sin\theta \\ \cos\varphi\sin\theta\sin\psi+\sin\varphi\cos\psi & -\sin\varphi\sin\theta\sin\psi+\cos\varphi\cos\psi & -\cos\theta\sin\psi \\ -\cos\varphi\sin\theta\cos\psi+\sin\varphi\sin\psi & \sin\varphi\sin\theta\cos\psi+\cos\varphi\sin\psi & \cos\theta\cos\psi \end{bmatrix}$$

$$\varphi = \tan^{-1}\left(\frac{-\mathbf{R}_{12}}{\mathbf{R}_{11}}\right)$$

$$\theta = \sin^{-1}\left(\mathbf{R}_{13}\right)$$

$$\psi = \tan^{-1}\left(\frac{-\mathbf{R}_{23}}{\mathbf{R}_{33}}\right)$$

Reference:
L Cheze, R Dumas, JJ Comtet, C Rumelhart, M Fayet. A joint coordinate system proposal for the study of the trapeziometacarpal joint kinematics. Computer Methods in Biomechanics and Biomedical Engineering 2009; 12(3): 277-82

See also:
Segment_Kinematics_VE.m
Extend_Segment_Fields.m
R2mobileXZY_array3.m
R2mobileZYX_array3.m
R2mobileZXY_array3.m

R2mobileXZY_array3.m

Purpose:
Computation of Euler angles from rotation matrix with a XYZ mobile (i.e., body) sequence for joint kinematics

Synopsis:
Joint_Euler_Angles = R2mobileXZY_array3(R)
Inputs: R (i.e., rotation matrix)
Outputs: Joint_Euler_Angles (i.e., θ_1, θ_2, θ_3, in line)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the 3 Euler angles (θ_1, θ_2, θ_3) from the rotation matrix (**R**) using a sequence of mobile axes XZY

Theoretical background:

$$\mathbf{R} = \begin{bmatrix} \cos\theta_2\cos\theta_3 & -\sin\theta_2 & \cos\theta_2\sin\theta_3 \\ \cos\theta_1\sin\theta_2\cos\theta_3 + \sin\theta_3\sin\theta_1 & \cos\theta_1\cos\theta_2 & \cos\theta_1\sin\theta_2\sin\theta_3 - \cos\theta_3\sin\theta_1 \\ \sin\theta_1\sin\theta_2\cos\theta_3 - \sin\theta_3\cos\theta_1 & \sin\theta_1\cos\theta_2 & \sin\theta_1\sin\theta_2\sin\theta_3 - \cos\theta_3\cos\theta_1 \end{bmatrix}$$

$$\theta_1 = \tan^{-1}\left(\frac{\mathbf{R}_{32}}{\mathbf{R}_{22}}\right)$$

$$\theta_2 = \sin^{-1}\left(-\mathbf{R}_{12}\right)$$

$$\theta_3 = \tan^{-1}\left(\frac{\mathbf{R}_{13}}{\mathbf{R}_{11}}\right)$$

Reference:
M Senk, L Cheze. Rotation sequence as an important factor in shoulder kinematics. Clinical Biomechanics 2006; 21(S1): S3-8.
A Bonnefoy-Mazure, J Slawinski, A Riquet, JM Lévèque, C Miller, L Cheze. Rotation sequence is an important factor in shoulder kinematics. Application to the elite players' flat serves. Journal of Biomechanics 2010; 43(10): 2022-5.

See also:
Joint_Kinematics.m
R2fixedZYX_array3.m
R2mobileZXY_array3.m
R2mobileZYX_array3.m

R2mobileZXY_array3.m

Purpose:
Computation of Euler angles from rotation matrix with a ZXY mobile (i.e., body) sequence for joint kinematics

Synopsis:
Joint_Euler_Angles = R2mobileZXY_array3(R)
Inputs: R (i.e., rotation matrix)
Outputs: Joint_Euler_Angles (i.e., θ_1, θ_2, θ_3, in line)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the 3 Euler angles (θ_1, θ_2, θ_3) from the rotation matrix (**R**) using a sequence of mobile axes ZXY

Theoretical background:

$$\mathbf{R} = \begin{bmatrix} -\sin\theta_3 \sin\theta_2 \sin\theta_1 + \cos\theta_3 \cos\theta_1 & -\cos\theta_2 \sin\theta_1 & \cos\theta_3 \sin\theta_2 \sin\theta_1 + \sin\theta_3 \cos\theta_1 \\ \sin\theta_3 \sin\theta_2 \cos\theta_1 + \cos\theta_3 \sin\theta_1 & \cos\theta_2 \cos\theta_1 & -\cos\theta_3 \sin\theta_2 \cos\theta_1 + \sin\theta_3 \sin\theta_1 \\ -\sin\theta_3 \cos\theta_2 & \sin\theta_2 & \cos\theta_3 \cos\theta_2 \end{bmatrix}$$

$$\theta_1 = \tan^{-1}\left(\frac{-\mathbf{R}_{12}}{\mathbf{R}_{22}}\right)$$

$$\theta_2 = \sin^{-1}\left(\mathbf{R}_{32}\right)$$

$$\theta_3 = \tan^{-1}\left(\frac{-\mathbf{R}_{31}}{\mathbf{R}_{33}}\right)$$

References:

G Wu, S Siegler, P Allard, C Kirtley, A Leardini, D Rosenbaum, M Whittle, DD D'Lima, L Cristofolini, H Witte, O Schmid, I Stokes. ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion - Part I: ankle, hip, and spine. Journal of Biomechanics 2002; 35(4): 543-8
G Wu, FC van der Helm, HE Veeger, M Makhsous, P Van Roy, C Anglin, J Nagels, AR Karduna, K McQuade, X Wang, FW Werner, B Buchholz. ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion - Part II: shoulder, elbow, wrist and hand. Journal of Biomechanics 2005; 38(5): 981-92

See also:
Joint_Kinematics.m
R2fixedZYX_array3.m
R2mobileXZY_array3.m
R2mobileZYX_array3.m

R2mobileZYX_array3.m

Purpose:
Computation of Euler angles from rotation matrix with a ZYX mobile (i.e., body) sequence for joint kinematics

Synopsis:
Joint_Euler_Angles = R2mobileZYX_array3(R)
Inputs: R (i.e., rotation matrix)
Outputs: Joint_Euler_Angles (i.e., θ_1, θ_2, θ_3, in line)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the 3 Euler angles (θ_1, θ_2, θ_3) from the rotation matrix (**R**) using a sequence of mobile axes ZYX

Theoretical background:

$$\mathbf{R} = \begin{bmatrix} \cos\theta_2\cos\theta_1 & \sin\theta_3\sin\theta_2\cos\theta_1 - \cos\theta_3\sin\theta_1 & \cos\theta_3\sin\theta_2\cos\theta_1 + \sin\theta_3\sin\theta_1 \\ \cos\theta_2\sin\theta_1 & \sin\theta_3\sin\theta_2\sin\theta_1 + \cos\theta_3\cos\theta_1 & \cos\theta_3\sin\theta_2\sin\theta_1 - \sin\theta_3\cos\theta_1 \\ -\sin\theta_2 & \sin\theta_3\cos\theta_2 & \cos\theta_3\cos\theta_2 \end{bmatrix}$$

$$\theta_1 = \tan^{-1}\left(\frac{\mathbf{R}_{21}}{\mathbf{R}_{11}}\right)$$

$$\theta_2 = \sin^{-1}\left(-\mathbf{R}_{31}\right)$$

$$\theta_3 = \tan^{-1}\left(\frac{\mathbf{R}_{32}}{\mathbf{R}_{33}}\right)$$

Reference:
GK Cole, BM Nigg, JL Ronsky MR Yeadon. Application of the joint coordinate system to three-dimensional joint attitude and movement representation: a standardisation proposal. Journal of Biomechanical Engineering 1993; 115(4): 344-349
R Dumas, T Robert, V Pomero, L Cheze. Joint and segment coordinate systems revisited. Computer Methods in Biomechanics and Biomedical Engineering 2012; 15(Suppl 1): 183-5

See also:
Joint_Kinematics.m
R2fixedZYX_array3.m
R2mobileXZY_array3.m
R2mobileZXY_array3.m

R2q_array3.m

Purpose:
Computation of quaternion from rotation matrix

Synopsis:
q = R2q_array3(R)
Inputs: R (i.e., rotation matrix)
Outputs: q (i.e., quaternion)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the quaternion (**q**) from the rotation matrix (**R**)

References:
SW Shepperd. Quaternion from rotation matrix. Journal of Guidance and Control 1978;1(3):223-224
J Lee, SY Shin. General construction of time-domain filters for orientation data. IEEE Transactions on Visualization and Computer Graphics 2002; 8(2): 119-28
FL Markley. Unit quaternion from rotation matrix. Journal of Guidance, Control, and Dynamics 2008; 31(2): 440-2

See also:
Extend_Segment_Fields.m
q2R_array3.m
qprod_array3.m
qinv_array3.m
qlog_array3.m

SARA_array3.m

Purpose:
Functional method for axis of rotation (AoR)

Synopsis:
[a,asi,asj,rA,rAsi,rAsj] = SARA_array3(Ti,Tj)
Inputs: Ti (i.e., homogenous matrix) of segment i (e.g., proximal), Tj (i.e., homogenous matrix) of segment j (e.g., distal)
Outputs: a (i.e., orientation of AoR expressed in ICS), asi (i.e., orientation of AoR expressed in SCS of segment i (e.g., proximal)), asj (i.e., orientation of AoR expressed in SCS of segment j (e.g., distal)), rA (i.e., position of a point A of AoR expressed in ICS), rAsi (i.e., position of a point A of AoR expressed in SCS of segment i (e.g., proximal)), rAsj (i.e., position of a point A of AoR expressed in SCS of segment j (e.g., distal))

Description:
Computation of the averaged axis of rotation (AoR) between two segments (e.g., proximal and distal) by symmetrical axis of rotation approach (SARA)

References:
RM Ehrig, WR Taylor, GN Duda, MO Heller. A survey of formal methods for determining functional joint axes. Journal of Biomechanics 2007; 40: 2150–7

See also:
SCoRE_array3.m

SCoRE_array3.m

Purpose:
Functional method for centre of rotation (CoR)

Synopsis:
[rC,rCsi,rCsj] = SCoRE_array3(Ti,Tj)
Inputs: Ti (i.e., homogenous matrix) of segment i (e.g., proximal), Tj (i.e., homogenous matrix) of segment j (e.g., distal)
Outputs: rC (i.e., position of CoR expressed in ICS), rCsi (i.e., position of CoR expressed in SCS of segment i (e.g., proximal)), rCsj (i.e., position of CoR expressed in SCS of segment j (e.g., distal))

Description:
Computation of the averaged Center of Rotation (CoR) between two segments (e.g., proximal and distal) by Symmetrical Centre of Rotation Estimation (SCoRE) method

References:
RM Ehrig, WR Taylor, GN Duda, MO Heller. A survey of formal methods for determining the centre of rotation of ball joints. Journal of Biomechanics 2006; 39: 2798–809

See also:
SARA_array3.m

Segment_Kinematics_HM.m

Purpose:
Computation of segment kinematics by homogenous matrix method

Synopsis:
Segment = Kinematics_HM(Segment,f,fc,n)
Inputs: Segment (*cf.* data structure), f (i.e., sampling frequency), fc (i.e., cut off frequency), n (i.e., number of sampled instants of time)
Outputs: Segment (*cf.* data structure)

Description:
Computation of homogenous matrices of velocity and acceleration (**W**, **H**) as a function of homogenous matrix of transformation from ICS to SCS (**T**)

Theoretical background:
$\mathbf{W} = \dot{\mathbf{T}}\mathbf{T}^{-1}$
$\mathbf{H} = \ddot{\mathbf{T}}\mathbf{T}^{-1}$

References:
N Doriot, L Cheze. A 3D Kinematic and dynamic study of the lower limb during the stance phase of gait using an homogeneous matrix approach. IEEE Transactions on Biomedical Engineering 2004;51(1):21-7
G Legnani, F Casolo, P Righettini, B Zappa, A homogeneous matrix approach to 3D kinematics and dynamics - I. Theory. Mechanisms and Machine Theory 1996; 31(5): 573–87

See also:
Inverse_Dynamics_HM.m
Dynamics_HM.m
Mprod_array3.m
Tinv_array3.m
Derive_array3.m
Mfilt_array3.m

Segment_Kinematics_VE.m

Purpose:
Computation of segment kinematics by Euler angles method

Synopsis:
Segment = Kinematics_VE(Segment,f,fc,n)
Inputs: Segment (*cf.* data structure), f (i.e., sampling frequency), fc (i.e., cut off frequency), n (i.e., number of sampled instants of time)
Outputs: Segment (*cf.* data structure)

Description:
Computation of segment angular velocity and acceleration ($\boldsymbol{\omega}$, $\boldsymbol{\alpha}$), linear velocity and acceleration of centre of mass (**v**, **a**) expressed in ICS, as a function of:

- position of proximal endpoint ($\mathbf{r}_P$) expressed in ICS
- Euler angles (φ, θ, ψ) or rotation matrix (**R**) of transformation from ICS to SCS
- position of centre of mass ($\mathbf{r}_C^s$) expressed in SCS

Theoretical background:

$$\boldsymbol{\omega} = \begin{pmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix}$$

$$\boldsymbol{\alpha} = \begin{pmatrix} \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\varphi} \end{pmatrix}$$

$$\mathbf{v} = \dot{\mathbf{r}}_P + \boldsymbol{\omega} \times \left(\mathbf{R}(\varphi,\theta,\psi) \cdot \mathbf{r}_C^s \right)$$

$$\mathbf{a} = \ddot{\mathbf{r}}_P + \boldsymbol{\alpha} \times \left(\mathbf{R}(\varphi,\theta,\psi) \cdot \mathbf{r}_C^s \right) + \boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \left(\mathbf{R}(\varphi,\theta,\psi) \cdot \mathbf{r}_C^s \right) \right)$$

References:
MP Kadaba, HK Ramakrishnan, ME Wootten, J Gainey, G Gorton, GV Cochran. Repeatability of kinematic, kinetic, and electromyographic data in normal adult gait. Journal of Orthopaedic Research 1989;7(6):849-60
RB Davis, S Ounpuu, D Tyburski, JR Gage. A gait analysis data collection and reduction technique. Human Movement Science 1991; 10: 575-87
CL Vaughan, BL Davis, JC O'Connor. Dynamics of human gait (2d edition). Human Kinetics, Champaign, Illinois, 1999
G Wu, PR Cavanagh. ISB recommendations for standardization in the reporting of kinematic data. Journal of Biomechanics 1995; 28(10): 1257-61

See also:
Inverse_Dynamics_VE.m
Dynamics_VE.m
R2fixedZYX_array3.m
Derrive_array3.m
Vfilt_array3.m

Segment_Kinematics_WQ.m

Purpose:
Computation of segment kinematics by quaternion method

Synopsis:
Segment = Kinematics_WQ(Segment,f,fc,n)
Inputs: Segment (*cf.* data structure), f (i.e., sampling frequency), fc (i.e., cut off frequency), n (i.e., number of sampled instants of time)
Outputs: Segment (*cf.* data structure)

Description:
Computation of segment angular velocity and acceleration (ω, α), linear velocity and acceleration of centre of mass ($\mathbf{v}$, $\mathbf{a}$) expressed in ICS, as a function of:

- position of proximal endpoint ($\mathbf{r}_P$) expressed in ICS
- quaternion ($\mathbf{q}$) of transformation from ICS to SCS
- position of centre of mass ($\mathbf{r}_C^s$) expressed in SCS

Theoretical background:

$$\begin{pmatrix} 0 \\ \boldsymbol{\omega} \end{pmatrix} = 2\dot{\mathbf{q}} \otimes \overline{\mathbf{q}}$$

$$\begin{pmatrix} 0 \\ \boldsymbol{\alpha} \end{pmatrix} = 2\left(\ddot{\mathbf{q}} \otimes \overline{\mathbf{q}} + \dot{\mathbf{q}} \otimes \dot{\overline{\mathbf{q}}}\right)$$

$$\begin{pmatrix} 0 \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} 0 \\ \dot{\mathbf{r}}_P \end{pmatrix} + \dot{\mathbf{q}} \otimes \begin{pmatrix} 0 \\ \mathbf{r}_C^s \end{pmatrix} \otimes \overline{\mathbf{q}} + \mathbf{q} \otimes \begin{pmatrix} 0 \\ \mathbf{r}_C^s \end{pmatrix} \otimes \dot{\overline{\mathbf{q}}}$$

$$\begin{pmatrix} 0 \\ \mathbf{a} \end{pmatrix} = \begin{pmatrix} 0 \\ \ddot{\mathbf{r}}_P \end{pmatrix} + \ddot{\mathbf{q}} \otimes \begin{pmatrix} 0 \\ \mathbf{r}_C^s \end{pmatrix} \otimes \overline{\mathbf{q}} + 2\left(\dot{\mathbf{q}} \otimes \begin{pmatrix} 0 \\ \mathbf{r}_C^s \end{pmatrix} \otimes \dot{\overline{\mathbf{q}}}\right) + \mathbf{q} \otimes \begin{pmatrix} 0 \\ \mathbf{r}_C^s \end{pmatrix} \otimes \ddot{\overline{\mathbf{q}}}$$

Reference:
R Dumas, R Aissaoui, J A de Guise. A 3D generic inverse dynanmic method using wrench notation and quaternion algebra. Computer Methods in Biomechanics and Biomedical Engineering 2004; 7(3): 159-66

See also:
Inverse_Dynamics_WQ.m
Dynamics_WQ.m
R2q_array3.m
q2R_array3.m
qprod_array3.m
qinv_array3.m
Derrive_array3.m
Vfilt_array3.m

Tinv_array3.m

Purpose:
Computation of homogenous matrix inverse

Synopsis:
B = Tinv_array3(A)
Inputs: A (i.e., homogenous matrix)
Outputs: B (i.e., homogenous matrix)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the inverse of an homogenous matrix (**T**)

Theoretical background:

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T \cdot \mathbf{r}_P \\ 000 & 1 \end{bmatrix} \text{ with } \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{r}_P \\ 000 & 1 \end{bmatrix}$$

See also:
Mprod_array3.m
Minv_array3.m

Vexp_array3.m

Purpose:
Computation of vector exponential

Synopsis:
q = Vexp_array3(halfktheta)
Inputs: halfktheta (i.e., spatial attitude vector divided by 2)
Outputs: q (i.e., quaternion)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the quaternion form the haft of spatial attitude vector

Theoretical background:

$$\mathbf{q} = \begin{pmatrix} \cos\left(\left\|\frac{\mathbf{k}\theta}{2}\right\|\right) \\ \dfrac{\frac{\mathbf{k}\theta}{2}}{\left\|\frac{\mathbf{k}\theta}{2}\right\|}\sin\left(\left\|\frac{\mathbf{k}\theta}{2}\right\|\right) \end{pmatrix}$$

Reference:
J Lee, SY Shin. General construction of time-domain filters for orientation data. IEEE Transactions on Visualization and Computer Graphics 2002; 8(2): 119-28

See also:
qlog_array3.m

Vfilt_array3.m

Purpose:
Filtering of vector

Synopsis:
Vf = Vfilt_array3(M,f,fc)
Inputs: V (i.e., vector), f (i.e., sampling frequency), fc (i.e., cut off frequency)
Outputs: Vf (i.e., matrix)

Description:
Filtering, along with the 3rd dimension (i.e., all sampled instants of time, *cf.* data structure), of the vector components by a 4th order Butterworth with special attention when the vector is a column of an homogenous matrix

See also:
Mfilt_array3.m

Note:
For Matlab user without Signal Processing Toolbox, butter.m can be replaced by myButter.m (www.mathworks.com/matlabcentral/answers/137778-butterworth-lowpass-filtering-without-signal-processing-toolbox)

Vnop_array3.m

Purpose:
Non-orthogonal projection on three basis vectors

Synopsis:
Vnop = Vnop_array3(V,e1,e2,e3)
Inputs: V (i.e., vector), e1, e2, e3 (i.e., basis vectors)
Outputs: Vnop (i.e., vector components)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the vector projections on three non-orthogonal basis vectors

Theoretical background:

$$\mathbf{V} = \underbrace{\frac{(\mathbf{e}_2 \times \mathbf{e}_3) \bullet \mathbf{V}}{(\mathbf{e}_1 \times \mathbf{e}_2) \bullet \mathbf{e}_3}}_{V_1} \mathbf{e}_1 + \underbrace{\frac{(\mathbf{e}_3 \times \mathbf{e}_1) \bullet \mathbf{V}}{(\mathbf{e}_1 \times \mathbf{e}_2) \bullet \mathbf{e}_3}}_{V_2} \mathbf{e}_2 + \underbrace{\frac{(\mathbf{e}_1 \times \mathbf{e}_2) \bullet \mathbf{V}}{(\mathbf{e}_1 \times \mathbf{e}_2) \bullet \mathbf{e}_3}}_{V_3} \mathbf{e}_3$$

References:
L Cheze. Comparison of different calculations of three-dimensional joint kinematics from video-based system data. Journal of Biomechanics 2000; 33(12):1695-9
G Desroches, L Cheze, R Dumas. Expression of joint moment in the joint coordinate system. Journal of Biomechanical Engineering 2010; 132(11): 114503

See also:
Joint_Kinematics.m

Vnorm_array3.m

Purpose:
Computation of a unitary vector

Synopsis:
Vn = Vnorm_array3(V)
Inputs: V (i.e., vector)
Outputs: Vn (i.e., vector)

Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the unitary vector from a vector

See also:
Vskew_array3.m

Vskew_array3.m

Purpose:
Computation of skew matrix of vector

Synopsis:
M = Vskew_array3(V)
Inputs: V (i.e., vector)
Outputs: M (i.e., matrix)

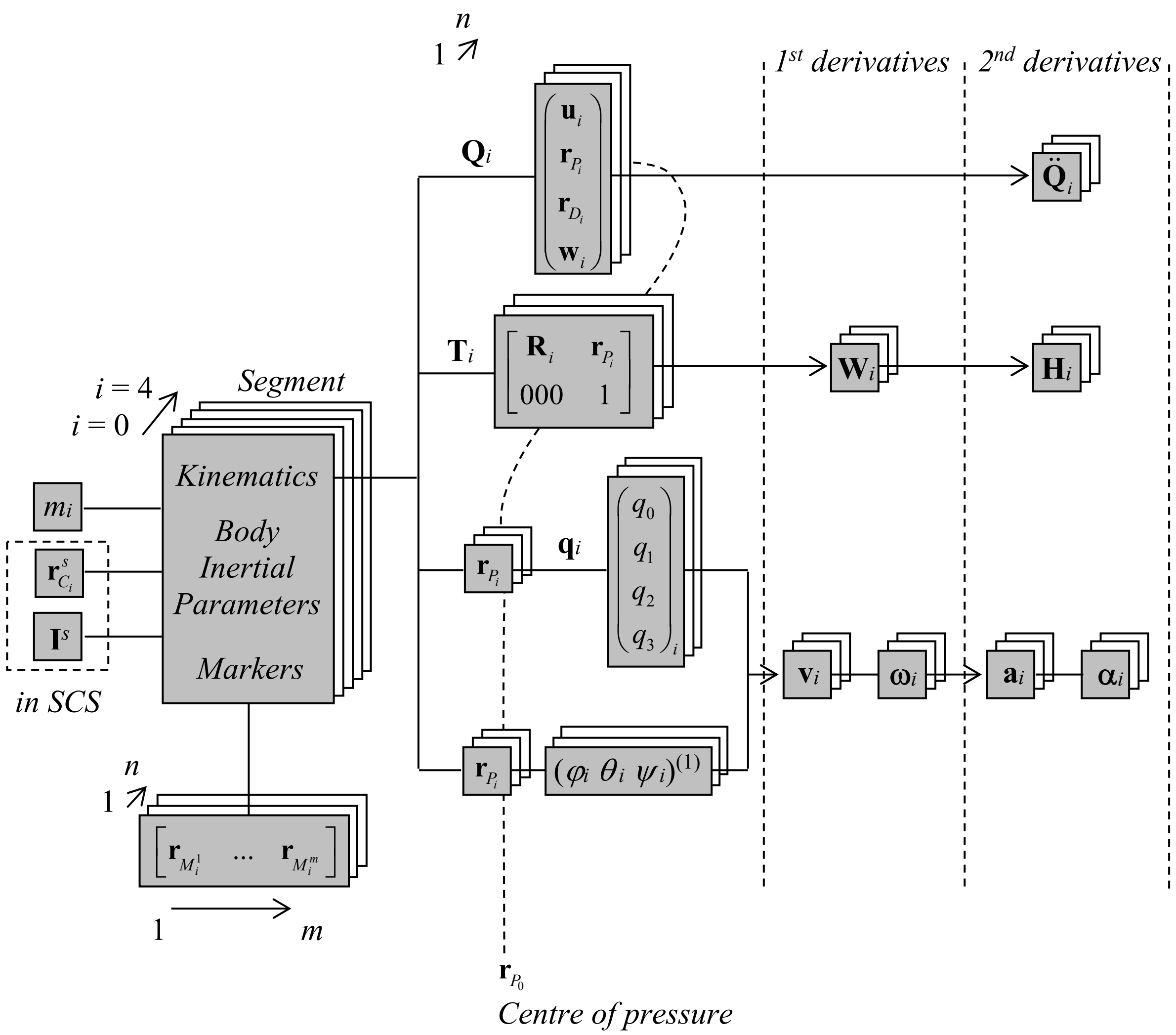
Description:
Computation, for all sampled instants of time (i.e., in 3rd dimension, *cf.* data structure), of the skew matrix from a vector

Theoretical background:

$$\tilde{\mathbf{V}} = \begin{bmatrix} 0 & -V_z & V_y \\ V_z & 0 & -V_x \\ -V_y & V_x & 0 \end{bmatrix}$$

See also:
Inverse_Dynamics_HM.m
Dynamics_WQ.m
q2R_array3.m
qprod_array3.m

Data format – Segment



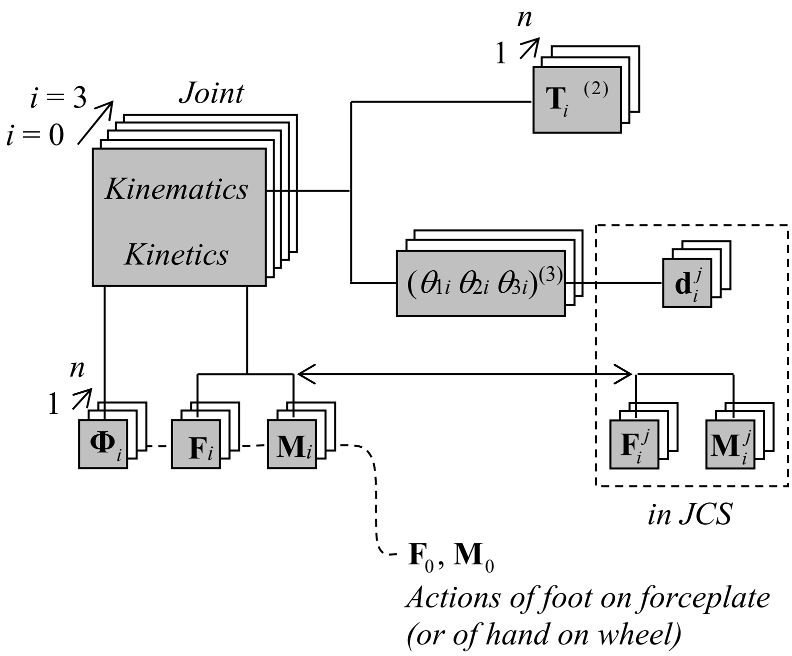
i = 0: forceplate (or wheel)
i = 1: foot (or hand)
i = 2: shank (or forearm)
i = 3: thigh (or arm)
i = 4: pelvis (or thorax)
n: number of sampled instants of time
m: number of markers

All kinematics $\mathbf{Q}$, $\mathbf{T}$, $\mathbf{q}$, $\varphi_i, \theta_i, \psi_i$ (1) represent the pose of the segment coordinate system (SCS) with respect to an inertial coordinate system (ICS).

---

(1) Sequence of fixed (i.e., space) axes ZYX

Data format – Joint



$i = 0$: centre of pressure
$i = 1$: ankle (or wrist)
$i = 2$: knee (or elbow)
$i = 3$: hip (or shoulder)
n: number of sampled instants of time

Kinematics $\mathbf{T}$, θ_{1i}, θ_{2i}, θ_{3i} [(3)], $\mathbf{d}_i^j$ represent the pose of the distal SCS with respect to the proximal SCS.
All kinetics $\boldsymbol{\Phi}_i$, $\mathbf{F}_i$, $\mathbf{M}_i$, $\mathbf{F}_i^j$, $\mathbf{M}_i^j$ represent the action of the proximal segment on the distal segment at the proximal joint centre of the distal segment, either expressed in the ICS or the joint coordinate system (JCS).

[(2)] $\mathbf{T}_i = \left[\mathbf{T}_{i+1}^{\mathbf{wu}}\right]^{-1} \mathbf{T}_i^{\mathbf{uv}}$ $\left(\text{or } \mathbf{T}_i = \left[\mathbf{T}_{i+1}^{\mathbf{wu}}\right]^{-1} \mathbf{T}_i^{\mathbf{uw}}\right)$

[(3)] Sequence of mobile (i.e. body) axes ZXY (or ZYX)

Additional notations

ICS: inertial coordinate system
SCS: segment coordinate system
JCS: joint coordinate system
^{s}: vector expression in SCS
^{j}: vector expression in JCS (i.e., non-orthogonal projection)
^{T}: transpose
$^{-1}$: inverse
$\bar{\ }$: quaternion conjugate
~: skew matrix of vector
$\dot{\ }$: 1st derivative
$\ddot{\ }$: 2nd derivative
$\bullet$: dot product
$\times$: cross product
$\| \ \|$: vector norm
$\mathbf{E}_{3\times3}$: identity matrix
$\mathbf{0}_{3\times3}$, $\mathbf{0}_{3\times1}$: zero matrices

Dealing with natural coordinates

The position vectors $\mathbf{r}_{P_i}$, $\mathbf{r}_{D_i}$ and the direction vectors $\mathbf{u}_i$, $\mathbf{w}_i$, all expressed in the ICS, are computed from the position $\mathbf{r}_{M_i^j}$ of classical skin-based markers.

Some rules shall be followed in order to build a non-orthogonal basis $\left(\mathbf{u}_i, \left(\mathbf{r}_{P_i} - \mathbf{r}_{D_i}\right), \mathbf{w}_i\right)$ consistent with both SCS axis (through $\mathbf{B}_i^{\mathbf{uv}}$) and JCS axes (through $\mathbf{B}_i^{\mathbf{uv}}$, $\mathbf{B}_i^{\mathbf{uw}}$ and $\mathbf{B}_i^{\mathbf{wu}}$):

- endpoints P_i and D_i are proximal and distal joint centres (i.e., estimated by regression equations or functional methods)
- $\mathbf{u}_i$ is normal to the frontal plane ($\mathbf{r}_{P_i} - \mathbf{r}_{D_i}$ and $\mathbf{u}_i$ lie in the sagittal plane)
- $\mathbf{w}_i$ is the distal joint flexion-extension axis (estimated anatomically or by functional methods)

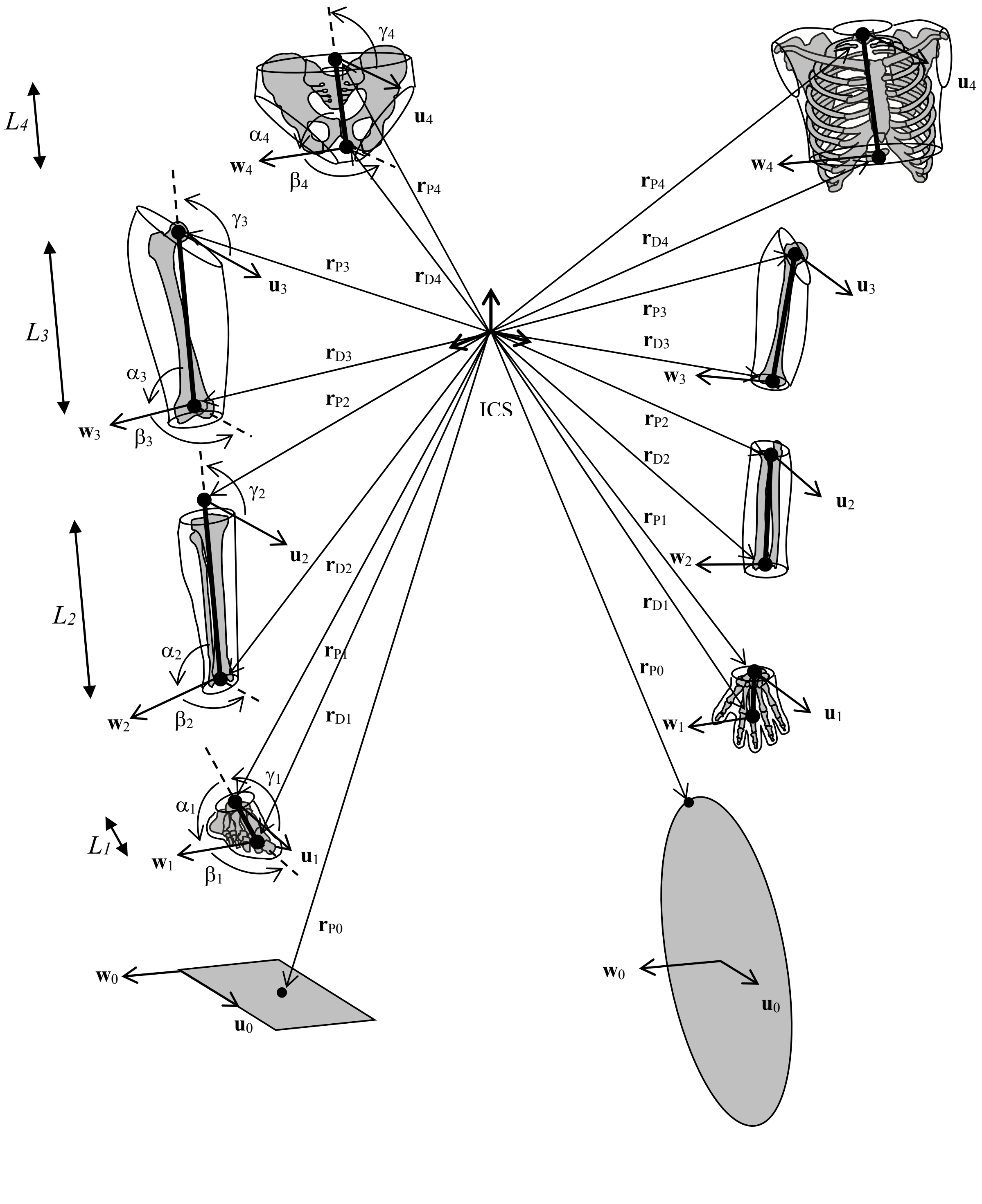
Reference:

R Dumas, L Cheze. 3D inverse dynamics in non-orthonormal segment coordinate system. Medical & Biological Engineering & Computing 2007; 45(3): 315-22

Example of construction of segment parameters Q

		Lower limb	Upper limb
i = 0: forceplate or wheel	$\mathbf{r}_{M_0}$	N/A	N/A
	$\mathbf{r}_{P_0}$	centre of pressure	centre of pressure
	$\mathbf{r}_{D_0}$	N/A (or centre of forceplate)	N/A (or centre of wheel)
	$\mathbf{u}_0$	X axis of forceplate (direction of progression)	X axis of wheelchair (direction of proplusion)
	$\mathbf{w}_0$	Z axis of forceplate	Z axis of wheelchair (axis of wheels)
i = 1: foot or hand	$\mathbf{r}_{M_1}$	calcaneus, 1rst and 5th metatarsal heads	2d and 5th metacarpal heads
	$\mathbf{r}_{P_1}$	$\mathbf{r}_{D_2}$	$\mathbf{r}_{D_2}$
	$\mathbf{r}_{D_1}$	metarsal-phalangeal joint centre (i.e., middle of metatarsal heads)	mecarpal-phalangeal joint centre (i.e., middle of metacarpal heads)
	$\mathbf{u}_1$	calcaneous to middle of metatarsal heads	normal to the plane formed by wrist joint centre (P_1) and metacarpal heads
	$\mathbf{w}_1$	1rst to 5th metatarsal head	5th to 2d metacarpal head
i = 2: shank or forearm	$\mathbf{r}_{M_2}$	fibula head, medial and lateral malleoli	ulnar and radial styloids
	$\mathbf{r}_{P_2}$	$\mathbf{r}_{D_3}$	$\mathbf{r}_{D_3}$
	$\mathbf{r}_{D_2}$	ankle joint centre (i.e., middle of malleoli)	wrist joint centre (i.e., middle of styloids)
	$\mathbf{u}_2$	normal to the plane formed by fibula head, knee and ankle joint centres (P_2 and D_2)	normal to the plane formed by elbow joint centre (P_2) and styloids
	$\mathbf{w}_2$	medial to lateral malleolus	ulnar to radial styloids
i = 3: thigh or arm	$\mathbf{r}_{M_3}$	great trochanter, medial and lateral femoral epicondyles	great tubercule, medial and lateral humeral epicondyles
	$\mathbf{r}_{P_3}$	hip joint centre (by regressions or functional methods)	shoulder joint centre (by regression equations or functional methods)
	$\mathbf{r}_{D_3}$	knee joint centre (i.e., middle of epicondyles)	elbow joint centre (i.e., middle of epicondyles)
	$\mathbf{u}_3$	normal to the plane formed by epicondyles and hip joint centre (P_3)	normal to the plane formed by epicondyles and shoulder joint centre (P_3)
	$\mathbf{w}_3$	medial to lateral epicondyle	medial to lateral epicondyle
i = 4: pelvis or thorax	$\mathbf{r}_{M_4}$	iliac spines	C7, Incisura Jugularis, Processus Xiphoideus, T8
	$\mathbf{r}_{P_4}$	lumbar joint centre (by regression equations, or middle of posterior superior iliac spines)	cervical joint centre (by regression equations, or middle of C7 and Incisura Jugularis)
	$\mathbf{r}_{D_4}$	middle of hip joint centres	thoracic joint centre (by regressions, or middle of T8 and Processus Xiphoideus)
	$\mathbf{u}_4$	from middle of posterior iliac spines to middle of anterior iliac spines	normal to both $\mathbf{r}_{P_4} - \mathbf{r}_{D_4}$ and $\mathbf{w}_4$
	$\mathbf{w}_4$	left to right hip joint centre	normal to the plane formed by C7, Incisura Jugularis and middle of T8 and Processus Xiphoideus

Schematic representation of segment parameters



Schematic representation of joint angles and displacements (for the lower limb)