Three-Dimensional Experimental Kinematics

Digitize locations of landmarks $\{r_i\}^{Pk}$ on body i for points k=1 to n at given time t

All points must be on same body i

Use landmark weighting factor $f^{Pk} = 1$ if point k is available at time t. Use $f^{Pk} = 0$ if point k not available at given time t .

Determine $\{\dot{r}_i\}^{Pk}$ $\{\ddot{r}_i\}^{Pk}$ $\{\dddot{r}_i\}^{Pk}$ at given time t .

Mean values

$$\{r_i\}^{mean} = \left(\sum_{k=1}^{n} f^{Pk} \{r_i\}^{Pk} \right) / \sum_{k=1}^{n} f^{Pk}$$

$$\{\dot{r}_i\}^{mean} = \left(\sum_{k=1}^{n} f^{Pk} \{\dot{r}_i\}^{Pk} \right) / \sum_{k=1}^{n} f^{Pk}$$

$$\{\ddot{r}_i\}^{mean} = \left(\sum_{k=1}^{n} f^{Pk} \{\ddot{r}_i\}^{Pk} \right) / \sum_{k=1}^{n} f^{Pk}$$

$$\{\dddot{r}_i\}^{mean} = \left(\sum_{k=1}^{n} f^{Pk} \{\dddot{r}_i\}^{Pk} \right) / \sum_{k=1}^{n} f^{Pk}$$

$$\{\ddddot{r}_i\}^{mean} = \left(\sum_{k=1}^{n} f^{Pk} \{\ddddot{r}_i\}^{Pk} \right) / \sum_{k=1}^{n} f^{Pk}$$

$$[X] = \left(\sum_{k=1}^{n} f^{Pk} \left(\{r_i\}^{Pk} - \{r_i\}^{mean} \right) \left(\{r_i\}^{Pk} - \{r_i\}^{mean} \right)^T \right) / \sum_{k=1}^{n} f^{Pk}$$

$$[M] = ([I_3]\mathrm{trace}([X])) - [X]$$

Velocity

$$[V] = \left(\sum_{k=1}^{n} f^{Pk} \{\dot{r}_i\}^{Pk} \left(\{r_i\}^{Pk} - \{r_i\}^{mean} \right)^T \right) / \sum_{k=1}^{n} f^{Pk}$$

$$\{\omega_i\} = [M]^{-1} \begin{Bmatrix} V_{32} - V_{23} \\ V_{13} - V_{31} \\ V_{21} - V_{12} \end{Bmatrix} = \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} \qquad \omega_i = \mathrm{norm}\{\omega_i\}$$

$$[\tilde{\omega}_i] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \qquad \{\hat{u}\} = \{\omega_i\} / \omega_i$$

Point ISA is on the instantaneous screw axis for body i at the root of the perpendicular from the centroid of the landmarks. Note that the ISA is not attached to the body. Any point P on the body coincident with the ISA has translational velocity $\dot{s}$ along the ISA.

$$\{\dot{r}_i\}^P = \dot{s}\{\hat{u}\} + [\tilde{\omega}_i]\left(\{r_i\}^P - \{r\}^{ISA}\right) \quad \text{for any point P attached to body i}$$

$$\{r\}^{ISA} = \{r_i\}^{mean} + [\tilde{\omega}]\{\dot{r}_i\}^{mean} / \omega_i^2$$

$$\dot{s} = \{\hat{u}\}^T \{\dot{r}_i\}^{mean}$$

Acceleration

$$[A] = \left(\sum_{k=1}^{n} f^{Pk} \{\ddot{r}_i\}^{Pk} \left(\{r_i\}^{Pk} - \{r_i\}^{mean} \right)^T \right) / \sum_{k=1}^{n} f^{Pk}$$

$$[B] = [A] - [\tilde{\omega}_i][\tilde{\omega}_i][X]$$

$$\{\dot{\omega}_i\} = [M]^{-1} \begin{Bmatrix} B_{32} - B_{23} \\ B_{13} - B_{31} \\ B_{21} - B_{12} \end{Bmatrix} = \begin{Bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{Bmatrix} \qquad \dot{\omega}_i = \text{norm}\{\dot{\omega}_i\}$$

$$[\tilde{\dot{\omega}}_i] = \begin{bmatrix} 0 & -\dot{\omega}_z & \dot{\omega}_y \\ \dot{\omega}_z & 0 & -\dot{\omega}_x \\ -\dot{\omega}_y & \dot{\omega}_x & 0 \end{bmatrix} \qquad [\beta_i] = [\tilde{\dot{\omega}}_i] + [\tilde{\omega}_i][\tilde{\omega}_i]$$

Point IAP is the instantaneous acceleration pole for body i. Note that the IAP is not attached to the body. Point P on the body coincident with IAP has zero acceleration.

$$\{\ddot{r}_i\}^P = [\beta_i]\left(\{r_i\}^P - \{r\}^{IAP}\right) \quad \text{for any point P attached to body i}$$

$$\{r\}^{IAP} = \{r_i\}^{mean} - [\beta_i]^{-1}\{\ddot{r}_i\}^{mean} \qquad \text{for} \qquad \{\ddot{r}_i\}^{P\_at\_IAP} = 0$$

Jerk

$$[J] = \left(\sum_{k=1}^{n} f^{Pk} \{\dddot{r}_i\}^{Pk} \left(\{r_i\}^{Pk} - \{r_i\}^{mean} \right)^T \right) / \sum_{k=1}^{n} f^{Pk}$$

$$[H] = [J] - \left(2[\tilde{\dot{\omega}}_i][\tilde{\omega}_i] + [\tilde{\omega}_i][\tilde{\dot{\omega}}_i] + [\tilde{\omega}_i][\tilde{\omega}_i][\tilde{\omega}_i] \right)[X]$$

$$\{\ddot{\omega}_i\} = [M]^{-1} \begin{Bmatrix} H_{32} - H_{23} \\ H_{13} - H_{31} \\ H_{21} - H_{12} \end{Bmatrix} = \begin{Bmatrix} \ddot{\omega}_x \\ \ddot{\omega}_y \\ \ddot{\omega}_z \end{Bmatrix} \qquad \ddot{\omega}_i = \text{norm}\{\ddot{\omega}_i\}$$

$$[\tilde{\ddot{\omega}}_i] = \begin{bmatrix} 0 & -\ddot{\omega}_z & \ddot{\omega}_y \\ \ddot{\omega}_z & 0 & -\ddot{\omega}_x \\ -\ddot{\omega}_y & \ddot{\omega}_x & 0 \end{bmatrix} \qquad [\eta_i] = [\tilde{\ddot{\omega}}_i] + 2[\tilde{\dot{\omega}}_i][\tilde{\omega}_i] + [\tilde{\omega}_i][\tilde{\dot{\omega}}_i] + [\tilde{\omega}_i][\tilde{\omega}_i][\tilde{\omega}_i]$$

Point IJP is the instantaneous jerk pole for body i. Note that the IJP is not attached to the body. Point P on the body coincident with IJP has zero jerk.

$$\{\dddot{r}_i\}^P = [\eta_i]\left(\{r_i\}^P - \{r\}^{IJP} \right) \quad \text{for any point P attached to body i}$$

$$\{r\}^{IJP} = \{r_i\}^{mean} - [\eta_i]^{-1}\{\dddot{r}_i\}^{mean} \quad \text{for} \quad \{\dddot{r}_i\}^{P\_at\_IJP} = 0$$

Second Order Screw Axis

$$\{\Omega\} = \begin{Bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{Bmatrix} = [\tilde{\omega}]\{\dot{\omega}\} / \omega^2 \qquad \Omega = \text{norm}\{\Omega\} \qquad \{\hat{t}\} = \{\Omega\} / \Omega$$

$$d = \left(\{\hat{t}\}^T [\beta]\left(\{r\}^{IAP} - \{r\}^{IHA} \right) \right) / \left(\{\hat{t}\}^T [\beta]\{\hat{u}\} \right)$$

$$\{c\} = \{r\}^{IHA} + d\{\hat{u}\}$$

$$\dot{S} = \left(\{\hat{t}\}^T [\tilde{\omega}][\beta]\left(\{c\} - \{r\}^{IAP} \right) / \omega^2 \right) - \dot{s}\Omega / \omega$$

```
% t_lm2kin3d.m - test 3D kinematics from landmark motion
% HJSIII, 14.01.14

clear

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% example data - CRSP
f = [      1        1        1        1        1        1        1        1     ];

r = [     11.000    9.000    9.000   11.000   11.000    9.000    9.000   11.000 ;
           1.000    1.000   -1.000   -1.000    1.000    1.000   -1.000   -1.000 ;
          10.000   10.000   10.000   10.000    8.000    8.000    8.000    8.000 ];

rd = [   -35.750  -35.750  -30.750  -30.750  -28.750  -28.750  -23.750  -23.750 ;
          27.500   22.500   22.500   27.500   27.500   22.500   22.500   27.500 ;
           8.000    1.000    1.000    8.000    8.000    1.000    1.000    8.000 ];

rdd = [  -45.250   -8.250  -19.250  -56.250  -58.250  -21.250  -32.250  -69.250 ;
        -233.000 -222.000 -209.500 -220.500 -198.000 -187.000 -174.500 -185.500 ;
        -130.375 -117.375 -117.375 -130.375 -105.875  -92.875  -92.875 -105.875 ];

rddd = [   0        0        0        0        0        0        0        0     ;
           0        0        0        0        0        0        0        0     ;
           0        0        0        0        0        0        0        0     ];

vel_test   = [        0    5.7703         0 ;
                -3.5000   -4.0203   14.4257 ;
                 2.5000    3.3716  -10.3041 ];

accel_test = [   8.7500    2.6440         0 ;
                 6.5000  -11.5414         0 ;
                -5.5000    3.7909         0 ];

jerk_test  = [   1.5000   10.0000         0 ;
               -53.8125         0         0 ;
                61.5625    9.0000         0 ];

axode_test = [   0.1622    5.7703    0.3815 ;
                 1.1824    0.3172    2.7819 ;
                 1.6554    0.2734    3.8947 ];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% example data - RSUR
f = [      1         1         1         1         1         1      ];

r = [     -9.8091   -6.7903   -4.2819   -4.7924   -7.8113  -10.3196 ;
         -15.4760   -8.0967   -3.7733   -6.8291  -14.2084  -18.5319 ;
          -1.9764   -0.7795   -7.0258  -14.4690  -15.6659   -9.4196 ];

rd = [     1.8124   -0.4639    7.5799   17.9000   20.1763   12.1325 ;
           0.1709    0.3955    0.8591    1.0982    0.8737    0.4100 ;
         -14.0582   -9.7011   -6.1500   -6.9559  -11.3130  -14.8641 ];

rdd = [  -40.6381 -115.2233 -105.8619  -21.9154   52.6698   43.3084 ;
          15.3096   35.2375   76.6641   98.1628   78.2349   36.8083 ;
        -148.5442 -103.5193  -58.6743  -58.8541 -103.8790 -148.7241 ];

rddd = [ 596.0567  523.6492    4.6583 -441.9252 -369.5177  149.4732 ;
           3.0571  -41.6333  -59.9366  -33.5496   11.1408   29.4441 ;
         435.0585  -35.2544 -134.4616  236.6441  706.9570  806.1642 ];

vel_test   = [   0.0383    0.4123   -0.0297 ;
                -1.3498  -10.4511    1.0468 ;
                 0.0895   -0.9450   -0.0694 ];

accel_test = [   3.3113  -52.7472         0 ;
                -7.8355   81.8870         0 ;
                 8.0368  -97.6836         0 ];

jerk_test  = [ -20.3604  -22.3536         0 ;
                61.6042   32.7874         0 ;
```

```
                 1.5851  -12.8695         0 ];

axode_test = [  -5.5403    0.6887   53.3304 ;
                -0.0060  -20.2001    0.0579 ;
                 2.2768   -0.2983  -21.9164 ];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% test function
[ vel, accel, jerk, axode ] = lm2kin3d( f, r, rd, rdd, rddd )

% bottom of t_lm2kin3d
```

```
function [ vel, accel, jerk, axode ] = lm2kin3d( f, r, rd, rdd, rddd )
% 3D kinematics of a rigid body from landmark motion
% HJSIII, 14.01.14
%
% USAGE
% [ vel, accel, jerk, axode ] = lm2kin3d( f, r, rd, rdd, rddd )
%
% INPUTS
% f    - 1 x n vector of weights - f(j)=1 means data valid, f(j)=0 means data not available
% r    - 3 x n matrix of x,y,z landmark location
% rd   - 3 x n matrix of x,y,z landmark velocity
% rdd  - 3 x n matrix of x,y,z landmark acceleration
% rddd - 3 x n matrix of x,y,z landmark jerk
%
% OUTPUTS
% vel   = [ w_vec  rISA  sISA ]
%           w_vec = 3x1 angular velocity vector
%           rISA  = 3x1 location on ISA at root of perpendicular from centroid of landmarks
%           sISA  = sliding velocity vector along ISA
%
% accel = [ wd_vec  rIAP  rdd_at_IAP ]
%           wd_vec     = 3x1 angular acceleration vector
%           rIAP       = 3x1 location of acceleration pole
%           rdd_at_IAP = 3x1 acceleration of point on body at IAP
%
% jerk  = [ wdd_vec  rIJP  rddd_at_IJP ]
%           wdd_vec     = 3x1 angular jerk vector
%           rIJP        = 3x1 location of jerk pole
%           rddd_at_IJP = 3x1 jerk of point on body at IJP
%
% axode = [ OMEGA_vec  c  Sd ]
%           OMEGA_vec = 3x1 rotation of second order screw
%           c         = 3x1 central point of generator
%           Sd        = 3x1 sliding velocity vector along second order screw

% constants
eps = 1e-14;

% number of coordinates and landmarks
[ ncoord, n ] = size( r );

% mean values
fmat = diag(f);
sf = trace( fmat );
rm = sum( fmat*r' )'       /sf;
rdm = sum( fmat*rd' )'     /sf;
rddm = sum( fmat*rdd' )'   /sf;
rdddm = sum( fmat*rddd' )' /sf;

% centered location
rc = r - rm*ones(1,n);
X = rc * fmat * rc' /sf;
M = trace(X) * eye(ncoord) - X;
Minv = inv( M );

% velocity
V = rd * fmat * rc' /sf;
w_vec = Minv * [ V(3,2)-V(2,3) ; V(1,3)-V(3,1) ; V(2,1)-V(1,2) ];
w = norm( w_vec );
w_mat = skew_sym( w_vec );

% general velocity solution
if w > eps,
  u = w_vec / w;
  sd = u' * rdm;
  rISA = rm + w_mat * rdm / w^2;
  sISA = sd * u;

% special case - w=0, pure translation
% rISA is at centroid of landmarks, sISA is translation velocity
else
  sd = norm( rdm );
  u = rdm / sd;
```

```
  rISA = rm;
  sISA = rdm;
end

% acceleration
A = rdd * fmat * rc' / sf;
B = A - w_mat*w_mat * X;
wd_vec = Minv * [ B(3,2)-B(2,3) ; B(1,3)-B(3,1) ; B(2,1)-B(1,2) ];
wd = norm( wd_vec );
wd_mat = skew_sym( wd_vec );
beta_mat = wd_mat + w_mat*w_mat;

% general acceleration solution
if abs(det(beta_mat)) > eps;
  rIAP = rm - inv(beta_mat) * rddm;
  rdd_at_IAP = zeros(ncoord,1);
else

% special case 1 - w=0, wd=0, pure translation
% rIAP is at centroid of landmarks, rdd_at_IAP is translation acceleration
  if w < eps,
    if wd < eps,
      sdd = norm( rddm );
      e = rddm / sdd;
      rIAP = rm;
      rdd_at_IAP = rddm;

% special case 2 - w=0, wd>0, pure angular acceleration
%                  similar to general angular velocity solution
% rIAP is at root of perpendicular to angular acceleration vector from centroid of landmarks
% rdd_at_IAP is translation acceleration
    else
       e = wd_vec / wd;
       sdd = e' * rddm;
       rIAP = rm + wd_mat * rddm / wd^2;
       rdd_at_IAP = sdd * e;
    end

% special case 3 - w>0, wd=0, pure angular velocity
%                  similar to zero angular velocity solution
  else
    if wd < eps,
      e = u;
      sdd = e' * rddm;
      rdd_at_IAP = sdd * e;
      rIAP = rm + (rddm-rdd_at_IAP) / w*w;

% special case 4 - w>0, wd>0, w_vec parallel to wd_vec
    else
      e = wd_vec / wd;
      sdd = e' * rddm;
      rdd_at_IAP = sd * e;

      w2ia = w*w*eye(3) - wd_mat;
      rIAP = rm + inv(w2ia) * (rddm-rdd_at_IAP);

    end
  end
end

% jerk
J = rddd * fmat * rc' / sf;
eta_mat_mwdd = 2*wd_mat*w_mat + w_mat*wd_mat + w_mat*w_mat*w_mat;
H = J - eta_mat_mwdd * X;
wdd_vec = Minv * [ H(3,2)-H(2,3) ; H(1,3)-H(3,1) ; H(2,1)-H(1,2) ];
wdd = norm( wdd_vec );
h = wdd_vec / wdd;
wdd_mat = skew_sym( wdd_vec );
eta_mat = eta_mat_mwdd + wdd_mat;
rIJP = rm - inv(eta_mat) * rdddm;
rddd_at_IJP = zeros(ncoord,1);

% second order screw
```

```
OMEGA_vec = w_mat * wd_vec /w/w;
OMEGA = norm( OMEGA_vec );
t = OMEGA_vec / OMEGA;
d = t' * beta_mat * (rIAP-rISA) / (t' * beta_mat * u );
c = rISA + d * u;
Sd = ( t' * w_mat * beta_mat * (c-rIAP) /w/w ) - sd * OMEGA /w;
Sd_vec = Sd * t;

% return arguments
vel   = [ w_vec  rISA  sISA  ];
accel = [ wd_vec  rIAP  rdd_at_IAP  ];
jerk  = [ wdd_vec  rIJP  rddd_at_IJP  ];
axode = [ OMEGA_vec  c  Sd_vec ];

% bottom of lm2kin3d
```