

1) Local coordinates for vertices on bodies 2 and 3 are provided below in CW closed boundary chains. Time samples for position and attitude of each body are provided as data in the attached MATLAB code. A sample MATLAB plot of motion for body 2 is shown below.

$$\{s_2\}^{t,ALL} = \left[\begin{bmatrix} 2 \\ 3 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \end{bmatrix} \begin{bmatrix} -1 \\ -2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right] \quad \{s_3\}^{t,ALL} = \left[\begin{bmatrix} 2 \\ 3 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right]$$

2) Run the MATLAB code and then modify it to additionally show motion of body 3 on the same MATLAB plot. Manually assess when collision occurs based on the plot and record in the table below. Attach hardcopy of your MATLAB plot and code.

3) Use the bounding circles test to determine when the objects would be candidates for collision detection and record your results below. Attach copy of your code.

4) Use the point-in-polygon test with MATLAB “inpolygon” to detect collision and record your results below. Attach copy of your code.

5) Use the edge intersection test with MATLAB “polyxpoly” to detect collision and record your results below. For example, if edge 4 on body 2 intersects edge 1 on body 3, record E4₂/E1₃. Attach copy of your code.

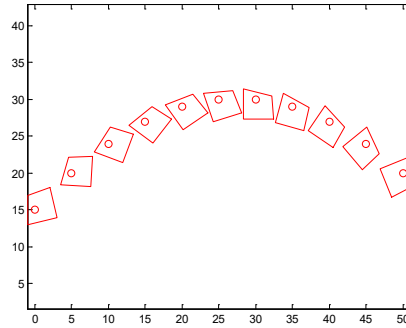
Time [sec]	5	6	7	8	9	10
Manual C = collision blank = no			C	C		
Bounding circles C = candidates blank = no			C	C		
Point-in-polygon Which vertices on body 2 are inside polygon 3 ?				V3		
Point-in-polygon Which vertices on body 3 are inside polygon 2 ?			V1,V4			
Edge intersection Which edges on 2 intersect which edges on 3 ?			E1 ₂ /E1 ₃ E1 ₂ /E3 ₃	E2 ₂ /E2 ₃ E3 ₂ /E2 ₃		

EXTRA CREDIT

Use Savitsky-Golay interpolants to compute the time for initial collision.

constant vel interpolation t = 6.9783 sec

constant accel interpolation t = 6.9000 sec



```

% show_body2.m - show motion for collision detection
% HJSIII, 19.04.24

clear

% motion data
%   time      x2      y2      phi2      x3      y3      phi3
%   [sec]     [in]    [in]    [rad]    [in]    [in]    [rad]
all = [ 0      0      15.0000    0      0      0      0 ;
       1.0000  5.0000  20.0000  -0.3000  6.6000  0      0.2000 ;
       2.0000  10.0000 24.0000  -0.6000  13.0684  1.3112  0.4000 ;
       3.0000  15.0000 27.0000  -0.9000  19.1474  3.8814  0.6000 ;
       4.0000  20.0000 29.0000  -1.2000  24.5947  7.6080  0.8000 ;
       5.0000  25.0000 30.0000  -1.5000  29.1929  12.3426  1.0000 ;
       6.0000  30.0000 30.0000  -1.8000  32.7589  17.8963  1.2000 ;
       7.0000  35.0000 29.0000  -2.1000  35.1505  24.0477  1.0000 ;
       8.0000  40.0000 27.0000  -2.4000  38.7165  29.6014  0.8000 ;
       9.0000  45.0000 24.0000  -2.7000  43.3147  34.3360  0.6000 ;
      10.0000  50.0000 20.0000  -3.0000  48.7620  38.0626  0.4000 ];
time = all(:,1)'; % size 1 x nt
r2_all = all(:,2:3)'; % size 2 x nt
phi2_all = all(:,4)'; % size 1 x nt
ntime = length( time );

% define object 2 in local coordinates
s2p_poly = [ 2  3 -1 -1  2 ; % local x2p [in]
            3 -1 -2  2  3 ]; % local y2p [in]
rho2 = max( sqrt( diag( s2p_poly'*s2p_poly ) ) ); % maximum radius
[ nr, n2 ] = size( s2p_poly ); % number of points for body 2

% new figure
figure( 1 )
clf

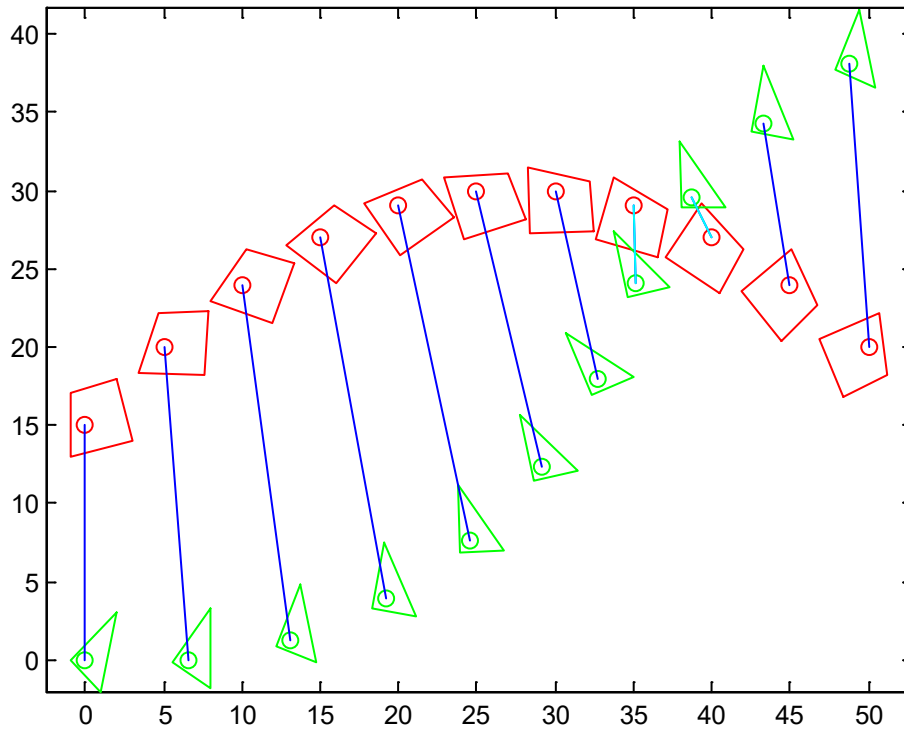
% plot origin and outline at each time sample
for itime = 1 : ntime,
    t = time(itime);

% body 2 = red
    r2 = r2_all(:,itime); % origin
    phi2 = phi2_all(itime); % angle
    A2 = [ cos(phi2)  -sin(phi2) ; % attitude matrix
          sin(phi2)   cos(phi2) ];
    r2_poly = r2*ones(1,n2) + A2*s2p_poly; % global locations for vertices
    plot( r2(1),r2(2),'ro' ) % plot origin
    axis equal
    hold on
    plot( r2_poly(1,:), r2_poly(2,:), 'r' ) % closed curve

end % bottom - for itime

% bottom - show_body2.m

```



```
>> show_collision
```

```
Time 7.0000
Bounding circles intersect
Vertex 1 on body 3 is inside body 2
Vertex 4 on body 3 is inside body 2
Edge 1 on body 2 intersects edge 1 on body 3
Edge 1 on body 2 intersects edge 3 on body 3
```

```
Time 8.0000
Bounding circles intersect
Vertex 3 on body 2 is inside body 3
Edge 2 on body 2 intersects edge 2 on body 3
Edge 3 on body 2 intersects edge 2 on body 3
```

```
dt_cv =
    0.9783
```

```
penetration_cv =
    2.8422e-14
```

```
dt_ca_all =
   -9.1757
   -6.8411
    4.5817
    0.9000
```

```
penetration_ca =
   -0.0011
```

```

% show_collision.m - show motion for collision detection
% HJSIII, 19.04.23

clear

% motion data
%   time      x2      y2      phi2      x3      y3      phi3
%   [sec]     [in]     [in]     [rad]     [in]     [in]     [rad]
all = [ 0      0      15.0000      0      0      0      0 ;
       1.0000  5.0000  20.0000  -0.3000  6.6000  0      0.2000 ;
       2.0000 10.0000  24.0000  -0.6000 13.0684  1.3112  0.4000 ;
       3.0000 15.0000  27.0000  -0.9000 19.1474  3.8814  0.6000 ;
       4.0000 20.0000  29.0000  -1.2000 24.5947  7.6080  0.8000 ;
       5.0000 25.0000  30.0000  -1.5000 29.1929 12.3426  1.0000 ;
       6.0000 30.0000  30.0000  -1.8000 32.7589 17.8963  1.2000 ;
       7.0000 35.0000  29.0000  -2.1000 35.1505 24.0477  1.0000 ;
       8.0000 40.0000  27.0000  -2.4000 38.7165 29.6014  0.8000 ;
       9.0000 45.0000  24.0000  -2.7000 43.3147 34.3360  0.6000 ;
      10.0000 50.0000  20.0000  -3.0000 48.7620 38.0626  0.4000 ];
time = all(:,1)'; % size 1 x nt
r2_all = all(:,2:3)'; % size 2 x nt
phi2_all = all(:,4)'; % size 1 x nt
r3_all = all(:,5:6)'; % size 2 x nt
phi3_all = all(:,7)'; % size 1 x nt
ntime = length( time );

% define object 2 in local coordinates
s2p_poly = [ 2  3 -1 -1 2 ; % local x2p [in]
            3 -1 -2  2 3 ]; % local y2p [in]
rho2 = max( sqrt( diag( s2p_poly'*s2p_poly ) ) ); % maximum radius
[ nr, n2 ] = size( s2p_poly ); % number of points for body 2

% define object 3 in local coordinates
s3p_poly = [ 2  1 -1 2 ; % local x3p [in]
            3 -2  0 3 ]; % local y3p [in]
rho3 = max( sqrt( diag( s3p_poly'*s3p_poly ) ) ); % maximum radius
[ nr, n3 ] = size( s3p_poly ); % number of points for body 3

% new figure
figure( 1 )
clf

% plot origins and outlines at each time sample
for itime = 1 : ntime,
    t = time(itime);

% body 2 = red
    r2 = r2_all(:,itime); % origin
    phi2 = phi2_all(itime); % angle
    A2 = [ cos(phi2)  -sin(phi2) ; % attitude matrix
          sin(phi2)   cos(phi2) ];
    r2_poly = r2*ones(1,n2) + A2*s2p_poly; % global locations for vertices
    plot( r2(1),r2(2),'ro' ) % plot origin
    axis equal
    hold on
    plot( r2_poly(1,:), r2_poly(2,:), 'r' ) % closed curve

% body 3 = green
    r3 = r3_all(:,itime);
    phi3 = phi3_all(itime);
    A3 = [ cos(phi3)  -sin(phi3) ;
          sin(phi3)   cos(phi3) ];
    r3_poly = r3*ones(1,n3) + A3*s3p_poly;
    plot( r3(1),r3(2),'go' ) % plot origin
    plot( r3_poly(1,:), r3_poly(2,:), 'g' ) % closed curve

% line between centers = blue
    plot( [ r2(1)  r3(1) ], [ r2(2)  r3(2) ], 'b' )

% bounding circles ++++++
% check if bounding circles (BC) intersect

```

```

d23 = norm( r3 - r2 );
if d23 < (rho2+rho3),
    plot( [ r2(1)  r3(1) ], [ r2(2)  r3(2) ], 'c' ) % show centerline = cyan

% show time
disp( ' ' )
disp( sprintf( 'Time %7.4f', t ) )
disp( 'Bounding circles intersect' )

% point on 3 inside polygon 2 ++++++
% check if vertices for body 3 are inside body 2
inside2 = inpolygon( r3_poly(1,:),r3_poly(2,:), r2_poly(1,:),r2_poly(2,:) );
i3_inside2 = find( inside2 );
if ~isempty( i3_inside2 ),

% show vertices for body 3 that are inside body 2
n3_inside2 = length( i3_inside2 );
for k = 1 : n3_inside2,
    i3 = i3_inside2( k ); % vertex on 3
    disp( sprintf( 'Vertex %d on body 3 is inside body 2', i3 ) )
end % bottom - for k
end % bottom - ~isempty

% point on 2 inside polygon 3 ++++++
% check if vertices for body 2 are inside body 3
inside3 = inpolygon( r2_poly(1,:),r2_poly(2,:), r3_poly(1,:),r3_poly(2,:) );
i2_inside3 = find( inside3 );
if ~isempty( i2_inside3 ),

% show vertices for body 3 that are inside body 2
n2_inside3 = length( i2_inside3 );
for k = 1 : n2_inside3,
    i2 = i2_inside3( k ); % vertex on 2
    disp( sprintf( 'Vertex %d on body 2 is inside body 3', i2 ) )
end % bottom - for k
end % bottom - if ~isempty

% edge intersection - ++++++
[ xi, yi, ii ] = polyxpoly( r2_poly(1,:), r2_poly(2,:), r3_poly(1,:), r3_poly(2,:) );
[ n_ii, nc ] = size( ii );
if ( n_ii > 0 )
    for i = 1 : n_ii,
        disp( sprintf( 'Edge %d on body 2 intersects edge %d on body 3', ii(i,1), ii(i,2) ) )
    end
end

% bottom if bounding circles overlap
end % bottom - if c23 - BC ++++++

% save global trajectories ++++++
r2_poly_time(:, :, itime) = r2_poly;
r3_poly_time(:, :, itime) = r3_poly;

end % bottom - for itime ++++++

% extract trajectories for specific vertices
% vertex 1 on body 3 (r3P) pierces edge 1 on body 2 (r2P to r2Q)
iP3 = 1;
r3P_time = squeeze( r3_poly_time(:, iP3, :) );

iP2 = 1;
r2P_time = squeeze( r2_poly_time(:, iP2, :) );
iQ2 = 2;
r2Q_time = squeeze( r2_poly_time(:, iQ2, :) );

% velocities and accelerations over all time by Savitsky-Golay interpolants
h = 1;
[ p3P, v3P, a3P, j3P ] = filt_7pt_mat( r3P_time, h );
[ p2P, v2P, a2P, j2P ] = filt_7pt_mat( r2P_time, h );
[ p2Q, v2Q, a2Q, j2Q ] = filt_7pt_mat( r2Q_time, h );

```

```

% collision at t=7 sec and t=8 sec
% position, velocity and acceleration at t=6 sec just before collision
itime = 7;           % provides t=6 sec
itime = 8;           % provides t=7 sec
r3P   = p3P(:,itime);
r3Pd  = v3P(:,itime);
r3Pdd = a3P(:,itime);

r2P   = p2P(:,itime);
r2Pd  = v2P(:,itime);
r2Pdd = a2P(:,itime);

r2Q   = p2Q(:,itime);
r2Qd  = v2Q(:,itime);
r2Qdd = a2Q(:,itime);

% compute time for collision assuming constant velocity
R = [ 0 -1 ; % rotator matrix
      1  0 ];
a = (r3Pd-r2Pd)*R*(r2Qd-r2Pd);
b = (r3P-r2P)*R*(r2Qd-r2Pd) + (r3Pd-r2Pd)*R*(r2Q-r2P);
c = (r3P-r2P)*R*(r2Q-r2P);
dt_cv = (-b -sqrt(b*b - 4*a*c)) / (2*a)

r3P_pred = r3P + r3Pd*dt_cv;
r2P_pred = r2P + r2Pd*dt_cv;
r2Q_pred = r2Q + r2Qd*dt_cv;
penetration_cv = (r3P_pred-r2P_pred)*R*(r2Q_pred-r2P_pred)

% predict using constant acceleration
c4 = (r3Pdd-r2Pdd)*R*(r2Qdd-r2Pdd)/4;
c3 = (r3Pd-r2Pd)*R*(r2Qdd-r2Pdd)/2 + (r3Pdd-r2Pdd)*R*(r2Qd-r2Pd)/2;
c2 = (r3P-r2P)*R*(r2Qdd-r2Pdd)/2 + (r3Pd-r2Pd)*R*(r2Qd-r2Pd) + ...
      (r3Pdd-r2Pdd)*R*(r2Q-r2P)/2;
c1 = (r3P-r2P)*R*(r2Qd-r2Pd) + (r3Pd-r2Pd)*R*(r2Q-r2P);
c0 = (r3P-r2P)*R*(r2Q-r2P);
p = [ c4 c3 c2 c1 c0 ];
dt_ca_all = roots( p )

dt_ca = 0.90;
r3P_pred = r3P + r3Pd*dt_ca + 0.5*r3Pdd*dt_ca*dt_ca;
r2P_pred = r2P + r2Pd*dt_ca + 0.5*r2Pdd*dt_ca*dt_ca;
r2Q_pred = r2Q + r2Qd*dt_ca + 0.5*r2Qdd*dt_ca*dt_ca;
penetration_ca = (r3P_pred-r2P_pred)*R*(r2Q_pred-r2P_pred)

% bottom - show_collision.m

```