**Two-dimensional autofill constraint vector and Jacobian**

1) Provide an initialization file (xxx\_ini.m) that contains -

a) initial estimates for generalized coordinates 

b) local body-fixed coordinates  of points used to create joints,

c) constants for joints and drivers, and

d) a table describing constraints using entries listed in Table 1.

Constraints in Table 1 correspond to mathematical constraints in Notes\_04\_04. Constraints may be listed in any order.

Bodies must be numbered sequentially starting at 1. Body 1 is fixed at the global origin.

The mechanism may be single loop or multiple loop, may be open loop or closed loop, and must be kinematically driven (nc=nq).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **constraint** | **col 1**  **ctype** | **col 2**  **body i** | **col 3**  **body j** | **col**  **4,5** | **col**  **6,7** | **col**  **8,9** | **col**  **10,11** |
| revolute | 1 | i | j |  |  | 0 0 | 0 0 |
| double revolute | 2 | i | j |  |  | L 0 | 0 0 |
| parallel  vectors | 3 | i | j |  |  |  |  |
| pin-in-slot | 4 | i | j |  |  |  | 0 0 |
| acceleration  angle  driver | 5 | i | j | C v | a 0 | 0 0 | 0 0 |

**Table 1 - Kinematic and driver constraint descriptors**

**acceleration angle driver = **

**Four bar**

**A**

**B**

**D**

**C**

**4**

**3**

**2**

**driver**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **col 1**  **ctype** | **col 2**  **body i** | **col 3**  **body j** | **col**  **4,5** | **col**  **6,7** | **col**  **8,9** | **col**  **10,11** | **NOTES** |
| 1  rev | 1 | 2 |  |  | 0 0 | 0 0 | r2A-r1A |
| 1  rev | 2 | 3 |  |  | 0 0 | 0 0 | r3B-r2B |
| 1  rev | 3 | 4 |  |  | 0 0 | 0 0 | r4C-r3C |
| 1  rev | 1 | 4 |  |  | 0 0 | 0 0 | r4D-r1D |
| 5  ang dri | 1 | 2 |  | 0 0 | 0 0 | 0 0 |  |

**acceleration angle driver = **

% fbq9\_ini.m - four bar nq=9 for autofill (web cutter)

% constants, assembly estimates and constraint table

% HJSIII, 20.02.28

% mechanism constants

lenAB = 4;

lenBC = 14.23;

lenCD = 20.32;

s1pA = [ 0 0 ]';

s1pD = [ 13.21 -2.03 ]';

s2pA = [ 0 0 ]';

s2pB = [ lenAB 0 ]';

s3pB = [ 0 0 ]';

s3pC = [ lenBC 0 ]';

s4pC = [ 0 0 ]';

s4pD = [ lenCD 0 ]';

% initial estimate for phi3 = 30 deg

phi2 = 30 \* d2r;

phi3 = 90 \* d2r;

phi4 = -60 \* d2r;

q = [ 0 0 phi2 3.5 2 phi3 3.5 16.23 phi4 ]';

% driver for crank - phi2 - phi2\_start - w2\*t

%phi2\_start = 30 \* d2r;

phi2\_start = 0;

w2 = +60 \* 2 \* pi / 60; % 60 rpm CCW, convert to rad/sec

% constraint entries

% ctype=1 revolute [ ctype body\_i body\_j sipP' sjpP' [0 0] [0 0] ]

% ctype=2 double revolute [ ctype body\_i body\_j sipP' sjpP' L 0 [0 0] ]

% ctype=3 parallel vectors [ ctype body\_i body\_j sipP' sipQ' sjpP' sjpQ' ]

% ctype=4 pin-in-slot [ ctype body\_i body\_j sipP' sipQ' sjpP' [0 0] ]

% ctype=5 relative angle driver [ ctype body\_i body\_j C v a 0 [0 0] [0 0] ]

constraints = [ 1 1 2 s1pA' s2pA' [0 0] [0 0] ; % rev r2A-r1A

1 2 3 s2pB' s3pB' [0 0] [0 0] ; % rev r3B-r2B

1 3 4 s3pC' s4pC' [0 0] [0 0] ; % rev r4C-r3C

1 1 4 s1pD' s4pD' [0 0] [0 0] ; % rev r4D-r1D

5 1 2 phi2\_start w2 0 0 [0 0] [0 0] ]; % phi2-phi1-phi2\_start-w2\*t

% bottom - fbq9\_ini

**Four bar with double revolute BC (note: body 3 is link CD)**

**A**

**B**

**D**

**C**

**3**

**2**

**driver**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **col 1**  **ctype** | **col 2**  **body i** | **col 3**  **body j** | **col**  **4,5** | **col**  **6,7** | **col**  **8,9** | **col 10,11** | **NOTES** |
| 1  rev | 1 | 2 |  |  | 0 0 | 0 0 | r2A-r1A |
| 2  dbl rev | 2 | 3 |  |  | lenBC 0 | 0 0 | r3C-r2B  lenBC |
| 1  rev | 1 | 3 |  |  | 0 0 | 0 0 | r3D-r1D |
| 5  ang dri | 1 | 2 |  | 0 0 | 0 0 | 0 0 |  |

**acceleration angle driver = **

% fbq6\_ini.m - four bar nq=6 for autofill (web cutter with double revolute)

% constants, assembly estimates and constraint table

% HJSIII, 20.02.28

% mechanism constants

lenAB = 4;

lenBC = 14.23;

lenCD = 20.32;

s1pA = [ 0 0 ]';

s1pD = [ 13.21 -2.03 ]';

s2pA = [ 0 0 ]';

s2pB = [ lenAB 0 ]';

s3pC = [ 0 0 ]'; % NOTE: link CD is called body 3

s3pD = [ lenCD 0 ]';

% initial estimates for phi2 = 30 deg

phi2 = 30 \* d2r;

phi3 = -60 \* d2r; % NOTE: link CD is called body 3

q = [ 0 0 phi2 3.5 16.23 phi3 ]';

% driver for crank - phi2 - phi2\_start - w2\*t

%phi2\_start = 30 \* d2r;

phi2\_start = 0;

w2 = +60 \* 2 \* pi / 60; % 60 rpm CCW, convert to rad/sec

% constraint entries

% ctype=1 revolute [ ctype body\_i body\_j sipP' sjpP' [0 0] [0 0] ]

% ctype=2 double revolute [ ctype body\_i body\_j sipP' sjpP' L 0 [0 0] ]

% ctype=3 parallel vectors [ ctype body\_i body\_j sipP' sipQ' sjpP' sjpQ' ]

% ctype=4 pin-in-slot [ ctype body\_i body\_j sipP' sipQ' sjpP' [0 0] ]

% ctype=5 relative angle driver [ ctype body\_i body\_j C v a 0 [0 0] [0 0] ]

constraints = [ 1 1 2 s1pA' s2pA' [0 0] [0 0] ; % rev r2A-r1A

2 2 3 s2pB' s3pC' lenBC 0 [0 0] ; % dbl rev r3C-r3B lenBC

1 1 3 s1pD' s3pD' [0 0] [0 0] ; % rev r3D-r1D

5 1 2 phi2\_start w2 0 0 [0 0] [0 0] ]; % phi2-phi1-phi2\_start-w2\*t

% bottom - fbq6\_ini

**Inverted slider crank**

**A**

**B**

**C**

**4**

**3**

**2**

**driver**

**Q4**

**B3**

**Q3**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **col 1**  **ctype** | **col 2**  **body i** | **col 3**  **body j** | **col**  **4,5** | **col**  **6,7** | **col**  **8,9** | **col**  **10,11** | **NOTES** |
| 1  rev | 1 | 2 |  |  | 0 0 | 0 0 | r2A-r1A |
| 1  rev | 2 | 3 |  |  | 0 0 | 0 0 | r3B-r2B |
| 3  par vec | 4 | 3 |  |  |  |  | r4Q-r4C  r3Q-r3B |
| 4  pin slot | 4 | 3 |  |  |  | 0 0 | r4C-r3C  r3B |
| 1  rev | 1 | 4 |  |  | 0 0 | 0 0 | r4C-r1C |
| 5  ang dri | 1 | 2 |  | 0 0 | 0 0 | 0 0 |  |

**acceleration angle driver = **

% iscq9\_ini.m - inverted slider crank nq=9 for autofill

% constants, assembly estimates and constraint table

% HJSIII, 20.02.28

% mechanism constants

lenAB = 2; % [cm]

lenB3Q3 = 0.5; % [cm]

lenAC = 7; % [cm]

lenC4Q4 = 10; % [cm]

s1pA = [ lenAC 0 ]';

s1pC = [ 0 0 ]';

s2pA = [ 0 0 ]';

s2pB = [ lenAB 0 ]';

s3pB = [ 0 0 ]';

s3pQ = [ lenB3Q3 0 ]';

s4pC = [ 0 0 ]';

s4pQ = [ lenC4Q4 0 ]';

% initial estimates for phi2 = 90 deg

phi2 = 90 \* d2r;

phi3 = 16 \* d2r;

phi4 = 16 \* d2r;

q = [ lenAC 0 phi2 lenAC lenAB phi3 0 0 phi4 ]';

% driver for crank - phi2 - phi2\_start - w2\*t

%phi2\_start = 90 \* d2r;

phi2\_start = 0;

w2 = +60 \* 2 \* pi / 60; % 60 rpm CCW, convert to rad/sec

% constraint entries

% ctype=1 revolute [ ctype body\_i body\_j sipP' sjpP' [0 0] [0 0] ]

% ctype=2 double revolute [ ctype body\_i body\_j sipP' sjpP' L 0 [0 0] ]

% ctype=3 parallel vectors [ ctype body\_i body\_j sipP' sipQ' sjpP' sjpQ' ]

% ctype=4 pin-in-slot [ ctype body\_i body\_j sipP' sipQ' sjpP' [0 0] ]

% ctype=5 relative angle driver [ ctype body\_i body\_j C v a 0 [0 0] [0 0] ]

constraints = [ 1 1 2 s1pA' s2pA' [0 0] [0 0] ; % rev r2A-r1A

1 2 3 s2pB' s3pB' [0 0] [0 0] ; % rev r3B-r2B

3 4 3 s4pC' s4pQ' s3pB' s3pQ' ; % par vec r4Q-r4C // r3Q-r3B

4 4 3 s4pC' s4pQ' s3pB' [0 0] ; % pin-in-slot r4Q-r4C guides r3B

1 1 4 s1pC' s4pC' [0 0] [0 0] ; % rev r4C-r1C

5 1 2 phi2\_start w2 0 0 [0 0] [0 0] ]; % phi2-phi1-phi2\_start-w2\*t

% bottom of iscq9\_ini.m

**Wanzer needle bar**

E

F

G

A

2

3

4

5

6

slot B

slot C

slot D

EF = 16 mm

FG = 56 mm

EG = 62 mm

r2 = 31 mm radius of disk 2

eB = 10 mm radial offset for slot B

eC = 10 mm radial offset for slot C

BC = 90º between slots B and C

w = 5 mm width of slots B, C, and D

(FG)2 = (EF)2 + (EG)2 – 2 (EF) (EG) cos 

 = 60.93°

2 = -25.133 rad/sec

E = 0, 0



F = 13.984, -7.774

G = 0, -62

**no sliders - only pins-in-slots - nb=3, nq=6, nc=6**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **col 1**  **ctype** | **col 2**  **body i** | **col 3**  **body j** | **col**  **4,5** | **col**  **6,7** | **col**  **8,9** | **col**  **10,11** | **NOTES** |
| 1  rev | 1 | 2 |  |  | 0 0 | 0 0 | r2A-r1A |
| 4  pin slot | 2 | 3 |  |  |  | 0 0 | r2BQ-r2BP  r3E |
| 4  pin slot | 2 | 3 |  |  |  | 0 0 | r2CQ-r2CP  r3F |
| 4  pin slot | 1 | 3 |  |  |  | 0 0 | r1D-r1A  r3G |
| 5  ang dri | 1 | 2 |  | 0 0 | 0 0 | 0 0 |  |

**acceleration angle driver = **

% wanzerq6\_ini.m - Wanzer sewing machine nq=6 for autofill - no blocks, pins-in-slots

% constants, assembly estimates and constraint table

% HJSIII, 20.02.29

% mechanism constants [mm]

eB = 10;

eC = 10;

s1pA = [ 0 0 ]';

s1pD = [ 0 -70 ]';

s2pA = [ 0 0 ]';

s2pBP = [ eB 29.3 ]';

s2pBQ = [ eB -29.3 ]';

s2pCP = [ -29.3 eC ]';

s2pCQ = [ 29.3 eC ]';

s3pE = [ 0 0 ]';

s3pF = [ 13.984 -7.774 ]';

s3pG = [ 0 -62 ]';

% initial guesses for phi2 = 30 deg

phi2 = 0 \* d2r;

phi3 = -20 \* d2r;

q = [ 0 0 phi2 10 20 phi3 ]';

% driver for crank - phi2 - phi2\_start - w2\*t

phi2\_start = 0;

w2 = -240 \* 2 \* pi / 60; % 240 rpm CW, convert to rad/sec

% constraint entries

% ctype=1 revolute [ ctype body\_i body\_j sipP' sjpP' [0 0] [0 0] ]

% ctype=2 double revolute [ ctype body\_i body\_j sipP' sjpP' L 0 [0 0] ]

% ctype=3 parallel vectors [ ctype body\_i body\_j sipP' sipQ' sjpP' sjpQ' ]

% ctype=4 pin-in-slot [ ctype body\_i body\_j sipP' sipQ' sjpP' [0 0] ]

% ctype=5 relative angle driver [ ctype body\_i body\_j C v a 0 [0 0] [0 0] ]

constraints = [ 1 1 2 s1pA' s2pA' [0 0] [0 0] ; % rev r2A-r1A

4 2 3 s2pBP' s2pBQ' s3pE' [0 0] ; % pin-in-slot r2BQ-r2BP guides r3E

4 2 3 s2pCP' s2pCQ' s3pF' [0 0] ; % pin-in-slot r2CQ-r2CP guides r3F

4 1 3 s1pA' s1pD' s3pG' [0 0] ; % pin-in-slot r1D-r1A guides r3G

5 1 2 phi2\_start w2 0 0 [0 0] [0 0] ]; % phi2-phi1-phi2\_start-w2\*t

% bottom - wanzerq6\_ini

**use sliders - nb=6, nq=15, nc=15**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **col 1**  **ctype** | **col 2**  **body i** | **col 3**  **body j** | **col**  **4,5** | **col**  **6,7** | **col**  **8,9** | **col**  **10,11** | **NOTES** |
| 1  rev | 1 | 2 |  |  | 0 0 | 0 0 | r2A-r1A |
| 3  par vec | 2 | 4 |  |  |  |  | r2BQ-r2BP  r4Q-r4E |
| 4  pin slot | 2 | 4 |  |  |  | 0 0 | r2BQ-r2BP  r4E |
| 1  rev | 3 | 4 |  |  | 0 0 | 0 0 | r4E-r3E |
| 3  par vec | 2 | 5 |  |  |  |  | r2CQ-r2CP  r5Q-r5F |
| 4  pin slot | 2 | 5 |  |  |  | 0 0 | r2CQ-r2CP  r3F |
| 1  rev | 3 | 5 |  |  | 0 0 | 0 0 | r5F-r3F |
| 3  par vec | 1 | 6 |  |  |  |  | r1D-r1A  r6Q-r6G |
| 4  pin slot | 1 | 6 |  |  |  | 0 0 | r1D-r1A  r6G |
| 1  rev | 3 | 6 |  |  | 0 0 | 0 0 | r6G-r3G |
| 5  ang dri | 1 | 2 |  | 0 0 | 0 0 | 0 0 |  |

**acceleration angle driver = **

% wanzerq15\_ini.m - Wanzer sewing machine nq=15 for autofill

% constants, assembly estimates and constraint table

% HJSIII, 20.02.29

% mechanism constants [mm]

eB = 10;

eC = 10;

s1pA = [ 0 0 ]';

s1pD = [ 0 -70 ]';

s2pA = [ 0 0 ]';

s2pBP = [ eB 29.3 ]';

s2pBQ = [ eB -29.3 ]';

s2pCP = [ -29.3 eC ]';

s2pCQ = [ 29.3 eC ]';

s3pE = [ 0 0 ]';

s3pF = [ 13.984 -7.774 ]';

s3pG = [ 0 -62 ]';

s4pE = [ 0 0 ]';

s4pQ = [ 10 0 ]';

s5pF = [ 0 0 ]';

s5pQ = [ 10 0 ]';

s6pG = [ 0 0 ]';

s6pQ = [ 10 0 ]';

% initial guesses for phi2 = 30 deg

phi2 = 0 \* d2r;

phi3 = -15 \* d2r;

phi4 = 90 \* d2r;

phi5 = 0 \* d2r;

phi6 = -90 \* d2r;

q = [ 0 0 phi2 10 20 phi3 10 20 phi4 20 10 phi5 0 -40 phi6 ]';

% driver for crank - phi2 - phi2\_start - w2\*t

phi2\_start = 0;

w2 = -240 \* 2 \* pi / 60; % 240 rpm CW, convert to rad/sec

% constraint entries

% ctype=1 revolute [ ctype body\_i body\_j sipP' sjpP' [0 0] [0 0] ]

% ctype=2 double revolute [ ctype body\_i body\_j sipP' sjpP' L 0 [0 0] ]

% ctype=3 parallel vectors [ ctype body\_i body\_j sipP' sipQ' sjpP' sjpQ' ]

% ctype=4 pin-in-slot [ ctype body\_i body\_j sipP' sipQ' sjpP' [0 0] ]

% ctype=5 relative angle driver [ ctype body\_i body\_j C v a 0 [0 0] [0 0] ]

constraints = [ 1 1 2 s1pA' s2pA' [0 0] [0 0] ; % rev r2A-r1A

3 2 4 s2pBP' s2pBQ' s4pE' s4pQ' ; % par vec r2BQ-r2BP // r4Q-r4E

4 2 4 s2pBP' s2pBQ' s4pE' [0 0] ; % pin-in-slot r2BQ-r2BP guides r4E

1 3 4 s3pE' s4pE' [0 0] [0 0] ; % rev r4E-r3E

3 2 5 s2pCP' s2pCQ' s5pF' s5pQ' ; % par vec r2CQ-r2CP // r5Q-r5F

4 2 5 s2pCP' s2pCQ' s5pF' [0 0] ; % pin-in-slot r2CQ-r2CP guides r5F

1 3 5 s3pF' s5pF' [0 0] [0 0] ; % rev r5F-r3F

3 1 6 s1pA' s1pD' s6pG' s6pQ' ; % par vec r1D-r1A // r6Q-r6G

4 1 6 s1pA' s1pD' s6pG' [0 0] ; % pin-in-slot r1D-r1A guides r6G

1 3 6 s3pG' s6pG' [0 0] [0 0] ; % rev r6G-r3G

5 1 2 phi2\_start w2 0 0 [0 0] [0 0] ]; % phi2-phi1-phi2\_start-w2\*t

% bottom - wanzerq15\_ini

% paf\_main.m - planar autofill constraints and Jacobian

% main with time loop

% HJSIII - 20.02.28

clear

% general constants

d2r = pi / 180;

R = [ 0 -1 ; 1 0 ];

% default start time

t = 0;

% define mechanism

%fbq9\_ini % q(9) = phiCD, keep(9) keep(18) keep(27)

%fbq6\_ini % q(6) = phiCD, keep(6) keep(12) keep(18)

%iscq9\_ini % q(9) = phi4, keep(9) keep(18) keep(27)

%wanzerq6\_ini % y3G = needle, q(6) = phi3, keep(6) keep(12) keep(18)

wanzerq15\_ini % q(14) = y6 = needle, keep(14) keep(29) keep(44)

% time loop

tpr = 2 \* pi / w2; % one revolution at constant speed

t\_start = 0; % start

t\_end = tpr; % end

nt = 180; % number of time steps

dt = (t\_end - t\_start) / nt;

keep = [];

for t = t\_start : dt : t\_end;

% kinematics

paf\_kin

% save

keep = [ keep ; q' qd' qdd' ];

end % bottom - for t

% output for wanzerq15

ang2 = 360 - keep(:,3)/d2r; % phi2, convert to deg, plot negative ang2 because CW

y6 = keep(:,14);

y6d = keep(:,29);

y6dd = keep(:,44)/9810; % convert mm/sec/sec to G

% position

figure( 1 )

clf

plot( ang2,y6,'r' )

% velocity

figure( 2 )

clf

plot( ang2,y6d,'r' )

% acceleration

figure( 3 )

clf

plot( ang2,y6dd,'r' )

% bottom - paf\_main

% paf\_phi.m - automatically fill planar constraints and Jacobian

% form PHI, JAC and velrhs

% HJSIII - 20.02.27

% nq, nb, nc

nq = length( q );

nb = (nq/3) + 1;

[ ntable, ncol ] = size( constraints );

nc = ntable + length( find( constraints(:,1)==1 ) );

% tables for positions and angles - r1 = [0 0]' and phi1 = 0

r\_all = zeros( 2,nb );

phi\_all = zeros( 1,nb );

% rip values from q

for bi = 2 : nb,

ci = 3\*(bi-2) + 1;

r\_all(:,bi) = q( ci:ci+1 );

phi\_all(bi) = q( ci+2 );

end

% tables for rotation matrices

A\_all = zeros( 2,2,nb );

for bi = 1 : nb;

phi = phi\_all(bi);

A\_all(:,:,bi) = [ cos(phi) -sin(phi) ;

sin(phi) cos(phi) ];

end

% allocate constraints, Jacobian and velrhs

PHI = zeros( nc, 1 );

JAC = zeros( nc, nq );

velrhs = zeros( nc, 1 );

% proceed through constraint table

i2 = 0;

for itable = 1 : ntable,

i1 = i2 + 1; % increment row in constraints and Jacobian

ctype = constraints( itable, 1 );

bi = constraints( itable, 2 );

ri = r\_all(:,bi);

phi\_i = phi\_all(bi);

Ai = A\_all(:,:,bi);

Bi = R \* Ai;

ci = 3\*(bi-2) + 1;

bj = constraints( itable, 3 );

rj = r\_all(:,bj);

phi\_j = phi\_all(bj);

Aj = A\_all(:,:,bj);

Bj = R \* Aj;

cj = 3\*(bj-2) + 1;

% revolute

if ctype == 1,

i2 = i1 + 1;

sipP = constraints( itable, 4:5 )';

sjpP = constraints( itable, 6:7 )';

riP = ri + Ai \* sipP;

rjP = rj + Aj \* sjpP;

PHI( i1:i2 ) = rjP - riP;

if ci > 0,

JAC( i1:i2, ci:ci+2 ) = -[ eye(2) Bi\*sipP ];

end

if cj > 0,

JAC( i1:i2, cj:cj+2 ) = [ eye(2) Bj\*sjpP ];

end

velrhs(i1:i2) = [ 0 0 ]';

end % bottom - revolute

% double revolute

if ctype == 2,

i2 = i1;

sipP = constraints( itable, 4:5 )';

sjpP = constraints( itable, 6:7 )';

L = constraints( itable, 8 );

riP = ri + Ai \* sipP;

rjP = rj + Aj \* sjpP;

dij = rjP - riP;

PHI( i1 ) = dij'\*dij - L\*L;

if ci > 0,

JAC( i1, ci:ci+2 ) = -2 \* dij' \* [ eye(2) Bi\*sipP ];

end

if cj > 0,

JAC( i1, cj:cj+2 ) = 2 \* dij' \* [ eye(2) Bj\*sjpP ];

end

velrhs( i1 ) = 0;

end % bottom - double revolute

% parallel vectors

if ctype == 3,

i2 = i1;

sipP = constraints( itable, 4:5 )';

sipQ = constraints( itable, 6:7 )';

sjpP = constraints( itable, 8:9 )';

sjpQ = constraints( itable, 10:11 )';

riP = ri + Ai \* sipP;

riQ = ri + Ai \* sipQ;

rjP = rj + Aj \* sjpP;

rjQ = rj + Aj \* sjpQ;

ai = riQ - riP;

aj = rjQ - rjP;

PHI( i1 ) = ai' \* R' \* aj;

if ci > 0,

JAC( i1, ci:ci+2 ) = [ 0 0 -ai'\*aj ];

end

if cj > 0,

JAC( i1, cj:cj+2 ) = [ 0 0 ai'\*aj ];

end

velrhs( i1 ) = 0;

end % bottom - parallel vectors

% pin-in-slot

if ctype == 4,

i2 = i1;

sipP = constraints( itable, 4:5 )';

sipQ = constraints( itable, 6:7 )';

sjpP = constraints( itable, 8:9 )';

riP = ri + Ai \* sipP;

riQ = ri + Ai \* sipQ;

rjP = rj + Aj \* sjpP;

dij = rjP - riP;

ai = riQ - riP;

PHI( i1 ) = ai' \* R' \* dij;

if ci > 0,

JAC( i1, ci:ci+2 ) = -ai'\*R'\*[ eye(2) Bi\*sipP ] - [ 0 0 ai'\*dij];

end

if cj > 0,

JAC( i1, cj:cj+2 ) = ai'\* R'\*[ eye(2) Bj\*sjpP ];

end

velrhs( i1 ) = 0;

end % bottom - pin-in-slot

% relative angle driver

if ctype == 5,

i2 = i1;

C = constraints( itable, 4 );

v = constraints( itable, 5 );

a = constraints( itable, 6 );

PHI( i1 ) = phi\_j -phi\_i -C -v\*t -a\*t\*t/2;

if ci > 0,

JAC( i1, ci+2 ) = -1;

end

if cj > 0,

JAC( i1, cj+2 ) = 1;

end

velrhs( i1 ) = v +a\*t;

end % bottom - relative angle driver

end % bottom - for itable

% bottom - paf\_phi

% paf\_kin.m - automatically fill planar constraints and Jacobian

% positon, velocity, and acceleration

% HJSIII, 20.02.25

% Newton-Raphson position solution

assy\_tol = 1e-5;

paf\_phi;

while max(abs(PHI)) > assy\_tol,

q = q - inv(JAC) \* PHI;

paf\_phi;

end

% velocity right hand side provided by paf\_phi

qd = inv(JAC) \* velrhs;

% tables for velocities - r1d = [0 0]' and phi1d = 0

% positions and angles should be valid from position solution

rd\_all = zeros( 2,nb );

phid\_all = zeros( 1,nb );

% rip values from qd

for bi = 2 : nb,

ci = 3\*(bi-2) + 1;

rd\_all(:,bi) = qd( ci:ci+1 );

phid\_all(bi) = qd( ci+2 );

end

% allocate acceleration right hand side

accrhs = zeros( nc, 1 );

% proceed through constraint table

[ ntable, ncol ] = size( constraints );

i2 = 0;

for itable = 1 : ntable,

i1 = i2 + 1; % increment row in constraints and Jacobian

ctype = constraints( itable, 1 );

bi = constraints( itable, 2 );

ri = r\_all(:,bi);

phi\_i = phi\_all(bi);

Ai = A\_all(:,:,bi);

Bi = R \* Ai;

ci = 3\*(bi-2) + 1;

rid = rd\_all(:,bi);

phi\_id = phid\_all(bi);

bj = constraints( itable, 3 );

rj = r\_all(:,bj);

phi\_j = phi\_all(bj);

Aj = A\_all(:,:,bj);

Bj = R \* Aj;

cj = 3\*(bj-2) + 1;

rjd = rd\_all(:,bj);

phi\_jd = phid\_all(bj);

% revolute

if ctype == 1,

i2 = i1 + 1;

sipP = constraints( itable, 4:5 )';

sjpP = constraints( itable, 6:7 )';

accrhs(i1:i2) = phi\_jd\*phi\_jd\*Aj\*sjpP - phi\_id\*phi\_id\*Ai\*sipP;

end % bottom - revolute

% double revolute

if ctype == 2,

i2 = i1;

sipP = constraints( itable, 4:5 )';

sjpP = constraints( itable, 6:7 )';

L = constraints( itable, 8 );

riP = ri + Ai \* sipP;

rjP = rj + Aj \* sjpP;

dij = rjP - riP;

riPd = rid + phi\_id\*Bi\*sipP;

rjPd = rjd + phi\_jd\*Bj\*sjpP;

dijd = rjPd - riPd;

gamma\_revolute = phi\_jd\*phi\_jd\*Aj\*sjpP - phi\_id\*phi\_id\*Ai\*sipP;

accrhs( i1 ) = 2\*dij'\*gamma\_revolute -2\*dijd'\*dijd;

end % bottom - double revolute

% parallel vectors

if ctype == 3,

i2 = i1;

sipP = constraints( itable, 4:5 )';

sipQ = constraints( itable, 6:7 )';

sjpP = constraints( itable, 8:9 )';

sjpQ = constraints( itable, 10:11 )';

accrhs( i1 ) = 0;

end % bottom - parallel vectors

% pin-in-slot

if ctype == 4,

i2 = i1;

sipP = constraints( itable, 4:5 )';

sipQ = constraints( itable, 6:7 )';

sjpP = constraints( itable, 8:9 )';

riP = ri + Ai \* sipP;

riQ = ri + Ai \* sipQ;

rjP = rj + Aj \* sjpP;

dij = rjP - riP;

ai = riQ - riP;

riPd = rid + phi\_id\*Bi\*sipP;

rjPd = rjd + phi\_jd\*Bj\*sjpP;

dijd = rjPd - riPd;

gamma\_revolute = phi\_jd\*phi\_jd\*Aj\*sjpP - phi\_id\*phi\_id\*Ai\*sipP;

accrhs( i1 ) = ai'\*( 2\*phi\_id\*dijd + R'\*(phi\_id\*phi\_id\*dij + gamma\_revolute) );

end % bottom - pin-in-slot

% relative angle driver

if ctype == 5,

i2 = i1;

C = constraints( itable, 4 );

v = constraints( itable, 5 );

a = constraints( itable, 6 );

accrhs( i1 ) = a;

end % bottom - relative angle driver

end % bottom - for itable

% acceleration solution

qdd = inv(JAC) \* accrhs;

% bottom - paf\_kin