Two-dimensional autofill constraint vector and Jacobian

1) Provide an initialization file (xxx_ini.m) that contains -
   a) initial estimates for generalized coordinates $\{q\}$
   b) local body-fixed coordinates $\{s_i\}'^P$ 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.

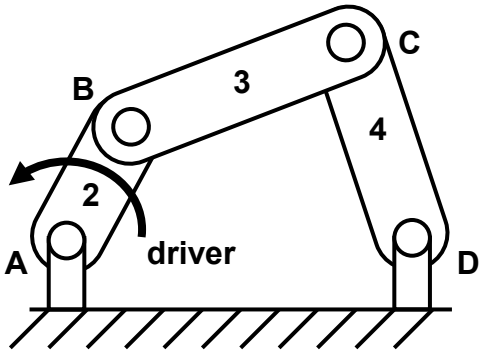
The mechanism may be single loop or multiple loop, may be open loop or closed loop, and must be kinematically driven (nc=nq).

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	$\left(\{s_i\}'^P\right)^T$	$\left(\{s_j\}'^P\right)^T$	0 0	0 0
double revolute	2	i	j	$\left(\{s_i\}'^P\right)^T$	$\left(\{s_j\}'^P\right)^T$	L 0	0 0
parallel vectors	3	i	j	$\left(\{s_i\}'^P\right)^T$	$\left(\{s_i\}'^Q\right)^T$	$\left(\{s_j\}'^P\right)^T$	$\left(\{s_j\}'^Q\right)^T$
pin-in-slot	4	i	j	$\left(\{s_i\}'^P\right)^T$	$\left(\{s_i\}'^Q\right)^T$	$\left(\{s_j\}'^P\right)^T$	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 = $\phi_j - \phi_i - C - v\,t - \frac{1}{2}a\,t^2$

Four bar



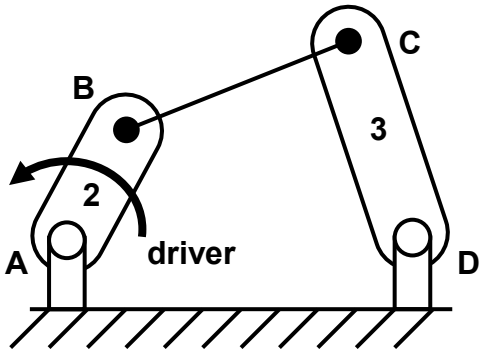
col 1 ctype	col 2 body i	col 3 body j	col 4,5	col 6,7	col 8,9	col 10,11	NOTES
1 rev	1	2	$\left(\{s_1\}'^{A}\right)^T$	$\left(\{s_2\}'^{A}\right)^T$	0 0	0 0	r2A-r1A
1 rev	2	3	$\left(\{s_2\}'^{B}\right)^T$	$\left(\{s_3\}'^{B}\right)^T$	0 0	0 0	r3B-r2B
1 rev	3	4	$\left(\{s_3\}'^{C}\right)^T$	$\left(\{s_4\}'^{C}\right)^T$	0 0	0 0	r4C-r3C
1 rev	1	4	$\left(\{s_1\}'^{D}\right)^T$	$\left(\{s_4\}'^{D}\right)^T$	0 0	0 0	r4D-r1D
5 ang dri	1	2	ϕ_{2_START} ω_2	0 0	0 0	0 0	

acceleration angle driver = $\phi_2 - \phi_1 - \phi_{2\_START} - \omega_2\ t$

```
% 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)



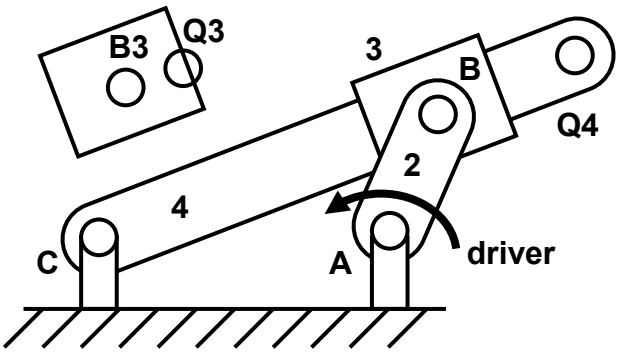
col 1 ctype	col 2 body i	col 3 body j	col 4,5	col 6,7	col 8,9	col 10,11	NOTES
1 rev	1	2	$\left(\{s_1\}'^{A}\right)^T$	$\left(\{s_2\}'^{A}\right)^T$	0 0	0 0	r2A-r1A
2 dbl rev	2	3	$\left(\{s_2\}'^{B}\right)^T$	$\left(\{s_3\}'^{B}\right)^T$	lenBC 0	0 0	r3C-r2B lenBC
1 rev	1	3	$\left(\{s_1\}'^{D}\right)^T$	$\left(\{s_3\}'^{D}\right)^T$	0 0	0 0	r3D-r1D
5 ang dri	1	2	ϕ_{2_START} ω_2	0 0	0 0	0 0	

acceleration angle driver = $\phi_2 - \phi_1 - \phi_{2\_START} - \omega_2\, t$

```
% 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



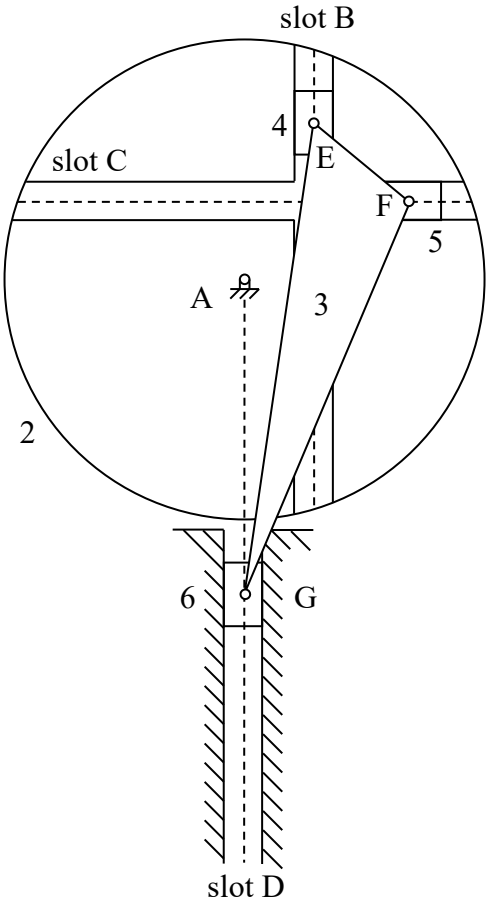
col 1 ctype	col 2 body i	col 3 body j	col 4,5	col 6,7	col 8,9	col 10,11	NOTES
1 rev	1	2	$\left(\{s_1\}'^{A}\right)^T$	$\left(\{s_2\}'^{A}\right)^T$	0 0	0 0	r2A-r1A
1 rev	2	3	$\left(\{s_2\}'^{B}\right)^T$	$\left(\{s_3\}'^{B}\right)^T$	0 0	0 0	r3B-r2B
3 par vec	4	3	$\left(\{s_4\}'^{C}\right)^T$	$\left(\{s_4\}'^{Q}\right)^T$	$\left(\{s_3\}'^{B}\right)^T$	$\left(\{s_3\}'^{Q}\right)^T$	r4Q-r4C r3Q-r3B
4 pin slot	4	3	$\left(\{s_4\}'^{C}\right)^T$	$\left(\{s_4\}'^{Q}\right)^T$	$\left(\{s_3\}'^{B}\right)^T$	0 0	r4C-r3C r3B
1 rev	1	4	$\left(\{s_1\}'^{C}\right)^T$	$\left(\{s_4\}'^{C}\right)^T$	0 0	0 0	r4C-r1C
5 ang dri	1	2	ϕ_{2_START} ω_2	0 0	0 0	0 0	

acceleration angle driver = $\phi_2 - \phi_1 - \phi_{2\_START} - \omega_2\ t$

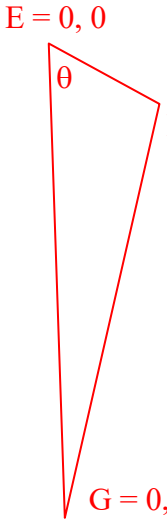
```
% iscq9_ini.m - inverted slider crank nq=9 for autofill
%   constants, assembly estimates and constraint table
% HJSIII, 20.02.28

% 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



EF = 16 mm
FG = 56 mm
EG = 62 mm
r_2 = 31 mm radius of disk 2
e_B = 10 mm radial offset for slot B
e_C = 10 mm radial offset for slot C
θ_{BC} = 90º between slots B and C
w = 5 mm width of slots B, C, and D



$(FG)^2 = (EF)^2 + (EG)^2 - 2\,(EF)\,(EG)\cos\theta$
$\theta = 60.93°$

$\omega_2 = -25.133$ rad/sec

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	$\left(\{s_1\}^{'A}\right)^T$	$\left(\{s_2\}^{'A}\right)^T$	0 0	0 0	r2A-r1A
4 pin slot	2	3	$\left(\{s_2\}^{'BP}\right)^T$	$\left(\{s_2\}^{'BQ}\right)^T$	$\left(\{s_3\}^{'E}\right)^T$	0 0	r2BQ-r2BP r3E
4 pin slot	2	3	$\left(\{s_2\}^{'CP}\right)^T$	$\left(\{s_2\}^{'CQ}\right)^T$	$\left(\{s_3\}^{'F}\right)^T$	0 0	r2CQ-r2CP r3F
4 pin slot	1	3	$\left(\{s_1\}^{'A}\right)^T$	$\left(\{s_1\}^{'D}\right)^T$	$\left(\{s_3\}^{'G}\right)^T$	0 0	r1D-r1A r3G
5 ang dri	1	2	ϕ_{2_START} ω_2	0 0	0 0	0 0	

acceleration angle driver = $\phi_2 - \phi_1 - \phi_{2\_START} - \omega_2\, t$

```
% 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	$\left(\{s_1\}^{'A}\right)^T$	$\left(\{s_2\}^{'A}\right)^T$	0 0	0 0	r2A-r1A
3 par vec	2	4	$\left(\{s_2\}^{'BP}\right)^T$	$\left(\{s_2\}^{'BQ}\right)^T$	$\left(\{s_4\}^{'E}\right)^T$	$\left(\{s_4\}^{'Q}\right)^T$	r2BQ-r2BP r4Q-r4E
4 pin slot	2	4	$\left(\{s_2\}^{'BP}\right)^T$	$\left(\{s_2\}^{'BQ}\right)^T$	$\left(\{s_4\}^{'E}\right)^T$	0 0	r2BQ-r2BP r4E
1 rev	3	4	$\left(\{s_3\}^{'E}\right)^T$	$\left(\{s_4\}^{'E}\right)^T$	0 0	0 0	r4E-r3E
3 par vec	2	5	$\left(\{s_2\}^{'CP}\right)^T$	$\left(\{s_2\}^{'CQ}\right)^T$	$\left(\{s_5\}^{'F}\right)^T$	$\left(\{s_5\}^{'Q}\right)^T$	r2CQ-r2CP r5Q-r5F
4 pin slot	2	5	$\left(\{s_2\}^{'CP}\right)^T$	$\left(\{s_2\}^{'CQ}\right)^T$	$\left(\{s_3\}^{'F}\right)^T$	0 0	r2CQ-r2CP r3F
1 rev	3	5	$\left(\{s_3\}^{'F}\right)^T$	$\left(\{s_5\}^{'F}\right)^T$	0 0	0 0	r5F-r3F
3 par vec	1	6	$\left(\{s_1\}^{'A}\right)^T$	$\left(\{s_1\}^{'D}\right)^T$	$\left(\{s_6\}^{'G}\right)^T$	$\left(\{s_6\}^{'Q}\right)^T$	r1D-r1A r6Q-r6G
4 pin slot	1	6	$\left(\{s_1\}^{'A}\right)^T$	$\left(\{s_1\}^{'D}\right)^T$	$\left(\{s_6\}^{'G}\right)^T$	0 0	r1D-r1A r6G
1 rev	3	6	$\left(\{s_3\}^{'G}\right)^T$	$\left(\{s_6\}^{'G}\right)^T$	0 0	0 0	r6G-r3G
5 ang dri	1	2	ϕ_{2_START} ω_2	0 0	0 0	0 0	

acceleration angle driver = $\phi_2 - \phi_1 - \phi_{2\_START} - \omega_2\ t$

```
% 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
```