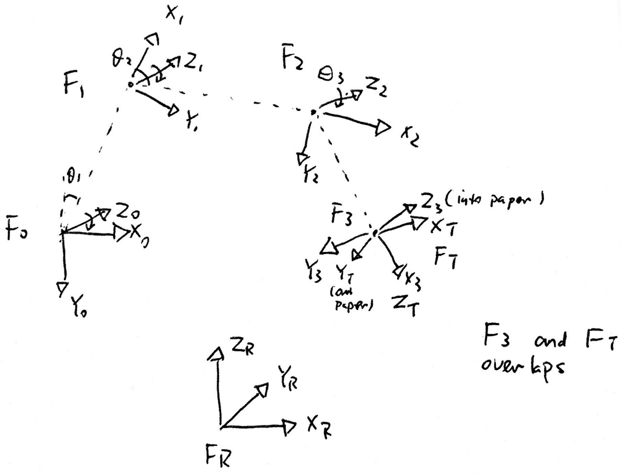
Table of Contents

RBE501 HW3 Keshuai Xu

1



F3 and FT overlaps

	θ	d	a	α
0	0	d+b	−c	$-\frac{\pi}{2}$
1	$\theta_1-\frac{\pi}{2}$	0	e	0
2	θ_2	0	f	0
3	θ_3	0	g	0
T	$-\frac{\pi}{2}$	0	0	$-\frac{\pi}{2}$

```
theta = sym('theta', [3 1]);
syms a b c d e f g;
```

```
% ans
% columns: theta d a alpha
dh = [0, d + b, -c, -sym(pi)/2; % FR->F0
    theta(1) - sym(pi)/2, 0, e, 0; % F1
    theta(2), 0, f, 0; % F2
    theta(3), 0, g, 0; % F3
    -sym(pi)/2, 0, 0, -sym(pi)/2] % FT
```

```
dh =
 
[             0, b + d, -c, -pi/2]
[ theta1 - pi/2,     0,  e,     0]
[        theta2,     0,  f,     0]
[        theta3,     0,  g,     0]
[         -pi/2,     0,  0, -pi/2]
```

2

```
T = sym(zeros(4,4,size(dh, 1)));
for joint=1:size(dh, 1)
    T(:,:,joint) =
 dh2mat(dh(joint,1),dh(joint,2),dh(joint,3),dh(joint,4));
end

% ans
T_R_T = T(:,:,1)*T(:,:,2)*T(:,:,3)*T(:,:,4)*T(:,:,5)
```

```
T_R_T =
 
[ sin(theta3)*(cos(theta2)*cos(theta1 - pi/2) - sin(theta2)*sin(theta1
 - pi/2)) + cos(theta3)*(cos(theta2)*sin(theta1
 - pi/2) + cos(theta1 - pi/2)*sin(theta2)),  0,
 cos(theta3)*(cos(theta2)*cos(theta1 - pi/2) - sin(theta2)*sin(theta1
 - pi/2)) - sin(theta3)*(cos(theta2)*sin(theta1 - pi/2)
 + cos(theta1 - pi/2)*sin(theta2)),     e*cos(theta1
 - pi/2) - c + g*cos(theta3)*(cos(theta2)*cos(theta1
 - pi/2) - sin(theta2)*sin(theta1 - pi/2)) -
 g*sin(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1
 - pi/2)*sin(theta2)) + f*cos(theta2)*cos(theta1 - pi/2) -
 f*sin(theta2)*sin(theta1 - pi/2)]
[

                  0, -1,

                                           0,



                           0]
[ cos(theta3)*(cos(theta2)*cos(theta1 - pi/2) - sin(theta2)*sin(theta1
 - pi/2)) - sin(theta3)*(cos(theta2)*sin(theta1
 - pi/2) + cos(theta1 - pi/2)*sin(theta2)),  0, -
 sin(theta3)*(cos(theta2)*cos(theta1 - pi/2) - sin(theta2)*sin(theta1
 - pi/2)) - cos(theta3)*(cos(theta2)*sin(theta1 - pi/2) +
 cos(theta1 - pi/2)*sin(theta2)), b + d - e*sin(theta1 - pi/2)
 - g*cos(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1 -
 pi/2)*sin(theta2)) - g*sin(theta3)*(cos(theta2)*cos(theta1 - pi/2) -
 sin(theta2)*sin(theta1 - pi/2)) - f*cos(theta2)*sin(theta1 - pi/2) -
 f*cos(theta1 - pi/2)*sin(theta2)]
[

                  0,  0,

                                           0,



                           1]
```

3

```
x_t = T_R_T * [0;0;0;1];
x_t = x_t(1:3,:);
J_upper = jacobian(x_t, theta);
J_lower = [0,0,0;
           1,1,1; % thetas contribute to rotation in y
           0,0,0];
% ans [xdot ydot zdot wz wy wz]'
J_arm = vertcat(J_upper, J_lower)

% ans [xdot zdot wy]'
J_reduced_dof = J_arm([1 3 5], :)
```

```
J_arm =

[ - e*sin(theta1 - pi/2) - g*cos(theta3)*(cos(theta2)*sin(theta1
 - pi/2) + cos(theta1 - pi/2)*sin(theta2)) -
 g*sin(theta3)*(cos(theta2)*cos(theta1 - pi/2) -
 sin(theta2)*sin(theta1 - pi/2)) - f*cos(theta2)*sin(theta1
 - pi/2) - f*cos(theta1 - pi/2)*sin(theta2), -
 g*cos(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1 -
 pi/2)*sin(theta2)) - g*sin(theta3)*(cos(theta2)*cos(theta1 -
 pi/2) - sin(theta2)*sin(theta1 - pi/2)) - f*cos(theta2)*sin(theta1
 - pi/2) - f*cos(theta1 - pi/2)*sin(theta2), -
 g*cos(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1 -
 pi/2)*sin(theta2)) - g*sin(theta3)*(cos(theta2)*cos(theta1 - pi/2) -
 sin(theta2)*sin(theta1 - pi/2))]
[


                                               0,
  0,

                           0]
[   g*sin(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1
 - pi/2)*sin(theta2)) - g*cos(theta3)*(cos(theta2)*cos(theta1 -
 pi/2) - sin(theta2)*sin(theta1 - pi/2)) - e*cos(theta1 - pi/2)
 - f*cos(theta2)*cos(theta1 - pi/2) + f*sin(theta2)*sin(theta1 -
 pi/2),   g*sin(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1
 - pi/2)*sin(theta2)) - g*cos(theta3)*(cos(theta2)*cos(theta1 -
 pi/2) - sin(theta2)*sin(theta1 - pi/2)) - f*cos(theta2)*cos(theta1
 - pi/2) + f*sin(theta2)*sin(theta1 - pi/2),
 g*sin(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1 -
 pi/2)*sin(theta2)) - g*cos(theta3)*(cos(theta2)*cos(theta1 - pi/2) -
 sin(theta2)*sin(theta1 - pi/2))]
[


                                               0,



  0,

                           0]
[


                                               1,



  1,

                           1]
[


                                               0,



  0,

                           0]


J_reduced_dof =

[ - e*sin(theta1 - pi/2) - g*cos(theta3)*(cos(theta2)*sin(theta1
 - pi/2) + cos(theta1 - pi/2)*sin(theta2)) -
 g*sin(theta3)*(cos(theta2)*cos(theta1 - pi/2) -
 sin(theta2)*sin(theta1 - pi/2)) - f*cos(theta2)*sin(theta1
 - pi/2) - f*cos(theta1 - pi/2)*sin(theta2), -
 g*cos(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1 -
 pi/2)*sin(theta2)) - g*sin(theta3)*(cos(theta2)*cos(theta1 -
 pi/2) - sin(theta2)*sin(theta1 - pi/2)) - f*cos(theta2)*sin(theta1
 - pi/2) - f*cos(theta1 - pi/2)*sin(theta2), -
 g*cos(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1 -
 pi/2)*sin(theta2)) - g*sin(theta3)*(cos(theta2)*cos(theta1 - pi/2) -
 sin(theta2)*sin(theta1 - pi/2))]
[   g*sin(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1
 - pi/2)*sin(theta2)) - g*cos(theta3)*(cos(theta2)*cos(theta1 -
 pi/2) - sin(theta2)*sin(theta1 - pi/2)) - e*cos(theta1 - pi/2)
 - f*cos(theta2)*cos(theta1 - pi/2) + f*sin(theta2)*sin(theta1 -
 pi/2),   g*sin(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1
 - pi/2)*sin(theta2)) - g*cos(theta3)*(cos(theta2)*cos(theta1 -
 pi/2) - sin(theta2)*sin(theta1 - pi/2)) - f*cos(theta2)*cos(theta1
 - pi/2) + f*sin(theta2)*sin(theta1 - pi/2),
 g*sin(theta3)*(cos(theta2)*sin(theta1 - pi/2) + cos(theta1 -
 pi/2)*sin(theta2)) - g*cos(theta3)*(cos(theta2)*cos(theta1 - pi/2) -
 sin(theta2)*sin(theta1 - pi/2))]
[


                                               1,



  1,

                           1]
```

4a

```
T_R_T_numeric = subs(T_R_T, [b, c, d, e, f, g], ...
                [361, 250, 380, 328, 323, 82.4]);
T_R_T_numeric = subs(T_R_T_numeric, theta, [sym(pi)/6; sym(pi)/2;
 sym(pi)/6]);

% ans
double(T_R_T_numeric) % mm
```

```
ans =

    0.8660         0    0.5000  234.9262
         0   -1.0000         0         0
    0.5000         0   -0.8660  792.1958
         0         0         0    1.0000
```

4b

```
J_arm_numeric = subs(J_arm, [b, c, d, e, f, g], ...
                [361, 250, 380, 328, 323, 82.4]); % mm
J_arm_numeric = subs(J_arm_numeric, theta, [sym(pi)/3; sym(pi)/2;
 sym(pi)/3]); % rad

q_dot_numeric = [sym(pi)/4; sym(pi)/4; sym(pi)/4]; % rad/s
x_dot_numeric = J_arm_numeric * q_dot_numeric; % mm, rad/s

x_dot_numeric_deg = double(x_dot_numeric);
x_dot_numeric_deg(4:6,:) = rad2deg(x_dot_numeric_deg(4:6,:));

% ans
x_dot_numeric_deg % mm/s, deg/s
```

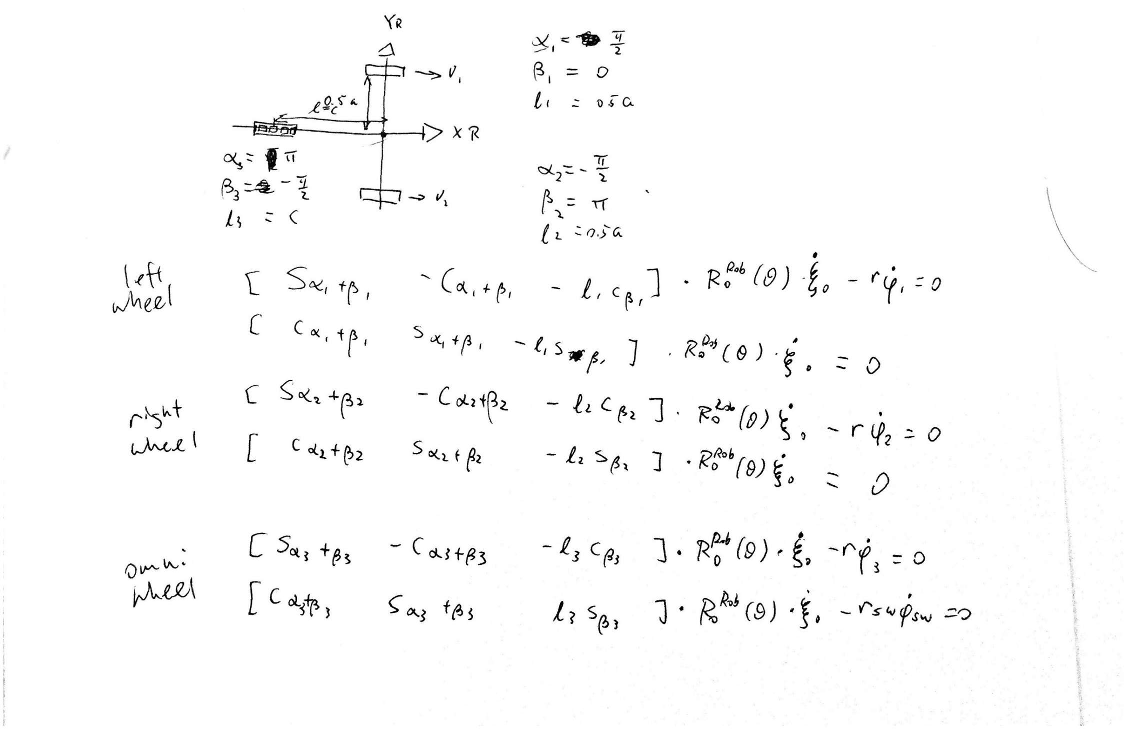
```
x_dot_numeric_deg =

 -478.7268
         0
 -379.7057
         0
  135.0000
         0
```

4c

```
F_numeric = [50; 0; 0; 0; 0; 0]; % N, N*mm
joint_torque = J_arm_numeric'*F_numeric;

% ans
double(joint_torque) % N*mm
```

```
ans =

   1.0e+04 *

   -0.9354
   -1.7554
   -0.3568
```

5


Y_R
X_R
v_1
v_2
$l = 0.5a$
c

$\alpha_1 = \frac{\pi}{2}$
$\beta_1 = 0$
$l_1 = 0.5a$

$\alpha_3 = \pi$
$\beta_3 = -\frac{\pi}{2}$
$l_3 = c$

$\alpha_2 = -\frac{\pi}{2}$
$\beta_2 = \pi$
$l_2 = 0.5a$

left wheel

$$[\, S_{\alpha_1+\beta_1} \quad -C_{\alpha_1+\beta_1} \quad -l_1 c_{\beta_1} \,] \cdot R_0^{Rob}(\theta) \cdot \dot{\xi}_0 - r\dot{\varphi}_1 = 0$$

$$[\, C_{\alpha_1+\beta_1} \quad S_{\alpha_1+\beta_1} \quad -l_1 S_{\beta_1} \,] \cdot R_0^{Rob}(\theta) \cdot \dot{\xi}_0 = 0$$

right wheel

$$[\, S_{\alpha_2+\beta_2} \quad -C_{\alpha_2+\beta_2} \quad -l_2 C_{\beta_2} \,] \cdot R_0^{Rob}(\theta)\, \dot{\xi}_0 - r\dot{\varphi}_2 = 0$$

$$[\, C_{\alpha_2+\beta_2} \quad S_{\alpha_2+\beta_2} \quad -l_2 S_{\beta_2} \,] \cdot R_0^{Rob}(\theta)\, \dot{\xi}_0 = 0$$

omni wheel

$$[\, S_{\alpha_3+\beta_3} \quad -C_{\alpha_3+\beta_3} \quad -l_3 C_{\beta_3} \,] \cdot R_0^{Rob}(\theta) \cdot \dot{\xi}_0 - r\dot{\varphi}_3 = 0$$

$$[\, C_{\alpha_3+\beta_3} \quad S_{\alpha_3+\beta_3} \quad l_3 S_{\beta_3} \,] \cdot R_0^{Rob}(\theta) \cdot \dot{\xi}_0 - r_{sw}\dot{\varphi}_{sw} = 0$$

6

```
syms theta_base r;
phi_dot = sym('phi_dot', [4,1]); %left right omni_big omni_small


alpha = [sym(pi)/2; -sym(pi)/2; sym(pi)];
beta = [0; sym(pi); -sym(pi)/2];
l = [0.5*a; 0.5*a; c];

% equations stolen from lecture slides
% left hand side vectors
R_rob_0 = @(th) [cos(th), sin(th), 0; -sin(th), cos(th), 0; 0, 0, 1];
cons_fixed_roll = @(a, b, l) [sin(a+b), -cos(a+b), -l*cos(b)];
cons_fixed_slide = @(a, b, l) [cos(a+b), sin(a+b), -l*sin(b)];
cons_omni_roll = @(a, b, l) [sin(a+b), -cos(a+b), -l*cos(b)];
cons_omni_slide = @(a, b, l) [cos(a+b), sin(a+b), l*sin(b)];

% ans
J1_rolling = ...
    [cons_fixed_roll(alpha(1), beta(1), l(1)); % left roll
    cons_fixed_roll(alpha(2), beta(2), l(2)); % right roll
    cons_omni_roll(alpha(3), beta(3), l(3))] % omni roll
C1_sliding = ...
    [cons_fixed_slide(alpha(1), beta(1), l(1)); % left slide
    cons_fixed_slide(alpha(2), beta(2), l(2)); % right slide
    cons_omni_slide(alpha(3), beta(3), l(3))] % omni slide

J2_rolling = ...
    [r;
    r;
    r]
C2_sliding = ...
    [0;
    0;
    r*phi_dot(4)]


left_hand_side = [J1_rolling;C1_sliding]
right_hand_side = [J2_rolling.*phi_dot(1:3);C2_sliding]

% remove non-linearly-independent or non controllable equations
% We are not removing them. We just don't need them -Chan
% ans
left_hand_side = left_hand_side([1,2,4],:)
right_hand_side = right_hand_side([1,2,4],:)
```

```
J1_rolling =

[ 1, 0, -a/2]
[ 1, 0,  a/2]
[ 1, 0,    0]


C1_sliding =

[ 0, 1,  0]
[ 0, 1,  0]
[ 0, 1, -c]


J2_rolling =

 r
 r
 r


C2_sliding =

          0
          0
 phi_dot4*r


left_hand_side =

[ 1, 0, -a/2]
[ 1, 0,  a/2]
[ 1, 0,    0]
[ 0, 1,    0]
[ 0, 1,    0]
[ 0, 1,   -c]


right_hand_side =

 phi_dot1*r
 phi_dot2*r
 phi_dot3*r
          0
          0
 phi_dot4*r


left_hand_side =

[ 1, 0, -a/2]
[ 1, 0,  a/2]
[ 0, 1,    0]


right_hand_side =

 phi_dot1*r
 phi_dot2*r
          0
```

7

```
% phi_dot1 is left wheel speed. phi_dot2 is right wheel speed.
 theta_base
% is the rotation of the base frame wrt world frame
xi_0_dot =
 inv(R_rob_0(theta_base))*inv(left_hand_side)*right_hand_side
```

```
xi_0_dot =

 (phi_dot1*r*cos(theta_base))/(2*(cos(theta_base)^2 +
 sin(theta_base)^2)) + (phi_dot2*r*cos(theta_base))/
(2*(cos(theta_base)^2 + sin(theta_base)^2))
 (phi_dot1*r*sin(theta_base))/(2*(cos(theta_base)^2 +
 sin(theta_base)^2)) + (phi_dot2*r*sin(theta_base))/
(2*(cos(theta_base)^2 + sin(theta_base)^2))

                                                   (phi_dot2*r)/a -
 (phi_dot1*r)/a
```

8

```
xi_0_dot_numeric = subs(xi_0_dot, [a, r, theta_base], [507, 143,
 sym(pi)/4]);
xi_0_dot_numeric = subs(xi_0_dot_numeric, phi_dot,
 [2*sym(pi);4*sym(pi);0;0]);
% ans
double(xi_0_dot_numeric) % mm/s rad/s
```

```
ans =

  952.9984
  952.9984
    1.7722
```

9

```
% x_base is position of F_r wrt F_w [x;y;z;thetax;thetay;thetaz]
x_base = sym('x_base', [6,1]);

T_W_R = [[inv(R_rob_0(x_base(6))),[x_base(1);x_base(2);0]];[0,0,0,1]];
T_W_T = T_W_R * T_R_T;
% ans
T_W_T = simplify(T_W_T)
```

```
T_W_T =

[ - cos(theta1 + theta2 + theta3 + x_base6)/2 - cos(theta1
 + theta2 + theta3 - x_base6)/2,  sin(x_base6), sin(theta1
 + theta2 + theta3 - x_base6)/2 + sin(theta1 + theta2
 + theta3 + x_base6)/2, x_base1 - c*cos(x_base6) +
 e*cos(x_base6)*sin(theta1) + f*cos(theta1)*cos(x_base6)*sin(theta2)
 + f*cos(theta2)*cos(x_base6)*sin(theta1) +
 g*cos(theta1)*cos(theta2)*cos(x_base6)*sin(theta3) +
 g*cos(theta1)*cos(theta3)*cos(x_base6)*sin(theta2) +
 g*cos(theta2)*cos(theta3)*cos(x_base6)*sin(theta1) -
 g*cos(x_base6)*sin(theta1)*sin(theta2)*sin(theta3)]
[   sin(theta1 + theta2 + theta3 - x_base6)/2 - sin(theta1
 + theta2 + theta3 + x_base6)/2, -cos(x_base6), cos(theta1
 + theta2 + theta3 - x_base6)/2 - cos(theta1 + theta2
 + theta3 + x_base6)/2, x_base2 - c*sin(x_base6) +
 e*sin(theta1)*sin(x_base6) + f*cos(theta1)*sin(theta2)*sin(x_base6)
 + f*cos(theta2)*sin(theta1)*sin(x_base6) +
 g*cos(theta1)*cos(theta2)*sin(theta3)*sin(x_base6) +
 g*cos(theta1)*cos(theta3)*sin(theta2)*sin(x_base6) +
 g*cos(theta2)*cos(theta3)*sin(theta1)*sin(x_base6) -
 g*sin(theta1)*sin(theta2)*sin(theta3)*sin(x_base6)]
[                                                          sin(theta1
 + theta2 + theta3),              0,
                      cos(theta1 + theta2 + theta3),
                                        b + d + f*cos(theta1 + theta2)
 + e*cos(theta1) + g*cos(theta1 + theta2 + theta3)]
[
                  0,              0,
                                                  0,




                                                 1]
```

10

```
T_W_T_numeric = subs(T_W_T, [b, c, d, e, f, g], ...
                [361, 250, 380, 328, 323, 82.4]);
T_W_T_numeric = subs(T_W_T_numeric, theta, [sym(pi)/6; sym(pi)/2;
 sym(pi)/6]);
T_W_T_numeric = subs(T_W_T_numeric, [a, r], [507, 143]);
T_W_T_numeric = subs(T_W_T_numeric, x_base, [2000; 1000; 0; 0; 0;
 sym(pi)/4]);
% ans
double(T_W_T_numeric) % mm
```

```
ans =

   1.0e+03 *

    0.0006    0.0007    0.0004    2.1661
    0.0006   -0.0007    0.0004    1.1661
    0.0005         0   -0.0009    0.7922
         0         0         0    0.0010
```

11

```
x_t_w = T_W_T * [0; 0; 0; 1];
J_arm_w_top = jacobian(x_t_w(1:3,:) , theta);

slice_zvect = @(m) m(1:3,3);
z_vect = sym(zeros(3,3));
z_vect(:,1) = slice_zvect(T_W_R*T(:,:,1)*T(:,:,2));
z_vect(:,2) = slice_zvect(T_W_R*T(:,:,1)*T(:,:,2)*T(:,:,3));
z_vect(:,3) = slice_zvect(T_W_R*T(:,:,1)*T(:,:,2)*T(:,:,3)*T(:,:,4));
J_arm_w_bottom = z_vect(1:3,:);

J_arm_w = vertcat(J_arm_w_top, J_arm_w_bottom);

xi_0_dot_full = [xi_0_dot(1:2,:); zeros(3,1); xi_0_dot(3,:)];
```

```
J_base = jacobian(xi_0_dot_full, phi_dot(1:2));

% [arm_joints wheels]
J = horzcat(J_arm_w, J_base)

% matrix-vector equation:
% [x_dot;y_dot;z_dot;wx;wy;wz] = J *
 [theta_dot1;theta_dot2;theta_dot3;phi_dot1;phi_dot2]
% where theta_dot are arm joint speeds and phi_dot are left and right
 wheel
% speeds
```

```
J =

[ e*cos(theta1)*cos(x_base6) + f*cos(theta1)*cos(theta2)*cos(x_base6)
 - f*cos(x_base6)*sin(theta1)*sin(theta2) +
 g*cos(theta1)*cos(theta2)*cos(theta3)*cos(x_base6) -
 g*cos(theta1)*cos(x_base6)*sin(theta2)*sin(theta3) -
 g*cos(theta2)*cos(x_base6)*sin(theta1)*sin(theta3) -
 g*cos(theta3)*cos(x_base6)*sin(theta1)*sin(theta2),
 f*cos(theta1)*cos(theta2)*cos(x_base6) -
 f*cos(x_base6)*sin(theta1)*sin(theta2) +
 g*cos(theta1)*cos(theta2)*cos(theta3)*cos(x_base6) -
 g*cos(theta1)*cos(x_base6)*sin(theta2)*sin(theta3) -
 g*cos(theta2)*cos(x_base6)*sin(theta1)*sin(theta3) -
 g*cos(theta3)*cos(x_base6)*sin(theta1)*sin(theta2),
 g*cos(theta1)*cos(theta2)*cos(theta3)*cos(x_base6) -
 g*cos(theta1)*cos(x_base6)*sin(theta2)*sin(theta3) -
 g*cos(theta2)*cos(x_base6)*sin(theta1)*sin(theta3) -
 g*cos(theta3)*cos(x_base6)*sin(theta1)*sin(theta2),
 (r*cos(theta_base))/(2*(cos(theta_base)^2 + sin(theta_base)^2)),
 (r*cos(theta_base))/(2*(cos(theta_base)^2 + sin(theta_base)^2))]
[ e*cos(theta1)*sin(x_base6) + f*cos(theta1)*cos(theta2)*sin(x_base6)
 - f*sin(theta1)*sin(theta2)*sin(x_base6) +
 g*cos(theta1)*cos(theta2)*cos(theta3)*sin(x_base6) -
 g*cos(theta1)*sin(theta2)*sin(theta3)*sin(x_base6) -
 g*cos(theta2)*sin(theta1)*sin(theta3)*sin(x_base6) -
 g*cos(theta3)*sin(theta1)*sin(theta2)*sin(x_base6),
 f*cos(theta1)*cos(theta2)*sin(x_base6) -
 f*sin(theta1)*sin(theta2)*sin(x_base6) +
 g*cos(theta1)*cos(theta2)*cos(theta3)*sin(x_base6) -
 g*cos(theta1)*sin(theta2)*sin(theta3)*sin(x_base6) -
 g*cos(theta2)*sin(theta1)*sin(theta3)*sin(x_base6) -
 g*cos(theta3)*sin(theta1)*sin(theta2)*sin(x_base6),
 g*cos(theta1)*cos(theta2)*cos(theta3)*sin(x_base6) -
 g*cos(theta1)*sin(theta2)*sin(theta3)*sin(x_base6) -
 g*cos(theta2)*sin(theta1)*sin(theta3)*sin(x_base6) -
 g*cos(theta3)*sin(theta1)*sin(theta2)*sin(x_base6),
 (r*sin(theta_base))/(2*(cos(theta_base)^2 + sin(theta_base)^2)),
 (r*sin(theta_base))/(2*(cos(theta_base)^2 + sin(theta_base)^2))]
[
                                            - f*sin(theta1 + theta2) -
 e*sin(theta1) - g*sin(theta1 + theta2 + theta3),


   - f*sin(theta1 + theta2) - g*sin(theta1 + theta2 + theta3),

                               -g*sin(theta1 + theta2 + theta3),
                                                          0,
                                                    0]
[


                                                                      -
sin(x_base6)/(cos(x_base6)^2 + sin(x_base6)^2),


            -sin(x_base6)/(cos(x_base6)^2 + sin(x_base6)^2),

             -sin(x_base6)/(cos(x_base6)^2 + sin(x_base6)^2),
                                                        0,
                                                  0]
[


  cos(x_base6)/(cos(x_base6)^2 + sin(x_base6)^2),


                cos(x_base6)/(cos(x_base6)^2 + sin(x_base6)^2),

                 cos(x_base6)/(cos(x_base6)^2 + sin(x_base6)^2),
                                                          0,
                                                    0]
[



                                          0,



                                                        0,


                                                         0,
                                                 -r/a,
                                            r/a]
```

12

```
J_numeric = subs(J, [b, c, d, e, f, g], ...
                [361, 250, 380, 328, 323, 82.4]);
J_numeric = subs(J_numeric, theta, [sym(pi)/6; sym(pi)/2; sym(pi)/6]);
J_numeric = subs(J_numeric, [a, r], [507, 143]);
J_numeric = subs(J_numeric, theta_base, sym(pi)/4);

% [torque_arm_joints; torque_wheels]
joint_wheel_torque = J_numeric'*F_numeric;

% ans
double(joint_wheel_torque(4:5,:)) % N*mm
```

```
ans =

   1.0e+03 *

    2.5279
    2.5279
```

Published with MATLAB® R2016a