```
function [q_next, w_next, wh_next, tau_w] =↙
step_attitude_wheels(q, w, wh, tau_body_cmd, p, dt)

tau_w_cmd = -tau_body_cmd;

% wheel torque and speed limits
tau_w = wheel_saturate(tau_w_cmd, wh, p);

% Applied body torque
tau_body = -tau_w;

% RK4 [q; w; wh]
x = [q; w; wh];

k1 = deriv(x, tau_body, tau_w, p);
k2 = deriv(x + 0.5*dt*k1, tau_body, tau_w, p);
k3 = deriv(x + 0.5*dt*k2, tau_body, tau_w, p);
k4 = deriv(x + dt*k3, tau_body, tau_w, p);

x_next = x + (dt/6)*(k1 + 2*k2 + 2*k3 + k4);

q_next = x_next(1:4);
q_next = q_next / norm(q_next);

w_next = x_next(5:7);
wh_next = x_next(8:10);

end

function dx = deriv(x, tau_body, tau_w, p)
q  = x(1:4);
w  = x(5:7);
wh = x(8:10);

% Quat kinematics
Omega = [ 0   -w(1) -w(2) -w(3);
```

```
          w(1)  0    w(3) -w(2);
          w(2) -w(3) 0     w(1);
          w(3)  w(2) -w(1) 0 ];
qdot = 0.5 * Omega * q;

% Rigid body
J = p.J_body;
wdot = J \ (tau_body - cross(w, J*w));

% Wheel dynamics
whdot = tau_w / p.Iw;

dx = [qdot; wdot; whdot];
end
```