```
function [thetalist, success] = IKinBodyIterations(Blist, M, T, thetalist0,
eomg, ev)
% *** CHAPTER 6: INVERSE KINEMATICS ***
% Takes Blist: The joint screw axes in the end-effector frame when the
%              manipulator is at the home position, in the format of a
%              matrix with the screw axes as the columns,
%       M: The home configuration of the end-effector,
%       T: The desired end-effector configuration Tsd,
%       thetalist0: An initial guess of joint angles that are close to
%                   satisfying Tsd,
%       eomg: A small positive tolerance on the end-effector orientation
%             error. The returned joint angles must give an end-effector
%             orientation error less than eomg,
%       ev: A small positive tolerance on the end-effector linear position
%           error. The returned joint angles must give an end-effector
%           position error less than ev.
% Returns thetalist: Joint angles that achieve T within the specified
%                    tolerances,
%         success: A logical value where TRUE means that the function found
%                  a solution and FALSE means that it ran through the set
%                  number of maximum iterations without finding a solution
%                  within the tolerances eomg and ev.
% Uses an iterative Newton-Raphson root-finding method.
% The maximum number of iterations before the algorithm is terminated has
% been hardcoded in as a variable called maxiterations. It is set to 20 at
% the start of the function, but can be changed if needed.
% Example Inputs:
%
% clear; clc;
% Blist = [[0; 0; -1; 2; 0; 0], [0; 0; 0; 0; 1; 0], [0; 0; 1; 0; 0; 0.1]];
% M = [[-1, 0, 0, 0]; [0, 1, 0, 6]; [0, 0, -1, 2]; [0, 0, 0, 1]];
% T = [[0, 1, 0, -5]; [1, 0, 0, 4]; [0, 0, -1, 1.6858]; [0, 0, 0, 1]];
% thetalist0 = [1.5; 2.5; 3];
% eomg = 0.01;
% ev = 0.001;
% [thetalist, success] = IKinBody(Blist, M, T, thetalist0, eomg, ev)
%
% Output:
% thetalist =
%    1.5707
%    2.9997
%    3.1415
% success =
%     1
waypoint_array(1,:)=thetalist0;

thetalist = thetalist0;
i = 0;
maxiterations = 20;
Tsb= FKinBody(M, Blist, thetalist);
Vb = se3ToVec(MatrixLog6(TransInv(FKinBody(M, Blist, thetalist)) * T));
eror(1) = norm(Vb(1:3)); eror(2)=norm(Vb(4:6));
```

```
disp('iteration =');
disp(i);
disp('theta =');
disp(thetalist);
disp('Tsb =');
disp(Tsb);
disp('Vb =');
disp(Vb);
disp('error =');
disp(eror);
err = norm(Vb(1: 3)) > eomg || norm(Vb(4: 6)) > ev;
while err && i < maxiterations
    thetalist = thetalist + pinv(JacobianBody(Blist, thetalist)) * Vb;
    i = i + 1;
    waypoint_array(i,:)=thetalist;
    Tsb= FKinBody(M, Blist, thetalist);
    Vb = se3ToVec(MatrixLog6(TransInv(FKinBody(M, Blist, thetalist)) * T));
    eror(1) = norm(Vb(1:3)); eror(2)=norm(Vb(4:6));
    err = norm(Vb(1: 3)) > eomg || norm(Vb(4: 6)) > ev;
    disp('iteration =');
    disp(i);
    disp('theta =');
    disp(thetalist);
    disp('Tsb =');
    disp(Tsb);
    disp('Vb =');
    disp(Vb);
    disp('error =');
    disp(eror);
end
success = ~ err;
writematrix(waypoint_array, 'waypoint_array.csv')
end
```

```
Not enough input arguments.

Error in IKinBodyIterations (line 44)
waypoint_array(1,:)=thetalist0;
```

Published with MATLAB® R2023b