```
function lab2(theta_1, theta_2, theta_3, theta_4, theta_5, theta_6)
% Lab 2 (Forward Kinematics)
% Computes the transformation matrix T of the end-effector using the joint
% angles of the UR3e robot
%
% Input: 6 joint angles theta_1...theta_6 (degrees)
% Output: none (but prints forward kinematics results to command window

% Code uses Modern Robotics' VecTose3() function
addpath('/Users/davidlim/Documents/ModernRobotics/packages/MATLAB/mr');

% First, define the rest of the screw axes, in (mm)
S1 = [0, 0, 1, -300, 0, 0]';
S2 = [0, 1, 0, -152-88, 0, 0]';
S3 = [0, 1, 0, -152-88, 0, 244]';
S4 = [0, 1, 0, -152-88, 0, 244+213]';
S5 = [0, 0, -1, 300-120+93-104, 244+213, 0]';
S6 = [0, 1, 0, -152-88+85, 0, 244+213]';

% Next, define the M matrix (the zero-position e-e transformation matrix),
% in (mm)
M = [1 0 0 244+213; ...
     0 1 0 -300+120-93+104+92+78; ...
     0 0 1 152+88-85; ...
     0 0 0 1];

% Convert degrees to radians
theta_1 = deg2rad(theta_1);
theta_2 = deg2rad(theta_2);
theta_3 = deg2rad(theta_3);
theta_4 = deg2rad(theta_4);
theta_5 = deg2rad(theta_5);
theta_6 = deg2rad(theta_6);

% Now, calculate the T matrix using forward kinematics
T =
expm(VecTose3(S1*theta_1))*expm(VecTose3(S2*theta_2))*expm(VecTose3(S3*theta_3
))* ...

expm(VecTose3(S4*theta_4))*expm(VecTose3(S5*theta_5))*expm(VecTose3(S6*theta_6
))*M;

% Output results to command window
disp('The transformation matrix T is:')
disp(T)
fprintf('The position of the end-effector is %4.2f, %4.2f, %4.2f (mm)
\n',T(1,4),T(2,4),T(3,4))
end
```

Published with MATLAB® R2024a