

```

%Programmed by John Wu
clear all;
clear drawRR;
clear drawLinks;

SetupRRParams();
delta_t = .01;
Goal_Loc = [10, 4]';

% guess for the configuration
config_deg = [45, -45]'; %beginning configuration - set to test
%% draw RR
%plot the robot arm and the goal location
drawRR(config_deg, P);
plot(Goal_Loc(1), Goal_Loc(2), 'r*', 'MarkerSize', 5);

omega = [.4 .4]; %sample omega control
endConfigDeg = [0, 0]';
% Here I calculate the ending configuration
    %Calculation for theta1_E
endConfigDeg(1) = acos((P.l1^2 + (Goal_Loc(1)^2 + Goal_Loc(2)^2) - P.l2^2)/(2*sqrt(Goal_Loc(1)^2+Goal_Loc(2)^2) *P.l1));
endConfigDeg(2) = acos((P.l1^2 + P.l2^2 - (Goal_Loc(1)^2 + Goal_Loc(2)^2))/(2*P.l1*P.l2));
if(atan2(Goal_Loc(2), Goal_Loc(1)) - endConfigDeg(1) < 0)
    disp("Less than");
    disp(rad2deg(endConfigDeg(1)));
    disp(rad2deg(atan2(Goal_Loc(2), Goal_Loc(1))));
    endConfigDeg(1) = rad2deg(atan2(Goal_Loc(2), Goal_Loc(1)) + endConfigDeg(1));
    endConfigDeg(2) = -rad2deg(pi - endConfigDeg(2));
else
    disp("Greater than");
    endConfigDeg(1) = rad2deg(atan2(Goal_Loc(2), Goal_Loc(1)) - endConfigDeg(1));
    endConfigDeg(2) = rad2deg(pi - endConfigDeg(2));
end

dev_Error = .001; % Used to show that we're close enough to our goal

for t = 0:delta_t:1000

    pose_endEffector = forwardKinematics(config_deg, P);
    J = calculateJacobian(config_deg, P);
    J_inv = pinv(J(1:2,:));
    delta_pos = Goal_Loc - pose_endEffector(1:2);
    distToGoal = norm(delta_pos);

    %TODO create a control strategy to move the robot end effector to the
    %goal location -- replace the following example code.
    omega1 = (endConfigDeg(1) - config_deg(1));
    omega2 = (endConfigDeg(2) - config_deg(2));
    greaterAngle = max([abs(omega1), abs(omega2)]);
    omega = [omega1/(greaterAngle*P.max_motor_speed) omega2/(greaterAngle*P.max_motor_speed)];

    %End TODO if you put your motor speeds in the omega roww vector [omega(1), omega(2)]
    config_next = config_deg + rad2deg(delta_t*omega');
    config_deg = config_next;
    drawRR(config_deg, P);

```

```
    if(disToGoal < .1)
        break;
    end

    pause(delta_t);
end
```