Evaluation scheme

Evaluation for each of the days is attached below

Day-01: Optimization Intro and MATLAB warmup

- The first day is only warmup and is not evaluated

Day-02: Robot Design [100]

Five-bar workspace optimisation [50]

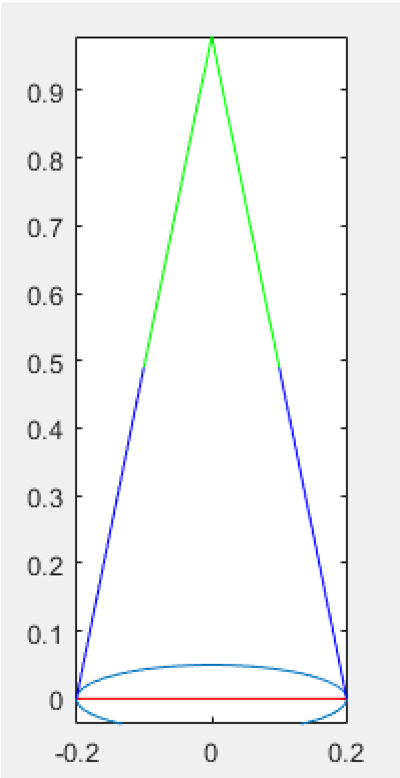
Working code to obtain the solution for the constrained problem [25]

- Q1. What do you observe upon solving the unbounded optimisation problem, i.e., what is the solution and what does it mean? [5]
- L1 and l3 are positive infinity and d is a negative number. It means the solutions is not feasible.

- Q2. What do you observe upon solving the bounded optimisation problem, i.e., what is the solution and what does it mean? [5]

- L1 = l3 = 0.5, d = 0.4. It means the solution is a symmetrical structure and the d equals the half of elliptical long axis.

- Q3. How do the results compare with the two-link robot optimised using pen and paper?
    - Draw the robot configuration corresponding to the optimal values [2.5]



-

    - Explain the resulting optima (why the link lengths and the offsets are so) [2.5]
- 1. the offset d equals the human waist width, because the closer are the two ends of the offset, the larger the overlapping area is, so is the workspace bigger.

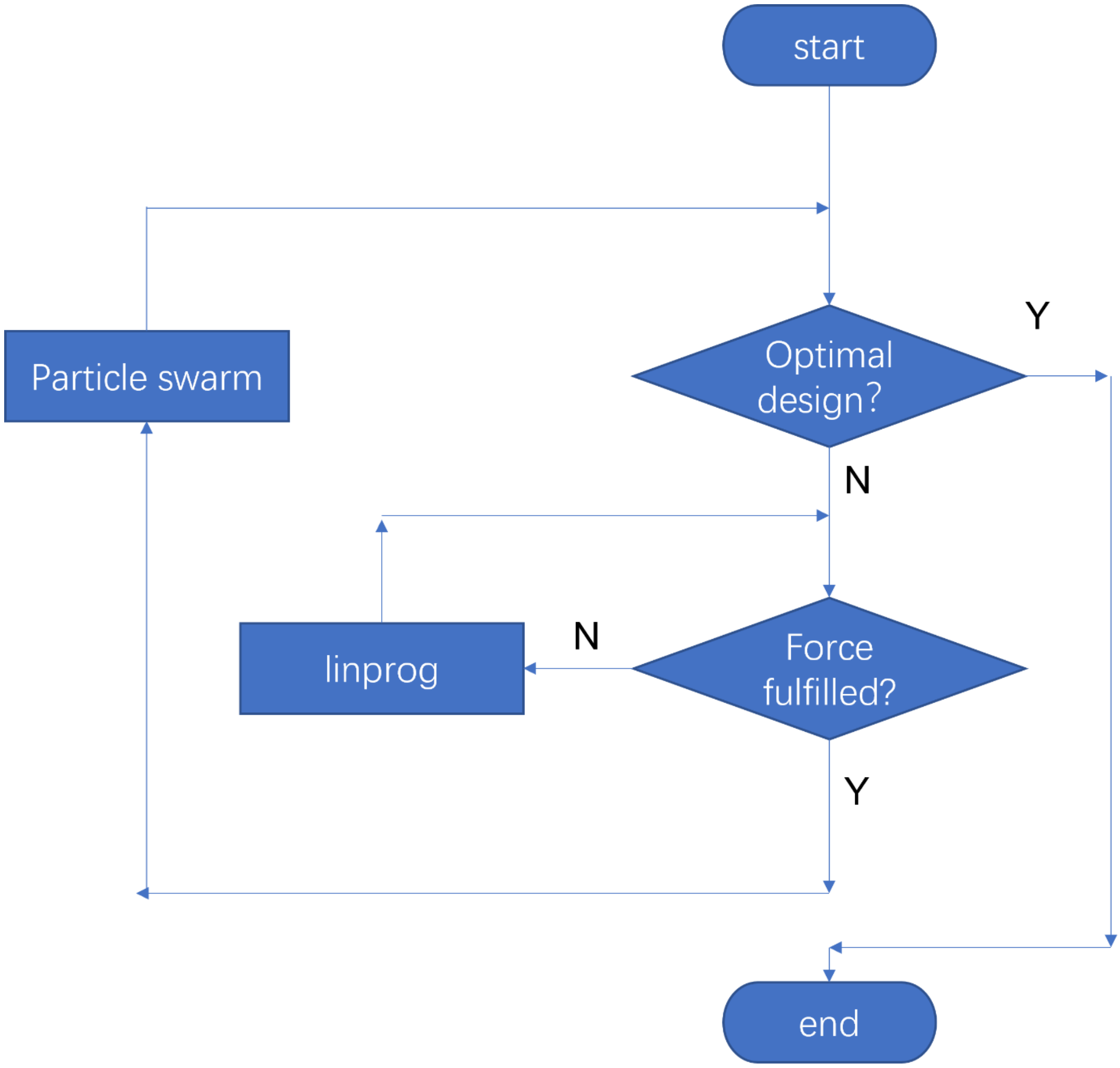
- 2. the link lengths are equal to L/2, so that there are no vacant space inside of the workspace.

- Q4. (Bonus) Considering collision between the robot links and the human ellipse as a non-linear constraint by checking intersection between links and the ellipse [10]

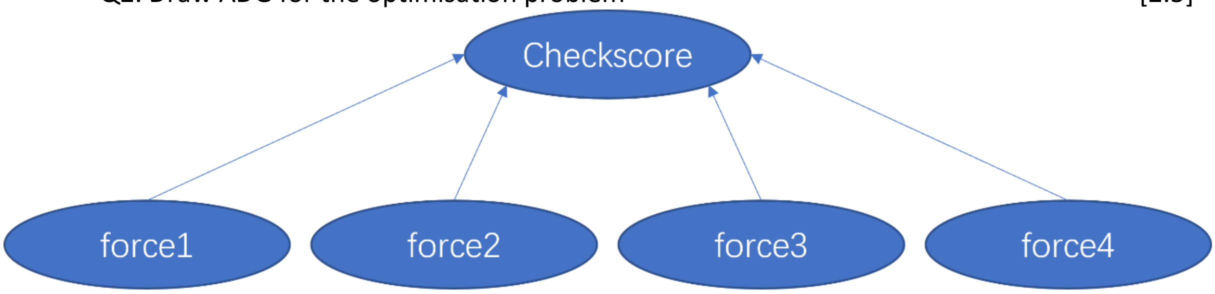
Attachment point optimisation [50]

Working code of the unconstrained problem [25]

- Q1. Draw the flow-chart for the bi-level optimisation problem (design and forces) [2.5]



- Q2. Draw ADG for the optimisation problem [2.5]



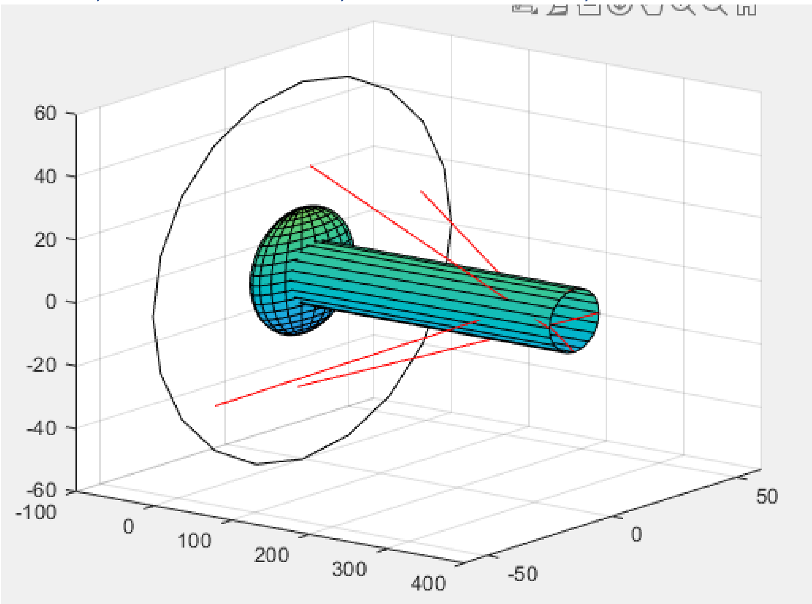
- Q3. Plot objective function value vs iterations of the optimisation [5]

Iteration	f-count	Best f(x)	Mean f(x)	Stall Iterations
0	50	-0.9983	-0.2319	0
1	100	-0.9983	-0.2	0
2	150	-0.9983	-0.2159	1
3	200	-0.9983	-0.2153	2
4	250	-0.9983	-0.2467	3
5	300	-0.9983	-0.2	4
6	350	-0.9983	-0.2	5
7	400	-0.9985	-0.2319	0
8	450	-0.9985	-0.2319	1
9	500	-0.9989	-0.2477	0
10	550	-0.9989	-0.2	1
11	600	-0.9989	-0.2636	2
12	650	-0.9989	-0.2638	3
13	700	-0.9996	-0.2957	0
14	750	-0.9996	-0.2798	1
15	800	-0.9996	-0.295	2
16	850	-0.9996	-0.279	3
17	900	-0.9996	-0.2953	4
18	950	-0.9997	-0.3258	0
19	1000	-0.9997	-0.3596	1
20	1050	-0.9997	-0.309	2
21	1100	-0.9997	-0.2798	3
22	1150	-0.9997	-0.4209	4
23	1200	-0.9997	-0.3733	5
24	1250	-0.9997	-0.3427	6
25	1300	-0.9997	-0.3068	7
26	1350	-0.9999	-0.3277	0
27	1400	-0.9999	-0.3915	1
28	1450	-0.9999	-0.3431	2
29	1500	-0.9999	-0.3272	3
30	1550	-0.9999	-0.3432	4
31	1600	-0.9999	-0.3754	5
32	1650	-0.9999	-0.3596	6
33	1700	-0.9999	-0.4395	7
34	1750	-0.9999	-0.4371	8
35	1800	-0.9999	-0.4232	9
36	1850	-0.9999	-0.4708	10
37	1900	-0.9999	-0.5034	11
38	1950	-0.9999	-0.4549	12
39	2000	-0.9999	-0.518	13
40	2050	-0.9999	-0.4549	14
41	2100	-0.9999	-0.5195	15
42	2150	-0.9999	-0.5513	16
43	2200	-0.9999	-0.4542	17
44	2250	-0.9999	-0.4856	18
45	2300	-0.9999	-0.5672	19

- Q4. Qualitatively how do you choose the termination iterations [2.5]
- Set FunctionTolerence to 1e-5

- Q5. What are the optimal values for the design variables? Add an image of [5]
- the cables and the link.

Optimal values are
[1.02238999590710, 36.9829379965471, 3.63277464059493, 39.4165655454048, 0.144897327264209, 22.0296502289045, 4.64556296796709, 58.0062724444185]



- Q6. Modify the swarm size with at least two different values and repeat the experiment and document how it affects the optimal value and the computation time. [5]

Swarm size	Computation time	Optimal value
10	5.5990	[3.65845364420898, 55.0779650726425, 1.95066329463461, 47.0955850396124, 0, 47.0280057665610, 4.94120459948632, 52.4618253149245]
25	11.7983	[0.0119464648439346, 48.1567837680549, 4.01742059040634, 57.2611686983488, 0.922212737280943, 51.7010400009197, 3.13006567085059, 59.7290017490030]
50	28.7134	[3.02236157123438, 31.9523960822731, 6.10142388847833, 41.9678171795023, 2.46807715144155, 45.9032667033343, 3.66063357466191, 48.8452460916907]

- Q7. What are the function and convergance tolerance for the chosen algorithm in the linear programming and how does it affect the solution? (see linprog documentation) [2.5]
- 1e-8 for 'interior-point-legacy', 1e-7 for 'dual-simplex', 1e-6 for interior-point
- The convergance tolerance affects the permissible error between iterations. And it affects the iteration number and stop conditions.

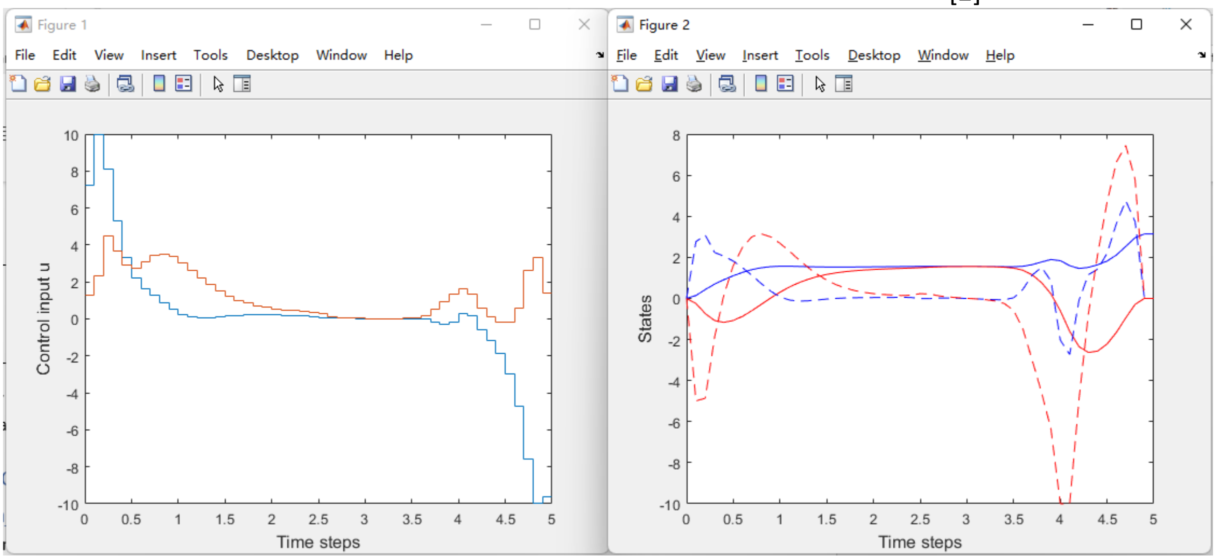
Day-03: Control Synthesis [100]

2R-robot trajectory optimisation [50]

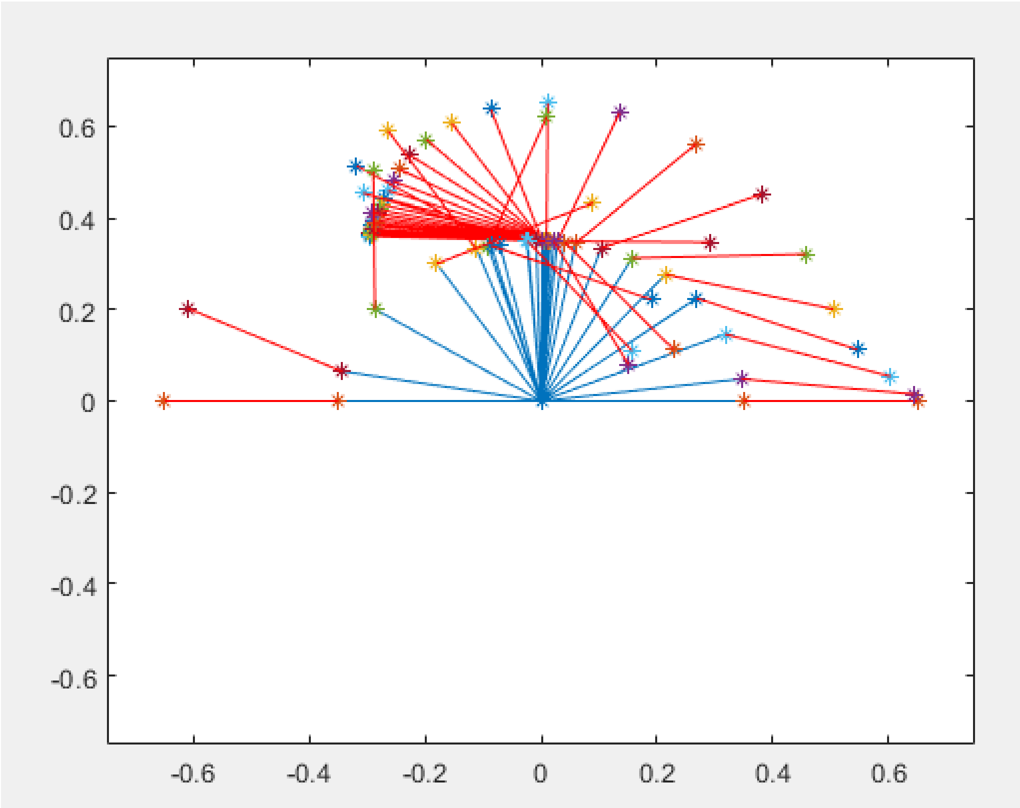
Working code of the robot moving from [0,0,0,0] to [pi,0,0,0] [25]

- Q1. Solve the problem using fmincon and provide reasons for your observations? [4]
  - Document the state and control values obtained for your chosen initial and final state values

[2]



  - Document the motion of the robot as a gif/video [2]



- Q2. Solve the problem using different algorithms available, `sqp`, `active-set`, `interior point` and note the following: [5]
  - Number of iterations [1]
  - Time taken [1]
  - Number of function evaluations [1]
  - Objective function value at optima [1]
  - Difference in the obtained solution by plotting the corresponding states and torques on the same plot [1]

algorithm	Number of iter	time	Nr. Function eval	Obj func optima
Interior-point	70	6.4212	21532	9.338418e+02
active-set	99	7.5390	30173	1515.63
sqp	52	3.6137	16049	9.351342e+02

- Q3. How does the final solution change when: (change at least to two values and document the solution obtained with at least one expected reason for the solution) [7]
  - tf is changed [1]
    the movement time of the 2R robot is changed. When tf is smaller, the control input of the robot is bigger and the time that the robot holds in a unchanged position is shorter. Because the robot wants to get to the destination faster so the control input must be bigger.
  - N is changed [1]
    The discretization number is changed. Fewer steps between t0 and tf are simulated. The computation time is shortened.
  - x0 is changed [1]
    The initial position is changed. This parameter affects the solution greatly.

- xf is changed [1]
  The final position is changed. This parameter affects the solution greatly.

- g is changed [1]
  The gravitational field is changed. This affects the control input.

- l1, l2, m1, m2 are changed [1]
  The design parameters of the robot is changed. These parameters affect the control input.

- Difference in the obtained solution by plotting the corresponding states and torques on the same plot [1]

- Q4. Give a better initial guess for the states and see how the simulation time and iterations and the solution quality changes (Ex. Use any interpolation between initial and final states as initial guess) [2]
- I use a rand(N,n) as initial guess and the simulation time is shorter and the iterations are less there is no obvious difference in solution quality.

- Q5. Modify the bounds on states and control and see how the behavior of the solution changes? can you find a solution for a small value of control bounds? Why? [2]

- If the bounds on states and control is smaller, the robot will continue to move and have little time for pausing in one position. For a small value of control bounds, there is no guarantee that we can find a solution.

- Q6 (Bonus). Modify the discretisation scheme from Euler to Hermite-Simphson or another and document the nature of the solution. See how system behavior changes [2.5]
- Q7 (Bonus). Give the obtained control input to the system and observe the system performance? Does the robot reach the final position as expected? [2.5]

Musculo-skeletal robot trajectory optimisation [50]

Working code for the musculo-skeletal robot [20]

- Q1. How does modifying these variables effect the solution obtained? [5]
  - tf is changed [1]
    the movement time of the muscular-skeletal robot is changed. When tf is smaller, the control input of the robot is larger and the time that the robot holds in a unchanged position is shorter. Because the robot wants to get to the destination faster so the control input must be bigger.

  - N is changed [1]
    The discretization number is changed. Fewer steps between t0 and tf are simulated. The computation time is shortened.

  - x0 is changed [1]
    The initial position is changed. This parameter affects the solution greatly.

  - xf is changed [1]
    The final position is changed. This parameter affects the solution greatly.

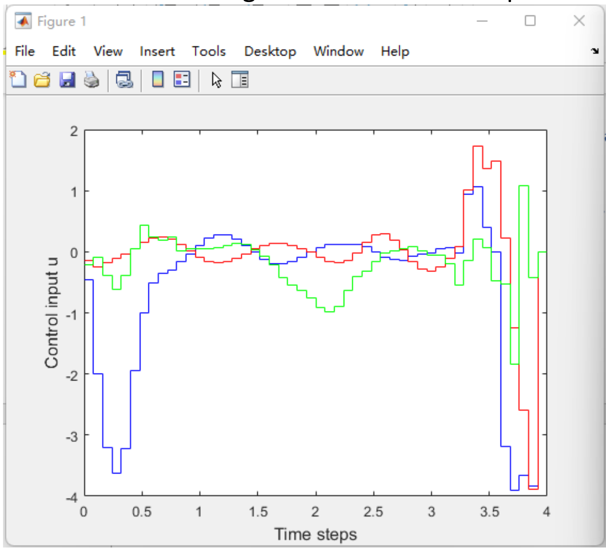
  - g is changed [1]

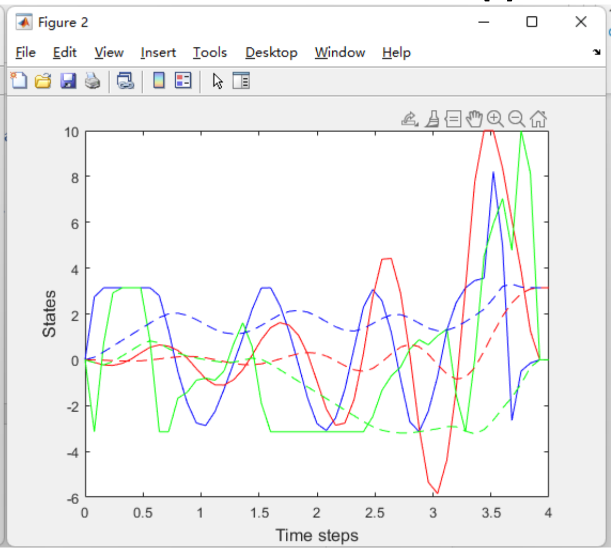
The gravitational field is changed. This affects the control input.

- l1, l2, m1, m2 are changed [1]

  The design parameters of the robot is changed. These parameters affect the control input.

- Q2. Plot the resulting states and control input [3]



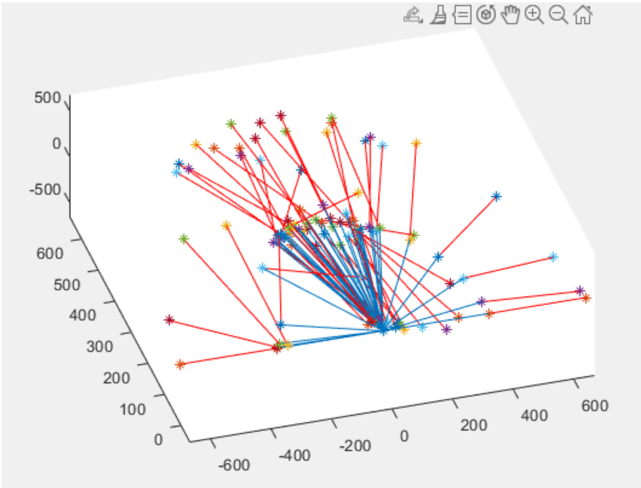


- Q3. Write a code similar to `robolinplot` to visualise the resulting trajectory of the musculo-skeletal robot

```
th1 = z(:,4);
th2 = z(:,5);
th3 = z(:,6);
%read l1 and l2 from the previous code
l1 = init.l1;
l2 = init.l2;
plot_bounds = l1+l2+0.1;

%The coordinates of the links of each of the manipulator
x1 = l1*cos(th1);
y1 = l1*sin(th1);
z1 = l1*sin(th2);
x2 = l1*cos(th1)+l2*cos(th1+th3);
y2 = l1*sin(th1)+l2*sin(th1+th3);
z2 = l1*sin(th2)+l2*sin(th2+th3);
figHandle = figure();
for i=1:init.N
    A = [0 x1(i)];
    B = [0 y1(i)];
    C = [0 z1(i)];
    %   subplot(1,3,1);
    plot3(A,B,C,'*')
    axis([0 plot_bounds -plot_bounds plot_bounds -plot_bounds plot_bounds])
    hold on
    line(A,B,C)
    hold on
    A2 = [x1(i) x2(i)];
    B2 = [y1(i) y2(i)];
    C2 = [z1(i) z2(i)];
    %   subplot(1,3,1);
```

```
-       plot3(A2,B2,C2,'*')
-
-       axis([-plot_bounds plot_bounds 0 plot_bounds -plot_bounds plot_bounds])
-       hold on
-       line(A2,B2,C2,'Color','red')
-       pause(0.1);
-   end
```



- [5]

- Q4. Repeat Q1-Q2, Q3.1-3.4, Q4, Q5 from the previous problem [17]

Day-04: Co-design [50]

2R-planar robot: Bi-level optimisation [50]

Working code of the bi-level optimisation [25]

- Q1. What is the trivial solution for link lengths when there are no constraints? Why? [5]
- L1 = l2 = 0, m1 = m2 = 0, because in this case the lower level optimization which is to optimize the energy from x0 to xf equal to zero.
- Q2. Add a cost for control input (J), workspace, cost of the robot with appropriate weighting to get a non-trivial solution? Qualitatively explain the solution. [10]

```
-   % Compute the objective function value
-   J = 0;
-   for i = 1:size(u)
-       J = J + u(i)^2;
-   End
-
```

- Solution:
- [0.203543195821526, 0.479105718516851, 0.112476786541377, 0.174243258914018]
-
- I only optimize the energy and don't care for workspace. The result for this choice is that l1 is shorter than l2 and m1 is smaller than m2. In this case the actuator can use minimum energy to swing up the robot against the gravity to the desired position.
-
- Q3. Choose different weighting between cost functions and document the result of the mechanical design variable values [5]

  This was not attempted in the class.

- Q4. (Bonus) Add strength constraints and re-optimise the problem and document and qualitatively explain the solution [5]

Total score possible: 250

But since bonus is included you do not have to solve all the questions.