SAINT MARTIN'S UNIVERSITY

Project: Four Bar Linkage

Alexander Benson

October 13, 2020

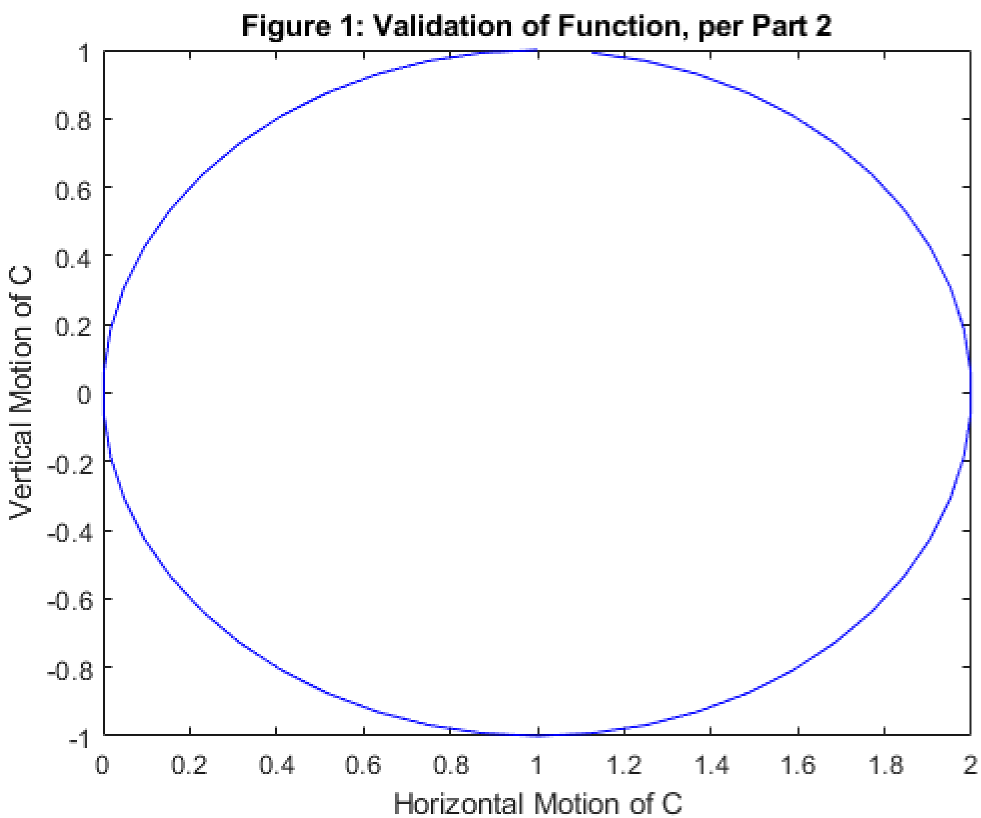
1 Project

1.1 Part 1

Completion of Part 1 is demonstrated by parts 2 and 3.

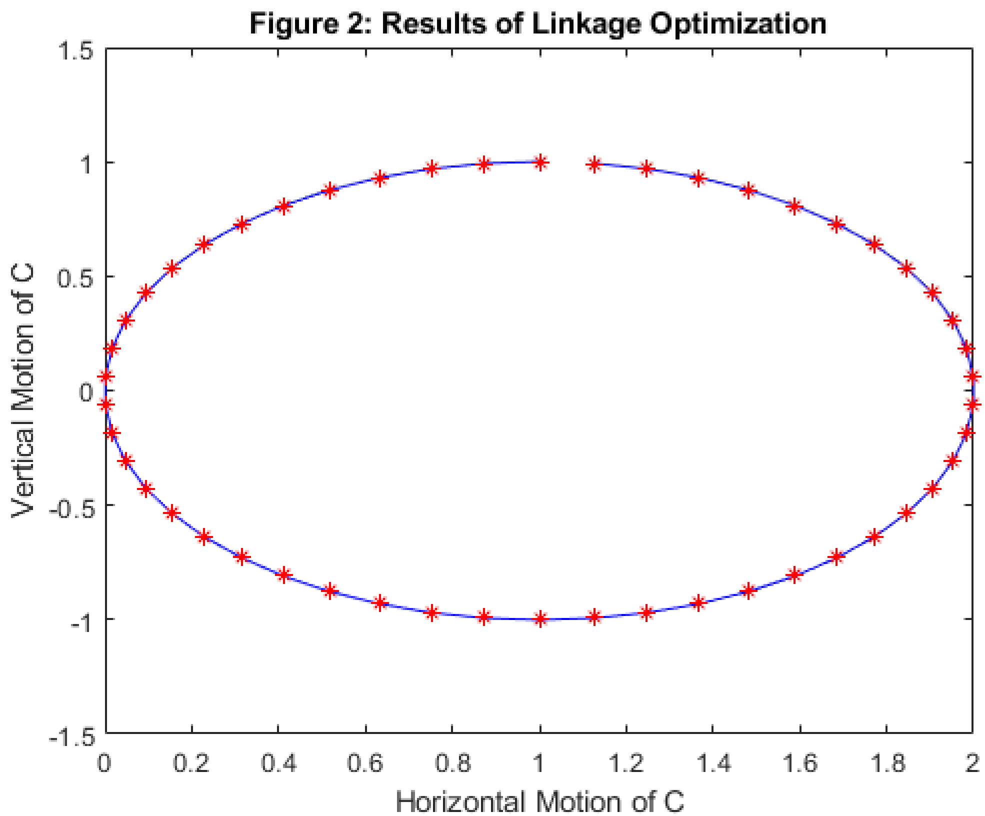
Special note goes to Bulatović and Dordević [1] for the derivation of part 1.

1.2 Part 2



1.3 Part 3

Optimal Linkage Lengths are as follows:
a: 1, h: 1
b: 1, g: 1



1.4 Constraint Derivation

The four bar linkage derived is explicitly a crank-crank system. To be grashof compliant such a system must meet the following equations:

- The term $T_1 = g + h - a - b$ must be negative, therefore $g + h \leq a + b$
- The term $T_2 = b + g - a - h$ must be negative, therefore $b + g \leq a + h$
- The term $T_3 = b + h - a - g$ must be positive, therefore $b + h \geq a + g$

To make these constraints compatible with FMINCON, they are rearranged to:

$$C = \begin{bmatrix} g + h - a - b \\ b + g - a - h \\ a + g - b - h \end{bmatrix} \leq 0$$

1.5 Lessons Learned

This project was a reminder of the importance of understanding the team you have:

- How they handle ambiguity
- The quality, rate, and timeliness of their work and who can be relied on to work independently

- Their communication style and skills - Their willingness to make decisions without consulting the team: the date extension surprised me given that I thought the team more than capable of solving the project given a work day's worth of modest effort.

The difficulty of deriving a formula/analysis function was reasonable: in my experience taking complicated concepts and quantifying them is the majority of any real-world optimization project.

2 Bibliography

[1] R. R. Bulatović and S. R. Dordević, "On the optimum synthesis of a four-bar linkage using differential evolution and method of variable controlled deviations," Mechanism and Machine Theory, vol. 44, no. 1, pp. 235–246, Jan. 2009, doi: 10.1016/j.mechmachtheory.2008.02.001.

3 Code

[ ]:
```
clear;clc;

%%%%%%%%%%%% Main File %%%%%%%%%%%%
%%%%%Graduate Four-Bar Linkage%%%%%%

%% Design Variables
x0 = [1,2,2,2];

LB = [1,1,1,1];
UB = [4,4,4,4];

%% Parameter Completion
a = []; b = []; aeq = []; beq = [];


%Optimize
options = optimset('Display','off','MaxFunEvals',5e4,'MaxIter',1e6);
[xopt, fopt] = fmincon(@objfun, x0, a, b, aeq, beq, LB, UB, @nonlincon, options);

C = fourbarfun(xopt);

objPath = importdata('goal_traj.mat');
optC = [objPath(:,1), objPath(:,2)].';

fprintf("Optimal Linkage Lengths are as follows:\n")
fprintf("a: %3.3g, h: %3.3g\n b: %3.3g, g: %3.3g", xopt(1),xopt(2),...
    xopt(3),xopt(4))

plot(C(1,:), C(2,:), 'b-',optC(1,:),optC(2,:),'r*');

legend('Design', 'Desired')
title('Figure 2: Results of Linkage Optimization')
```

```
xlabel('Horizontal Motion of C')
ylabel('Vertical Motion of C')
```

[ ]:
```
function [out] = objfun(x)
% The Objective Function of the Four Bar Linkage Project

C = fourbarfun(x);
objPath = importdata('goal_traj.mat');
optC = [objPath(:,1), objPath(:,2)].';


out = 0.5*sum((optC(1)-C(1)).^2)+0.5*sum((optC(2)-C(2)).^2);
end
```

[ ]:
```
function [C,Ceq] = nonlincon(x)
%Non Linear Constraints for the Four Bar Linkage Assignment
a = x(1);%Length of Input Link
h = x(2);%Length of Floating Link
b = x(3);%Length of Output Link
g = x(4);%Length of Ground Link


C(1) = g+h-a-b;
C(2) = b+g-a-h;
C(3) = -b-h+a+g;
Ceq=[];
end
```

[ ]:
```
function current_traj = fourbarfun(x)
% A function describing the planar motion of a four bar linkage
% Per MME 523, Project 1, P2

%% Defining Links
a = x(1);%Length of Input Link
h = x(2);%Length of Floating Link
b = x(3);%Length of Output Link
g = x(4);%Length of Ground Link


%% Predefined Coordinates
A = [0,0];
B = [g,0];

%% Defining Rotation
steps = 50;
omega = 1;
delta = (2*pi)/(omega*steps);
```

```
alpha = linspace(pi/2, 5*pi/2-delta, steps);

%% Pathing
C = zeros(2,steps);
D = zeros(2,steps);


for index = 1:steps
    alpha_Val = alpha(index);
    D(:,index) = a*[cos(alpha_Val);sin(alpha_Val)];

    s = sqrt((B(1)- D(1,index))^2+(B(2)-D(2,index))^2);
    %At -90 and 90, arctan is asymptotic, positive inf -> 90degrees


    zeta = atan(D(2,index)/(B(1)-D(1,index)));
    gamma = acos((s^2+b^2-h^2)/(2*s*h));

%% Atan must be inverted for negative denom
    if B(1)-D(1,index)<= 0
       zeta = -zeta;
    end

%% Gamma must be inverted when B passes below the ground link
    if D(2, index) <= 0
       gamma = -gamma;
    end

    beta = pi-gamma-zeta;
    C(:,index) = [A(1)+B(1)+b*cos(beta);A(2)+B(2)+b*sin(beta)];
end
current_traj = C;
end
```