2015

Three-Position Four-Bar Linkage Mechanism Synthesis, Static Balancing and Optimization of Automotive Engine Hood

Onur Denizhan
Lehigh University

Three-Position Four-Bar Linkage Mechanism Synthesis, Static Balancing and Optimization of Automotive Engine Hood

By
Onur Denizhan

A Thesis
Presented to the Graduate and Research Committee
Of Lehigh University
In Candidacy for the Degree of
Master of Science
In
Mechanical Engineering and Mechanics

Lehigh University
July 2015

CERTIFICATION OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

Date

Thesis Advisor, Dr. Meng-Sang Chew

Chairperson of Department, Dr. Gary Harlow

ACKNOWLEDGEMENTS

First of all, I was extremely fortunate to have Professor Meng-Sang Chew as my M.Sc. advisor. He was not only my research advisor but also the source of confidence and motivation. I would like to thank Professor Chew for his guidance, direction, patience, and advice. I believe, without doubt, that this work wouldn't have been completed without his guidance. His efforts were invaluable towards my academic progress.

There are many people who have helped me accomplish my goal of achieving my Master's Degree in Mechanical Engineering, to whom I will forever be indebted. First, I would like to thank Chunjian Pan who has attempted to help me MATLAB coding and is my one of the best friend at Lehigh. Also, I would like to thank Cosan Dakiran and Emre Akoz who have helped me during my research. In addition, I appreciate to Zeki Okan Ilhan, Fatih Aktas, Soner Cinoglu, Fatma Ayancik Cinoglu, Dana Reuther, Denis Aslangil, Mustafa Usta, Doruk Erdem Yunus, Jaharah Yunus, Laura Mejia-Suarez and Cuneyt Ferahlar for their friendship at Lehigh.

Lastly but not the least, I would like to thank my family for their love, continuous encouragement and believing in me despite the miles between us. I am grateful to my father, Nezir Denizhan, for his endless compassion and encouragement in every decision I made in my life; he has been my role model as a successful person. I am indebted to my mother, Sengul Denizhan, for her care and love and heartfelt thanks and love to my older brother, Gokhan Denizhan.

TABLE OF CONTENTS

Acknowledgements .... **iv**

List of Tables .... **viii**

List of Figures .... **ix**

Abstract **1**

Chapter 1: Introduction **2**

1.1.Introduction 2

1.2.Description of Project 2

1.3.Optimization 4

Chapter 2: Extension Spring Application **6**

2.1.Relative Background Information 6

2.2.First Link Application 7

2.2.1. Formulization of Extension Spring for First Link Application 8

2.2.2. Results 11

2.3.Second Link Application 14

2.3.1. Formulization of Extension Spring for Second Link Application 15

2.3.2. Results 19

Chapter 3: Compression Spring Application **21**

3.1.Relative Background Information 21

3.2.First Link Application 22

3.2.1. Formulization of Compression Spring for First Link Application 23

3.2.2. Results 27

3.3.Second Link Application..31

3.3.1. Formulization of Compression Spring for Second Link Application 32

3.3.2. Results 35

Chapter 4: Torsion Spring Application 38

4.1.Relative Background Information...38

4.2.First Link Application 39

4.2.1. Formulization of Torsion Spring for First Link Application...........................40

4.2.2. Results 43

4.3.Second Link Application..46

4.3.1. Formulization of Torsion Spring for Second Link Application 47

4.3.2. Results 50

Chapter 5: Conclusion and Potential Future Work **52**

5.1. Conclusion..52

5.1.1. First Link Applications...52

5.1.2. Second Link Applications 53

5.2. Potential Future Work 55

Chapter 6: References **56**

Chapter 7: Appendices **58**

Appendix A: Automotive Engine Hood Three-Position Four-Bar Linkage Mechanism Synthesis MATLAB Codes 58

Appendix B: Extension Spring Optimization MATLAB Codes for First Link 62

Appendix C: Compression Spring Optimization MATLAB Codes for First Link 68

Appendix D: Torsion Spring Optimization MATLAB Codes for First Link................74

Appendix E: Extension Spring Optimization MATLAB Codes for Second Link........80

Appendix F: Compression Spring Optimization MATLAB Codes for Second Link...86

Appendix G: Torsion Spring Optimization MATLAB Codes for Second Link 92

Biography...**98**

LIST OF TABLES

Table 2-1 Design variables results after optimization for first link 12

Table 2-2 Design variables results after optimization for second link 20

Table 3-1 Design variables results before and after optimization for compression spring on the first link 30

Table 3-2 Design variables results before and after optimization for compression……...37

Table 4-1 Design variables results before and after optimization for torsion spring…….45

Table 4-2 Design variables results before and after optimization for torsion spring…….51

Table 5-1 Compression spring, extension spring and torsion spring design variables after optimization for first link…………………………………………………………………53

Table 5-2 Compression spring, extension spring and torsion spring design variables after optimization for second link……………………………………………………….......54

LIST OF FIGURES

Figure 1-1 Vehicle engine hood mechanism 3

Figure 1-2 Mechanism three position linkage synthesis 5

Figure 2-1 Extension spring first link application 7

Figure 2-2 Force with θ and without any spring on the first link 12

Figure 2-3 Force with θ after optimization for the first link………………………..……….13

Figure 2-4 Extension spring second link application 14

Figure 2-5 Force with θ and without any spring on the second link 19

Figure 2-6 Force with θ after optimization for the second link 20

Figure 3-1 Compression spring first link application 22

Figure 3-2 Force with θ with initial values compression spring on the first link 28

Figure 3-3 Force with θ after optimization compression spring on the first link 29

Figure 3-4 Compression spring second link application.. 31

Figure 3-5 Force with θ with initial values compression spring on the second link 36

Figure 3-6 Force with θ after optimization compression spring 37

Figure 4-1 Torsion spring first link application 39

Figure 4-2 Force with θ with initial values torsion spring on the first link 43

Figure 4-3 Force with θ after optimization torsion spring on the first link 44

Figure 4-4 Torsion spring second link application 46

Figure 4-5 Force with θ with initial values torsion spring on the second link 50

Figure 4-6 Force with θ after optimization torsion spring on the second link 51

ABSTRACT

Vehicles need to have regular maintenance and repair so that hoods need to be easy to opened and closed. With the addition of more women car owners, this requirement that automobile hoods requiring lower opening and closing efforts become even more important. This research looks into vehicle engine hood mechanisms that incorporate a pin go to assist in the opening of the hood. The design of such mechanisms as divided into two steps. The first is carried out using three position synthesis. In the second step, different types of springs such as extension, compression and torsion springs are applied and optimization procedures is used to determine the various location for the springs and the spring parameters to minimize the effort to raise the hood. A sensitivity analysis is also carried out on the best of the three spring types and configurations to determine its suitability for practical implementation.

CHAPTER 1: INTRODUCTION

1.1.Introduction

An automotive engine hood is to be designed so that it stays at various open positions. This is becoming more important because more cars are now owned by women who have to take of simple maintenance. Automobile engine hoods may be too heavy for some and a mechanism is needed to assist in opening and keeping the hood in an open position.

This design requires a spring wherein, it sores potential energy that is given up by the closing of the hood and releases it back when the hood is open. What is required and needed in the design is to locate the attachment points of the spring on the mechanism that is optimum in opening the hood and in keeping the hood in a range of stable open positions. This will require the interaction between mechanism synthesis in conjunction with optimization methods.

1.2. Description of Project

There are many kinds of mechanisms for automotive engine hoods that can help to keep engine hoods at different open positions. There are also several kinds of components in hood mechanisms such as springs, dampers or props. One basic mechanism which is used during this study is shown Figure 1.1.

In this study, three kinds of spring: extension spring, compression spring and torsion spring are applied on first link and second link separately. The basic goal is to exert a minimum force when someone opens or closes the engine hood.

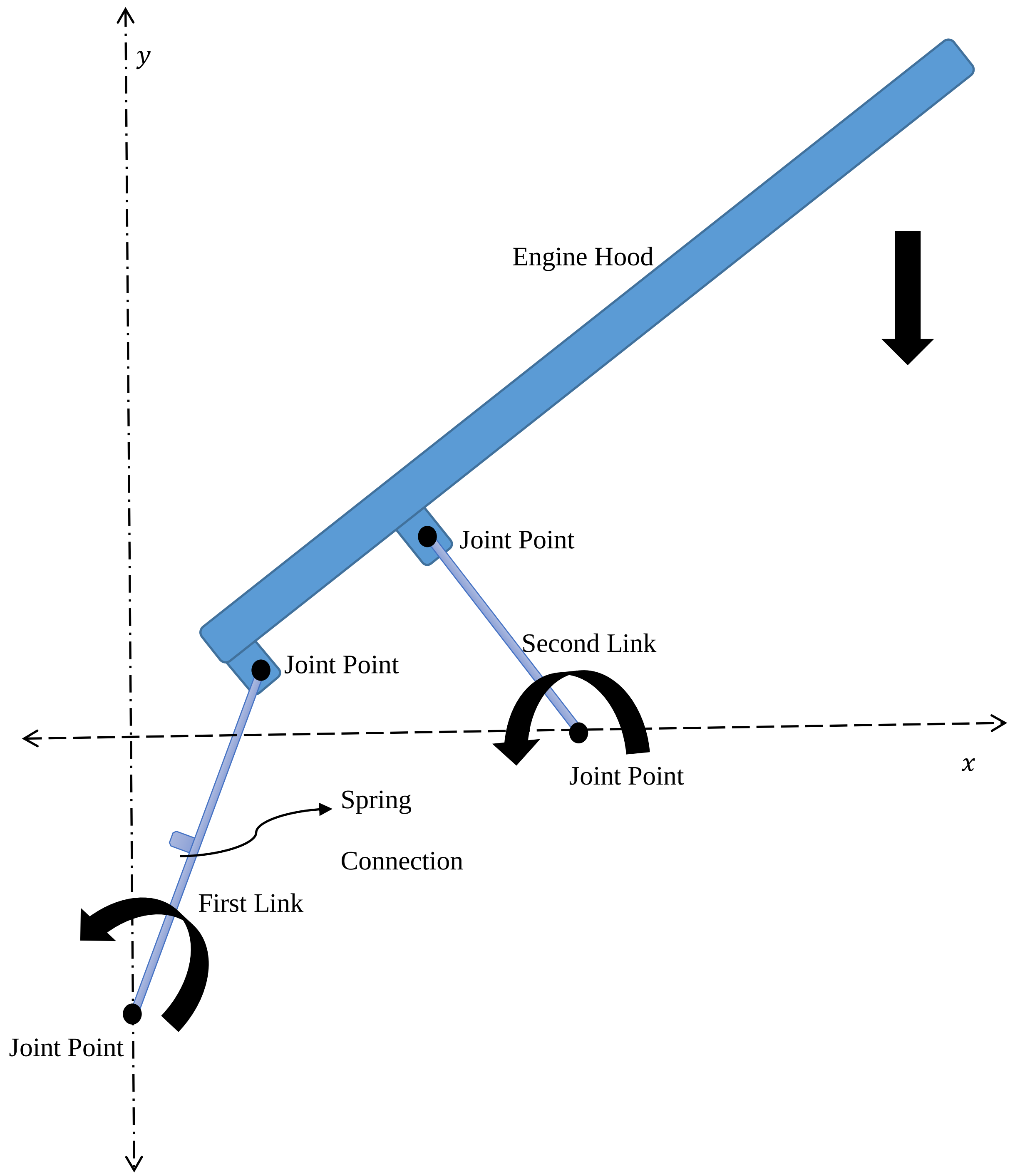

Figure 1-1 Vehicle engine hood mechanism

Both of link motions and hood motions are shown in by the direction of arrows in Figure 1.1.when hood is closing. Before optimization of the linkage mechanism, rigid-body motion synthesis is required [15]. Three position synthesis for such a four-bar linkage is shown in Figure 1.2, for the open, closed and intermediate positions of the hood.

1.3 Optimization

Optimization is an essential part of design activity in all major disciplines such as location, language or product development [17]. Optimization is described as the process of searching for the best solution in accordance to a given criterion, in conjunction with a given mathematical model of the system. In this study, different formulations are described in each chapter for different springs at various locations for optimization process.

In this study, the Sequential Quadratic Programming (SQP) Nonlinear Optimization Method as provided in the MATLAB Optimization Toolbox is used. SQP methods represent the state of the art in nonlinear programming methods [16] and is one of the most successful algorithms for the numerical solution of constrained nonlinear optimization problems. Moreover, Lagrange multipliers and sensitivity analysis are applied springs to have the best optimal results. Lagrange multipliers and sensitivity analysis are an optimization method and are useful in some conditions to find optimal values in nonlinear optimization process [18]. That method are used in particular cases and conditions in nonlinear optimization problems.

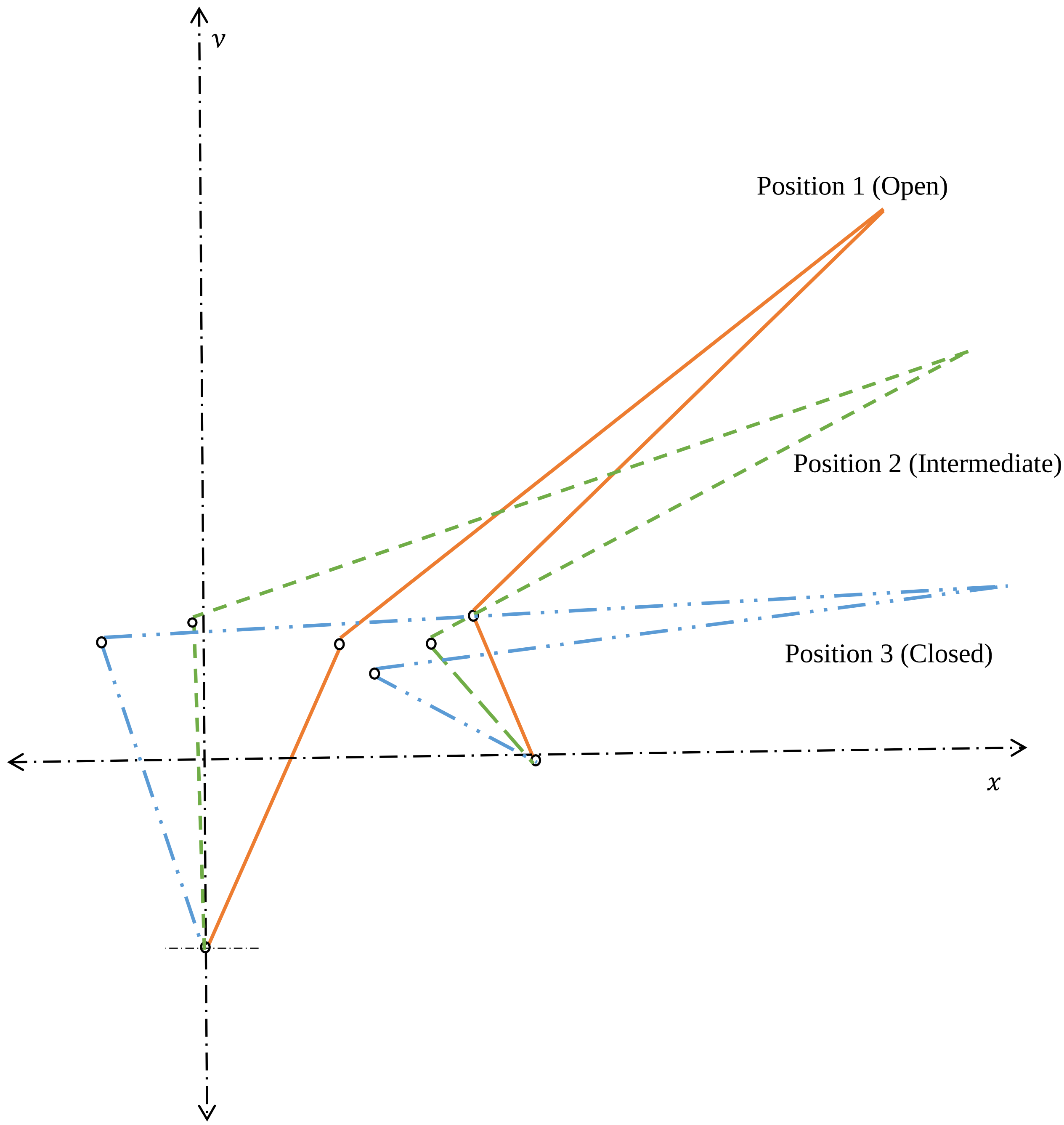

Figure 1-2 Mechanism three position linkage synthesis

CHAPTER 2: EXTENSION SPRING APPLICATION

2.1. Relative Background Information

In the design of mechanical systems that incorporates such components as gear, cams or shafts. The helical extension spring is among one of the most component in use today. Springs that absorb and store energy by offering resistance to a pulling force are known as extension springs. Various types of ends are used to attach this type of spring to the source of the force. [6].

Literature on numerical methods provides several examples of extension spring optimization. Deb and Goyal [1], Kannan and Kramer [2] and Sandgren [3] used springs to illustrate their optimization methods with mixed variables. The problems that were used, were oversimplified mathematical model. Parades, Sartor and Masclet [4] also looked at an optimization problem, using industrial software package, with given specifications for extension springs.

In some cases, extension springs are used for static balancing of the weights in a given mechanical system that operates in a vertical plane. [5]. In addition, the complexity of balancing systems with springs may come from the need of using zero-free length springs. [5].

All of these studies show that helical extension springs are frequently used in mechanical systems for static balancing or dynamic balancing, and have been applied to all kinds of applications.

2.2. First Link Application

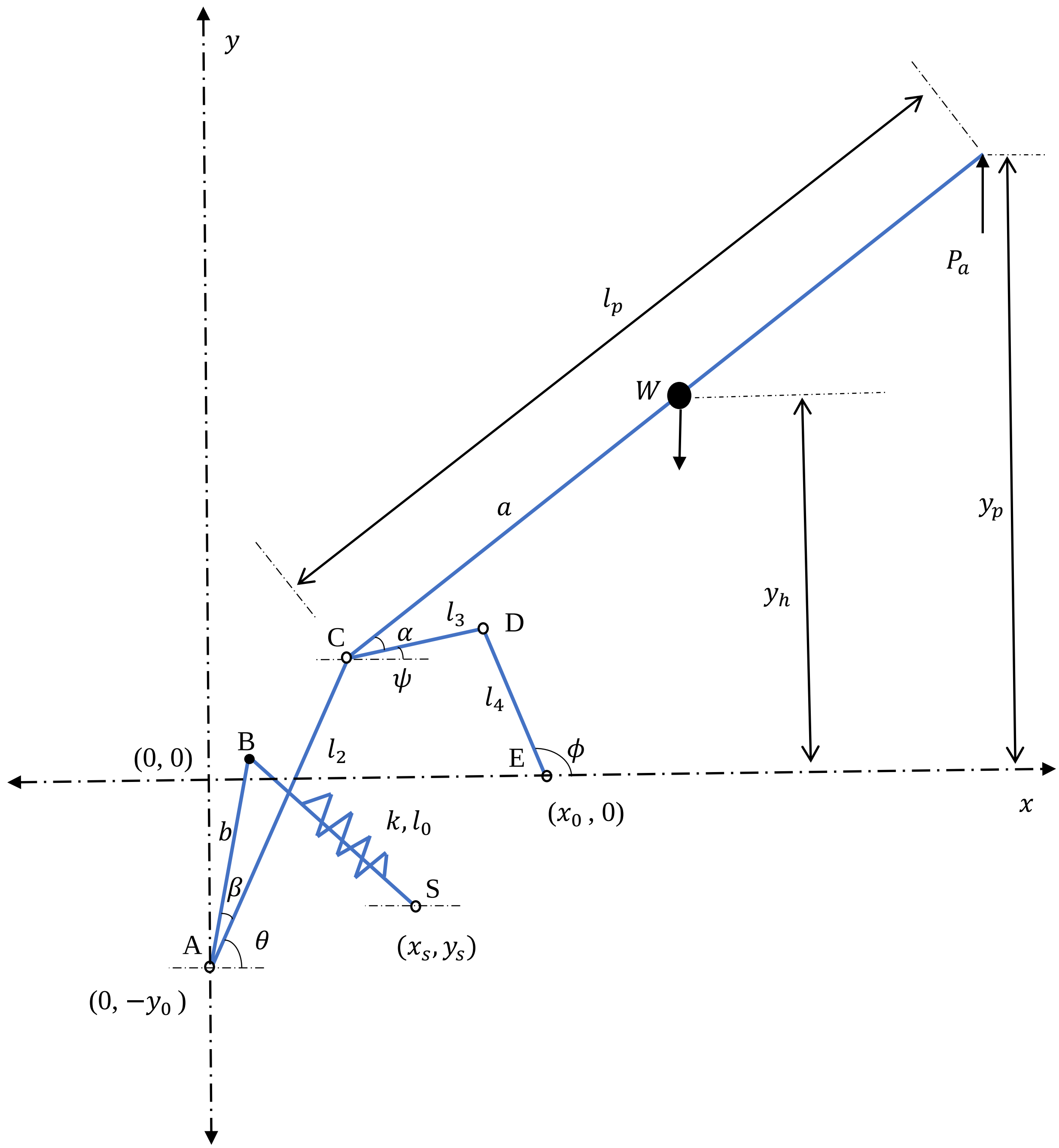

Figure 2-1 Extension spring first link application

The system shown in Figure 2.1 consists of an extension spring that is attached to first link [AC] of linkage. During the operation of this hood linkage, the extension spring stores potential energy that is given up when the closing of the hood, and is released back when the hood is opening. In this configuration, the second link [DE] is left without any attachment to any other mechanical assistance.

During optimization process k, β, l_0, b, x_s and y_s are specified as design variables. These are shown Figure 2.1 for an extension spring. k is the constant of extension spring. β is the angle between links [AC] and [AB]. l_0 is length of the extension spring, b is length of [AB] and x_s and y_s are starting coordinate positions of the extension spring attachment points.

2.2.1. Formulization of Extension Spring for First Link Application

All of variables are shown in Figure 2.1. In addition, F_ψ and F_ϕ are velocity coefficients of system.

Loop equations:

$$l_2 e^{i\theta} + l_3 e^{i\psi} + \left(-l_4 e^{i\phi}\right) - x_0 - i y_0 = 0 \qquad (2.1)$$

Reel Part:

$$l_2 \cos\theta + l_3 \cos\psi - l_4 \cos\phi - x_0 = 0 \qquad (2.2)$$

Imaginary Part*:*

$$l_2 \sin\theta + l_3 \sin\psi - l_4 \sin\phi - y_0 = 0 \qquad (2.3)$$

Velocity coefficients:

$$-l_2 \sin\theta\,\delta\theta - l_3 \sin\psi F_\psi\,\delta\theta + l_4 \sin\phi\, F_\phi \delta\theta = 0 \tag{2.4}$$

$$l_2 \cos\theta\,\delta\theta + l_3 \cos\psi\, F_\psi \delta\theta - l_4 \cos\phi\, F_\phi \delta\theta = 0 \tag{2.5}$$

Length of spring: l_s

$$l_s^2 = (b_x - x_s)^2 + (b_y - y_s)^2 \tag{2.6}$$

$$l_s = \sqrt{(b_x - x_s)^2 + (b_y - y_s)^2} \tag{2.7}$$

$$b_x = b\cos(\theta + \beta) \tag{2.8}$$

$$b_y = b\sin(\theta + \beta) \tag{2.9}$$

Y position of c. g. of hood: y_h

$$y_h = -y_0 + l_2 \sin\theta + a\sin(\psi + \alpha) \tag{2.10}$$

Y position of applied force: y_p

$$y_p = -y_0 + l_2 \sin\theta + l_p \sin(\psi + \alpha) \tag{2.11}$$

Virtual displacement

a. δl_s:

$$2l_s\,\delta l_s = 2(b_x - x_s)(-b\sin(\theta+\beta)\delta\theta) + 2(b_y - y_s)(b\cos(\theta+\beta)\delta\theta) \tag{2.12}$$

$$\delta l_s = \left[\frac{-b(b_x - x_s)\sin(\theta+\beta) + b(b_y - y_s)\cos(\theta+\beta)}{l_s}\right]\delta\theta \tag{2.13}$$

b. δy_h:

$$\delta y_h = l_2\cos\theta\,\delta\theta + a\cos(\psi+\alpha)\left(\frac{\delta\psi}{\delta\theta}\right)\delta\theta \tag{2.14}$$

c. δy_p:

$$\delta y_p = l_2\cos\theta\,\delta\theta + l_p\cos(\psi+\alpha)\left(\frac{\delta\psi}{\delta\theta}\right)\delta\theta \tag{2.15}$$

Conservative potential

a. Hood:

$$V_h = W y_h \tag{2.16}$$

$$\delta V_h = W\delta y_h \tag{2.17}$$

b. Spring*:*

$$V_s = \frac{1}{2}k[l_s - l_0]^2 \tag{2.18}$$

$$\delta V_s = k[l_s - l_0]\delta l_s \tag{2.19}$$

Virtual work and force

$$\delta W = P_a \delta y_p - \delta V_s - \delta V_h \tag{2.20}$$

$$= P_a \delta y_p - k(l_s - l_0)\delta l_s - W \delta y_h \tag{2.21}$$

$$\Big[P_a\big\{l_2 \cos\theta + l_p \cos(\psi + \alpha)F_\psi\big\} - W\big\{l_2 \cos\theta + a \cos(\psi + \alpha)F_\psi\big\} - k(l_s - l_0)\frac{b}{l_s}\big\{(b_x - x_s)\sin(\theta + \beta) + \big(b_y - y_s\big)\cos(\theta + \beta\big\}\Big] = 0 \tag{2.22}$$

$$P_a = \frac{W\big\{l_2 \cos\theta + a\cos(\psi + \alpha)\,F_\psi\big\} + k(l_s - l_0)\frac{b}{l_s}\big\{(b_x - x_s)\sin(\theta + \beta) + \big(b_y - y_s\big)\cos(\theta + \beta)\big\}}{l_2 \cos\theta + l_p \cos(\psi + \alpha)F_\psi} \tag{2.23}$$

2.2.2. Results

Firstly, k was taken as $0\ N/m$ in Equation (2.23). It means that there are no springs in system that is shown in Figure 2.1. The goal of this that differences in the force application to lift the hood between a case with extension spring on first link [AC] and are without.

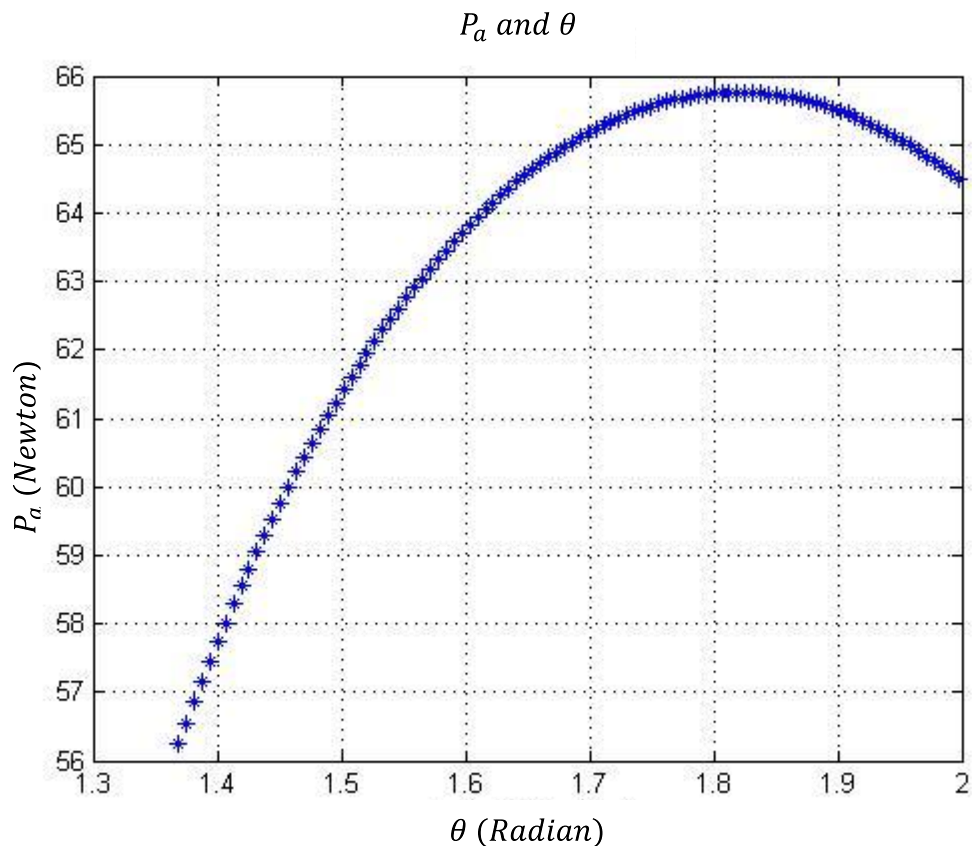

Figure 2-2 Force with θ and without any spring on the first link

After the optimization process, all design variables are shown in below in Table 2.1.

Table 2-1 Design variables results after optimization for first link

k (N/m)	β $(radian)$	l_0 (m)	b (m)	x_s (m)	y_s (m)
666.93	-0.018	0.05	0.20	0.40	0

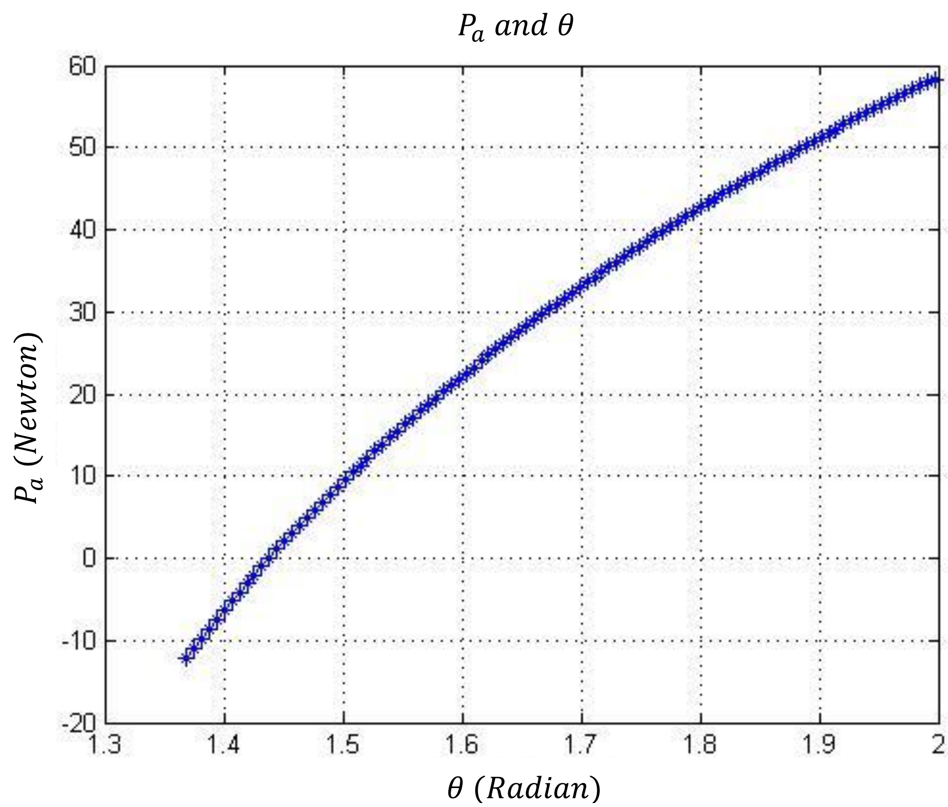

Figure 2-3 Force with θ after optimization for the first link

All the results are discussed together in Chapter 5.

2.3. Second Link Application

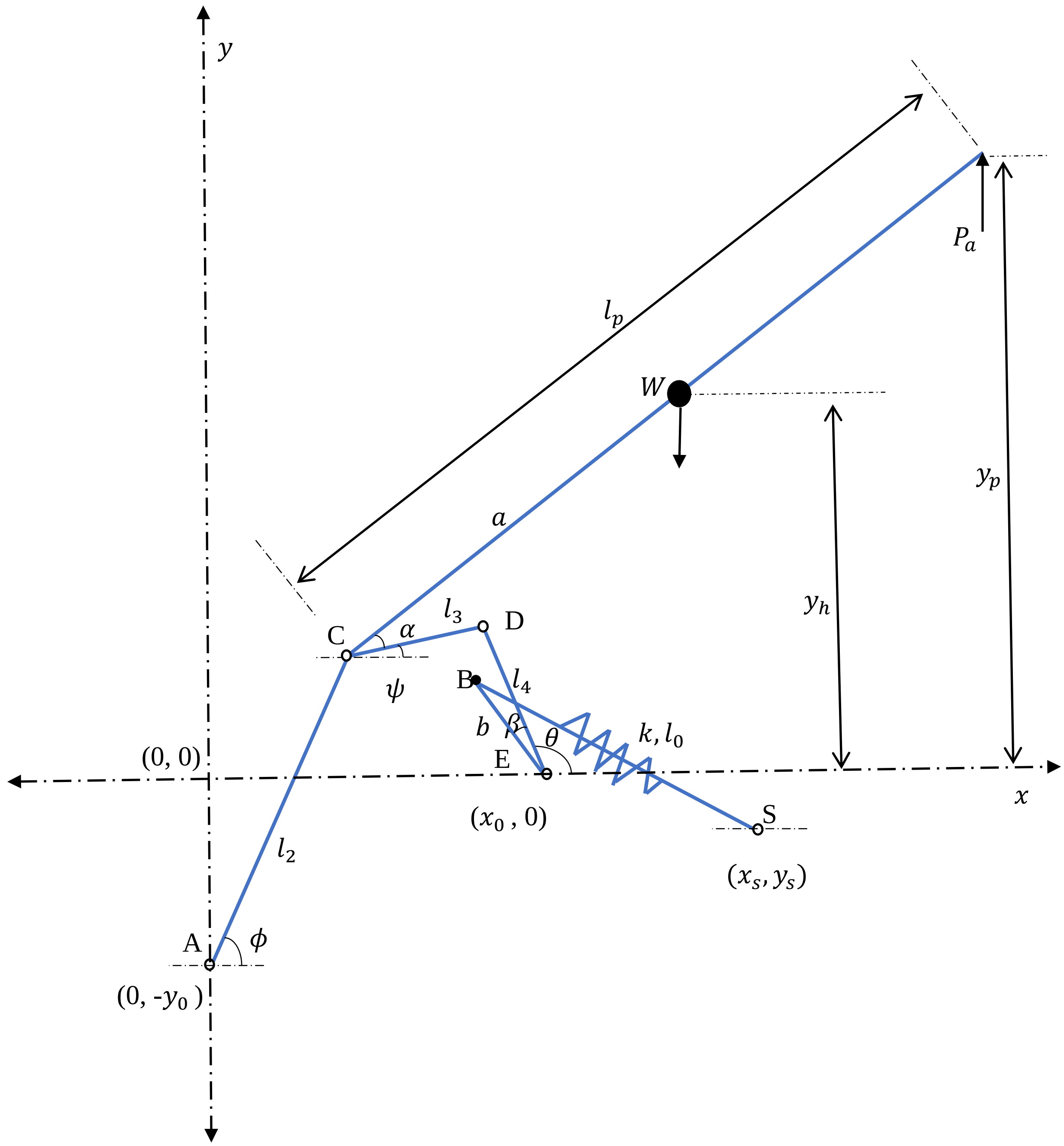

Figure 2-4 Extension spring second link application

The system shown in Figure 2.4 consists of an extension spring that is attached to second link [DE] of linkage. During the operation of this hood linkage, the extension spring stores potential energy that is given up when the closing of the hood, and is released back when the hood is opening. In this configuration, the first link [AC] is left without any attachment to any other mechanical assistance.

During optimization process k, β, l_0, b, x_s and y_s are specified as design variables. These are shown Figure 2.4 for an extension spring. k is the constant of extension spring. β is the angle between links [BE] and [DE]. l_0 is length of the extension spring, b is length of [BE] and x_s and y_s are starting coordinate positions of the extension spring attachment points.

2.3.1. Formulization of Extension Spring for Second Link Application

All of variables are shown in Figure 2.4. In addition, F_ψ and F_θ are velocity coefficients of system.

Loop equations:

$$l_2 e^{i\phi} + l_3 e^{i\psi} + \left(-l_4 e^{i\theta}\right) - x_0 - iy_0 = 0 \tag{2.24}$$

Reel Part:

$$l_2 \cos\phi + l_3 \cos\psi - l_4 \cos\theta - x_0 = 0 \tag{2.25}$$

Imaginary Part:

$$l_2 \sin\phi + l_3 \sin\psi - l_4 \sin\theta - y_0 = 0 \tag{2.26}$$

Velocity Coefficients:

$$-l_2 \sin\phi\, \delta\phi - l_3 \sin\psi F_\psi\, \delta\phi + l_4 \sin\theta\, F_\theta \delta\phi = 0 \tag{2.27}$$

$$l_2 \cos\phi\, \delta\phi + l_3 \cos\psi\, F_\psi \delta\phi - l_4 \cos\theta\, F_\theta \delta\phi = 0 \tag{2.28}$$

Length of spring: l_s

$$l_s^2 = (b_x - x_s)^2 + (b_y - y_s)^2 \tag{2.29}$$

$$l_s = \sqrt{(b_x - x_s)^2 + (b_y - y_s)^2} \tag{2.30}$$

$$b_x = b\cos(\theta + \beta) \tag{2.31}$$

$$b_y = b\sin(\theta + \beta) \tag{2.32}$$

Y position of c. g. of hood: y_h

$$y_h = -y_0 + l_2 \sin\phi + a\sin(\psi + \alpha) \tag{2.33}$$

Y position of applied force: $\boldsymbol{y_p}$

$$y_p = -y_0 + l_2 \sin\phi + l_p \sin(\psi + \alpha) \tag{2.34}$$

Virtual displacement:

d. δl_s:

$$2l_s\,\delta l_s = 2(b_x - x_s)(-b\sin(\theta+\beta)\delta\theta) + 2(b_y - y_s)(b\cos(\theta+\beta)\delta\theta) \tag{2.35}$$

$$\delta l_s = \left[\frac{-b(b_x - x_s)\sin(\theta+\beta) + b(b_y - y_s)\cos(\theta+\beta)}{l_s}\right]\delta\theta \tag{2.36}$$

e. δy_h:

$$\delta y_h = l_2\cos\phi\,\delta\phi + a\cos(\psi+\alpha)\left(\frac{\delta\psi}{\delta\phi}\right)\delta\phi \tag{2.37}$$

f. δy_p:

$$\delta y_p = l_2\cos\phi\,\delta\phi + l_p\cos(\psi+\alpha)\left(\frac{\delta\psi}{\delta\phi}\right)\delta\phi \tag{2.38}$$

***Conservative potential*:**

c. Hood*:*

$$V_h = W y_h \tag{2.39}$$

$$\delta V_h = W \delta y_h \tag{2.40}$$

d. Spring:

$$V_s = \frac{1}{2} k [l_s - l_0]^2 \tag{2.41}$$

$$\delta V_s = k [l_s - l_0] \delta l_s \tag{2.42}$$

Virtual work and force:

$$\delta W = P_a \delta y_p - \delta V_s - \delta V_h \tag{2.43}$$

$$= P_a \delta y_p - k(l_s - l_0) \delta l_s - W \delta y_h \tag{2.44}$$

$$\Big[P_a\{l_2 \cos\phi + l_p \cos(\psi + \alpha) F_\psi\} - W\{l_2 \cos\phi + a \cos(\psi + \alpha) F_\psi\} - k(l_s - l_0)\frac{b}{l_s}\{(b_x - x_s)\sin(\theta + \beta) + (b_y - y_s)\cos(\theta + \beta\}\Big] = 0 \tag{2.45}$$

$$P_a = \frac{W\{l_2 \cos\phi + a \cos(\psi + \alpha) F_\psi\} + k(l_s - l_0)\frac{b}{l_s}\{(b_x - x_s)\sin(\theta + \beta) + (b_y - y_s)\cos(\theta + \beta)\}}{l_2 \cos\phi + l_p \cos(\psi + \alpha) F_\psi} \tag{2.46}$$

2.3.2. Results

Firstly, k was taken as $0\ N/m$ in Equation (2.46). It means that there are no springs in system that is shown in Figure 2.5. The goal of this is to determine the differences in the force application to lift the hood between a case with extension spring on second link [DE] and are without.

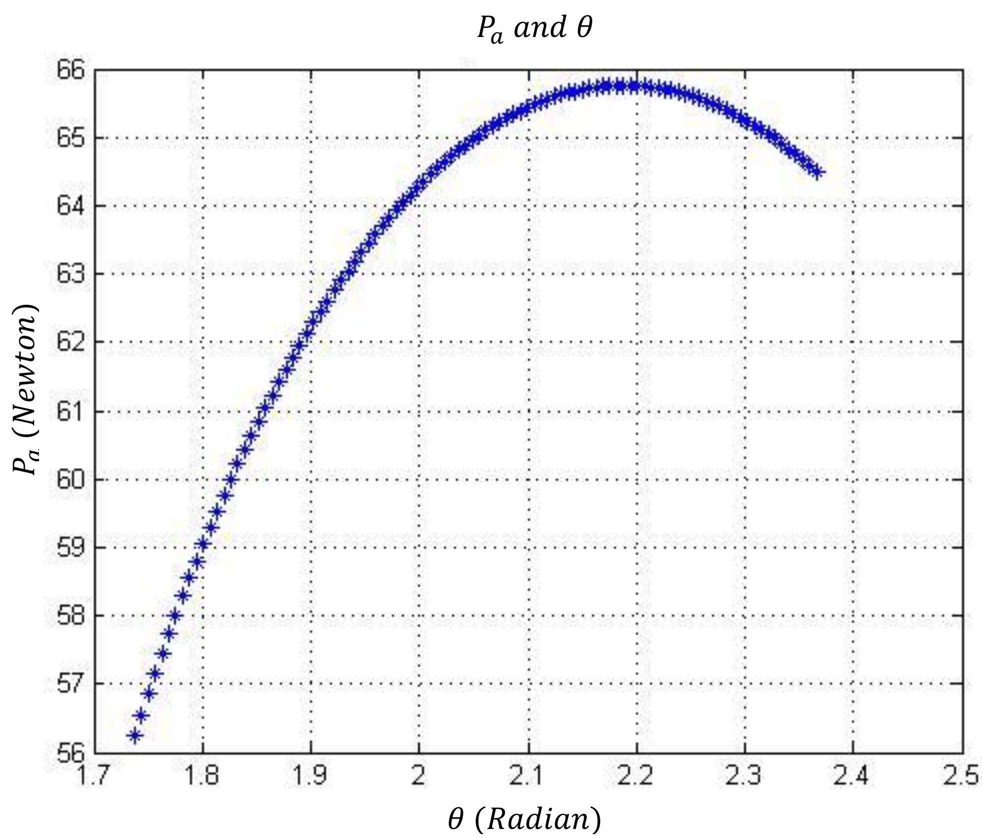

Figure 2-5 Force with θ and without any spring on the second link

After the optimization process, all design variables are shown in below in Table 2.2.

Table 2-2 Design variables results after optimization for second link

$k\ (N/m)$	$\beta\ (radian)$	$l_0\ (m)$	$b\ (m)$	$x_s\ (m)$	$y_s\ (m)$
100	-0.175	0.05	0.20	0.283	0

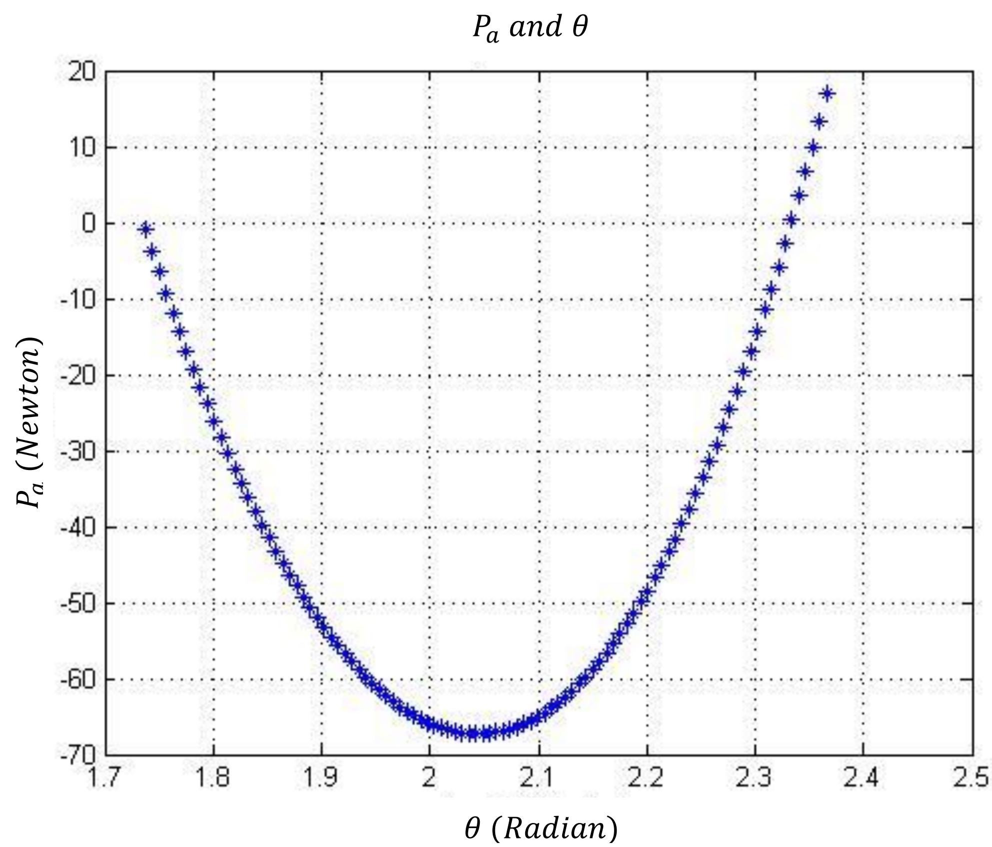

Figure 2-6 Force with θ after optimization for the second link

All the results are discussed together in Chapter 5.

CHAPTER 3: COMPRESSION SPRING APPLICATION

3.1 Relative Background Information

In this chapter, helical compression springs are used to store potential energy in the hood closing. Compression spring is one of the most common spring configuration used, and is found in many applications such as automotive and aerospace fields. A compression spring is a helical spring that works in an opposite ween from a helical extension spring.

Most of patents and articles focuses on tension and material of compression springs today. One example of a prior compression spring work is shown in U.S. Pat. No. 3,892,398 to Marsh, et al. [11]. In this patent compression springs are used in vehicle suspension systems. Other prior articles; Genova [12], Matthew and Tesar, [13] and Hain [14] relate to specific problems of mechanism-spring combinations which include helical compression springs.

3.2. First Link Application

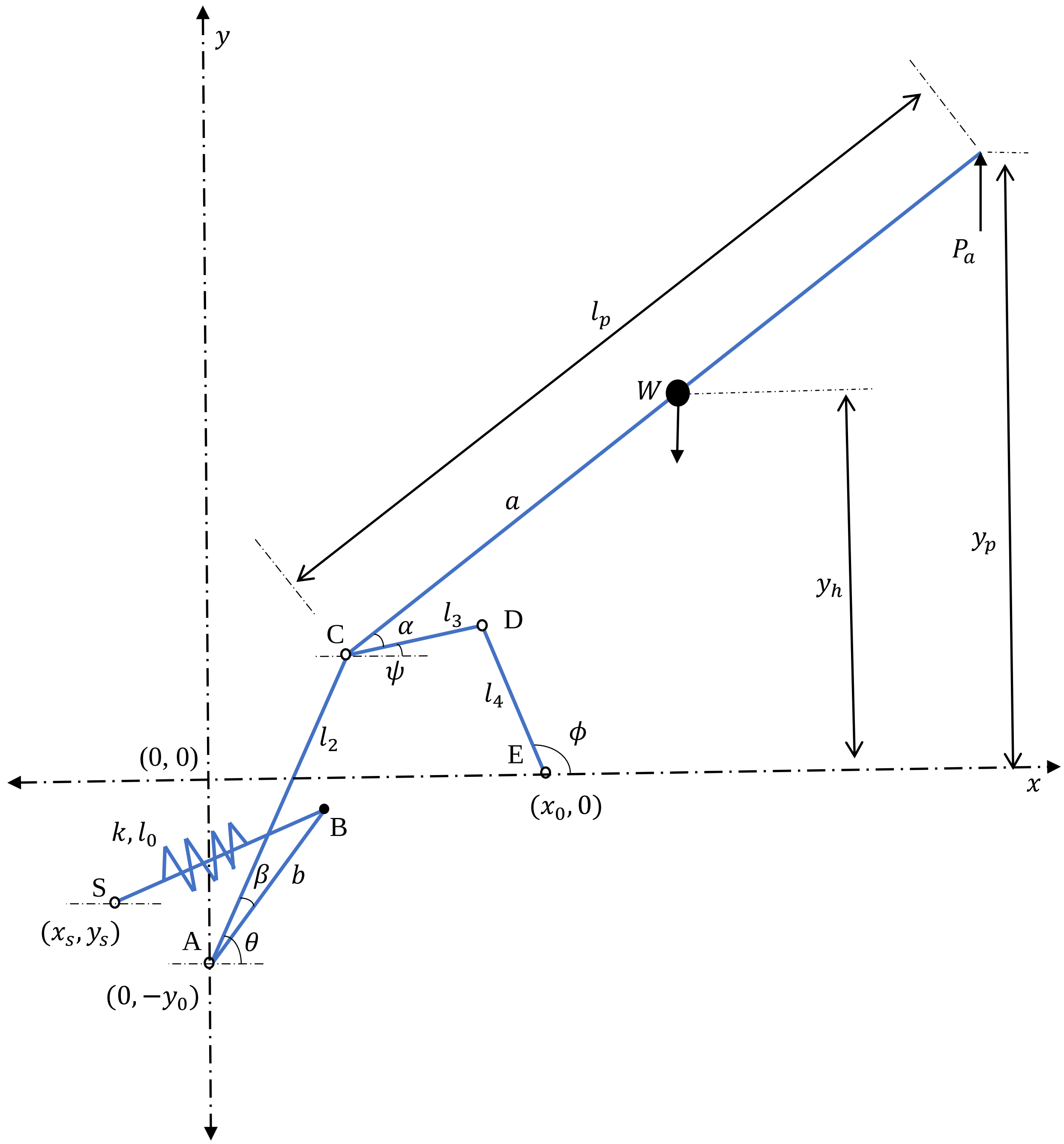

Figure 3-1 Compression spring first link application

The linkage system is shown in Figure 3.1. The compression spring is attached to the first link [AC] of linkage. In the hood application, the compression spring stores potential energy when the hood is closing and releases it back when the hood is opening. In this configuration, the second link [DE] is left without any attachment to any other mechanical assistance.

During optimization process k, β, l_0, b, x_s and y_s are specified as design variables. These are shown Figure 3.1 for a compression spring. k is the constant of compression spring. β is the angle between links [AC] and [AB]. l_0 is length of the compression spring, b is length of [AB] and x_s and y_s are starting coordinate positions of the compression spring attachment points.

3.2.1. Formulization of Compression Spring for First Link Application

All of variables are shown in Figure3.1. In addition, F_ψ and F_ϕ are velocity coefficients of system.

Loop equations:

$$l_2 e^{i\theta} + l_3 e^{i\psi} + \left(-l_4 e^{i\phi}\right) - x_0 - iy_0 = 0 \tag{3.1}$$

Reel Part:

$$l_2 \cos\theta + l_3 \cos\psi - l_4 \cos\phi - x_0 = 0 \tag{3.2}$$

Imaginary Part*:*

$$l_2 \sin\theta + l_3 \sin\psi - l_4 \sin\phi - y_0 = 0 \tag{3.3}$$

Velocity coefficients:

$$-l_2 \sin\theta\, \delta\theta - l_3 \sin\psi F_\psi\, \delta\theta + l_4 \sin\phi\, F_\phi \delta\theta = 0 \tag{3.4}$$

$$l_2 \cos\theta\, \delta\theta + l_3 \cos\psi\, F_\psi \delta\theta - l_4 \cos\phi\, F_\phi \delta\theta = 0 \tag{3.5}$$

Length of spring: $\boldsymbol{l_s}$

$$l_s^2 = (b_x - x_s)^2 + (b_y - y_s)^2 \tag{3.6}$$

$$l_s = \sqrt{(b_x - x_s)^2 + (b_y - y_s)^2} \tag{3.7}$$

$$b_x = b\cos(\theta + \beta) \tag{3.8}$$

$$b_y = b\sin(\theta + \beta) \tag{3.9}$$

Y position of c. g. of hood: $\boldsymbol{y_h}$

$$y_h = -y_0 + l_2 \sin\theta + a\sin(\psi + \alpha) \tag{3.10}$$

Y position of applied force: $\boldsymbol{y_p}$

$$y_p = -y_0 + l_2 \sin\theta + l_p \sin(\psi + \alpha) \tag{3.11}$$

Virtual displacement

g. δl_s:

$$2l_s\,\delta l_s = 2(b_x - x_s)(-b\sin(\theta+\beta)\delta\theta) + 2(b_y - y_s)(b\cos(\theta+\beta)\delta\theta) \tag{3.12}$$

$$\delta l_s = \left[\frac{-b(b_x - x_s)\sin(\theta+\beta) + b(b_y - y_s)\cos(\theta+\beta)}{l_s}\right]\delta\theta \tag{3.13}$$

h. δy_h:

$$\delta y_h = l_2\cos\theta\,\delta\theta + a\cos(\psi+\alpha)\left(\frac{\delta\psi}{\delta\theta}\right)\delta\theta \tag{3.14}$$

i. δy_p:

$$\delta y_p = l_2\cos\theta\,\delta\theta + l_p\cos(\psi+\alpha)\left(\frac{\delta\psi}{\delta\theta}\right)\delta\theta \tag{3.15}$$

Conservative potential

e. Hood:

$$V_h = W y_h \tag{3.16}$$

$$\delta V_h = W\delta y_h \tag{3.17}$$

f. Spring*:*

$$V_s = -\frac{1}{2}k[l_s - l_0]^2 \tag{3.18}$$

$$\delta V_s = -k[l_s - l_0]\delta l_s \tag{3.19}$$

Virtual work and force

$$\delta W = P_a \delta y_p - \delta V_s - \delta V_h \tag{3.20}$$

$$= P_a \delta y_p - [-k(l_s - l_0)\delta l_s] - W\delta y_h \tag{3.21}$$

$$\Big[P_a\{l_2 \cos\theta + l_p \cos(\psi + \alpha)F_\psi\} - W\{l_2 \cos\theta + a\cos(\psi + \alpha)F_\psi\} + k(l_s - l_0)\frac{b}{l_s}\{(b_x - x_s)\sin(\theta + \beta) + (b_y - y_s)\cos(\theta + \beta\}\Big] = 0 \tag{3.22}$$

$$P_a = \frac{W\{l_2 \cos\theta + a\cos(\psi + \alpha)\,F_\psi\} - k(l_s - l_0)\frac{b}{l_s}\{(b_x - x_s)\sin(\theta + \beta) + (b_y - y_s)\cos(\theta + \beta)\}}{l_2 \cos\theta + l_p \cos(\psi + \alpha)F_\psi} \tag{3.23}$$

Lagrange multipliers and sensitivity

In compression spring, six design variables are defined that are $k, \beta,\ l_0, b,\ x_s$ and y_s. In optimization process, these design variables are defined as $k\ is\ x_1,\ l_0\ is\ x_2, b\ is\ x_3,$ $x_s\ is\ x_4,\ y_s\ is\ x_5\ and\ \beta\ is\ x_6$. Equation (3.23) is an objective function and design variables plug in Equation (3.23) in below:

$$f(x) = \frac{W\{l_2 \cos\theta + a\cos(\psi + \alpha)\,F_\psi\} - x_1(l_s - x_2)\frac{x_3}{l_s}\{(b_x - x_4)\sin(\theta + x_6) + (b_y - x_5)\cos(\theta + x_6)\}}{l_2 \cos\theta + l_p \cos(\psi + \alpha)F_\psi} \tag{3.24}$$

Except design variable x_5, all other design variables have boundary values;

$$x_1 = 650\ ,\ \ x_2 = 0.38\ ,\ \ x_3 = 0.1\ , x_4 = -0.1\ ,\ \ x_6 = -0.175 \tag{3.25}$$

Assumed, $f(x)$ refers objective function and $g_1(x) = k_1,\ g_2(x) - k_2 +\ \ \dots$ (3.26) represent subjective functions. Thus, main Lagrange multipliers formula is;

$$L(x) = f(x) + \lambda_1[g_1(x) - k_1] + \lambda_2[g_2(x) - k_2] + \cdots \tag{3.27}$$

In that problem $x_1, x_2, x_3, x_4, x_5$and x_6 are subjective functions. At the same time, sensitivity analysis was done for design variable x_5 because only x_5 value is not boundary value only among all design variables.

3.2.2. Results

The parameters, k, β, l_0, b, x_s and y_s were taken as an arbitrary values in Equation (3.23). This means that the optimization of those values begin with some initial values for compression spring application on first link [AC] in system which is shown in Figure 3.1. The goal of this that compare differences between results with a compression spring on first link [AC] with initial values, with the optimum results after optimization.

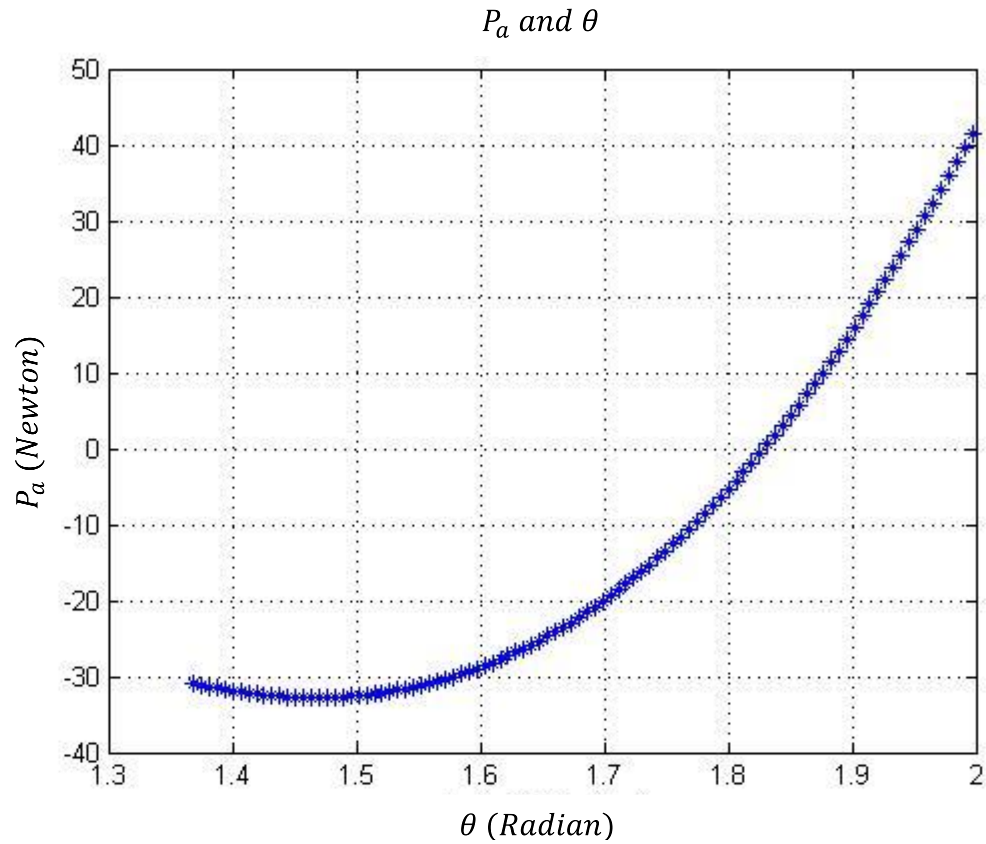

Figure 3-2 Force with θ with initial values compression spring on the first link

After Lagrange multipliers and sensitivity analysis x_5 is found -0.472 and λ is found as -0.372.

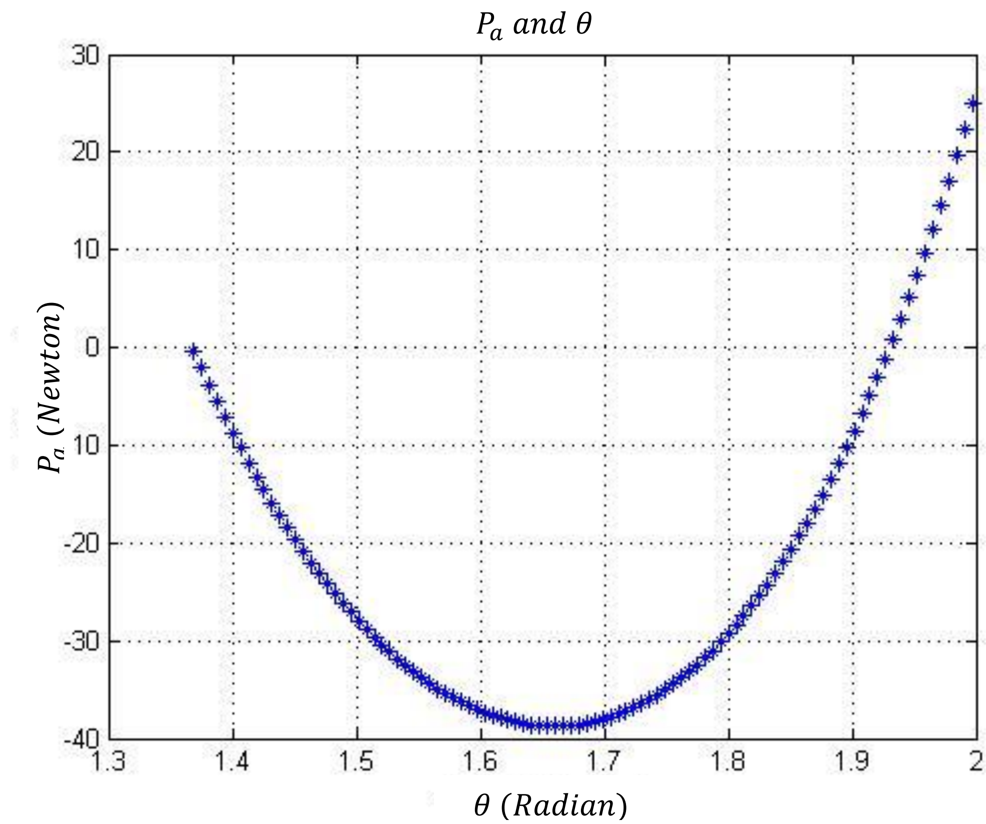

Figure 3-3 Force with θ after optimization compression spring on the first link

Before and after the optimization process, all design variables are shown in below in Table 2.2. Also, the results are discussed in Chapter 5.

Table 3-1 Design variables results before and after optimization for compression spring on the first link

	$k\ (N/m)$	$\beta\ (radian$	$l_0\ (m)$	$b\ (m)$	$x_s\ (m)$	$y_s\ (m)$
Before Optimization	100	0	0.2	0.1821	-0.1	-0.2
After Optimization	650	-0.175	0.38	0.1	-0.1	-0.384

3.3. Second Link Application

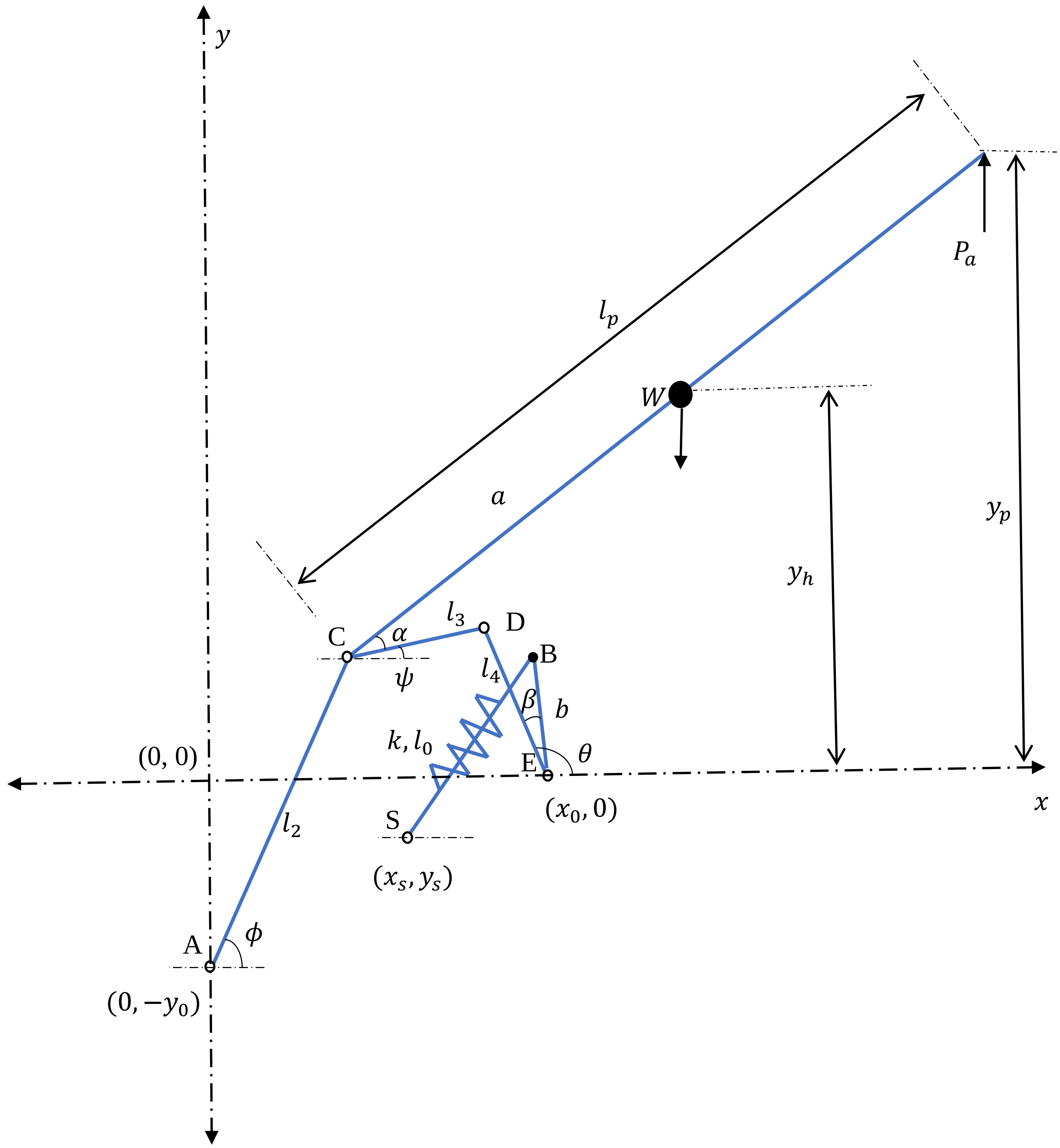

Figure 3-4 Compression spring second link application

The hood system is shown in Figure 3.4. A compression spring is then applied to the second link [DE] of system. During operation of the hood, the compression spring stores potential energy, when the hood, is closed the spring release back the store energy when the hood is opened. In this second link [DE] configuration the first link [AC] does not have any springs attached to it.

During optimization process k, β, l_0, b, x_s , and y_s are taken as design variables. These are shown Figure 3.4 for a compression spring. k is the constant of compression spring. β is the angle between links [DE] and [BE]. l_0 is length of the compression spring, b is length of [BE] and x_s and y_s are starting coordinate positions of the compression spring attachment points.

3.3.1. Formulization of Compression Spring for Second Link Application

All of variables are shown in Figure3.4. In addition, F_ψ and F_ϕ are velocity coefficients of system.

Loop equations:

$$l_2 e^{i\phi} + l_3 e^{i\psi} + \left(-l_4 e^{i\theta}\right) - x_0 - iy_0 = 0 \tag{3.28}$$

Reel Part:

$$l_2 \cos\phi + l_3 \cos\psi - l_4 \cos\theta - x_0 = 0 \tag{3.29}$$

Imaginary Part:

$$l_2 \sin\phi + l_3 \sin\psi - l_4 \sin\theta - y_0 = 0 \tag{3.30}$$

Velocity coefficients:

$$-l_2 \sin\phi\, \delta\phi - l_3 \sin\psi F_\psi\, \delta\phi + l_4 \sin\theta\, F_\theta \delta\phi = 0 \tag{3.31}$$

$$l_2 \cos\phi\, \delta\phi + l_3 \cos\psi\, F_\psi \delta\phi - l_4 \cos\theta\, F_\theta \delta\phi = 0 \tag{3.32}$$

Length of spring: $\boldsymbol{l_s}$

$$l_s^2 = (b_x - x_s)^2 + (b_y - y_s)^2 \tag{3.33}$$

$$l_s = \sqrt{(b_x - x_s)^2 + (b_y - y_s)^2} \tag{3.34}$$

$$b_x = b\cos(\theta + \beta) \tag{3.35}$$

$$b_y = b\sin(\theta + \beta) \tag{3.36}$$

Y position of c. g. of hood: $\boldsymbol{y_h}$

$$y_h = -y_0 + l_2 \sin\phi + a\sin(\psi + \alpha) \tag{3.37}$$

Y position of applied force: $\boldsymbol{y_p}$

$$y_p = -y_0 + l_2 \sin\phi + l_p \sin(\psi + \alpha) \tag{3.38}$$

Virtual displacement:

j. δl_s:

$$2l_s\,\delta l_s = 2(b_x - x_s)(-b\sin(\theta+\beta)\delta\theta) + 2(b_y - y_s)(b\cos(\theta+\beta)\delta\theta) \tag{3.39}$$

$$\delta l_s = \left[\frac{-b(b_x - x_s)\sin(\theta+\beta) + b(b_y - y_s)\cos(\theta+\beta)}{l_s}\right]\delta\theta \tag{3.40}$$

k. δy_h:

$$\delta y_h = l_2\cos\phi\,\delta\phi + a\cos(\psi+\alpha)\left(\frac{\delta\psi}{\delta\phi}\right)\delta\phi \tag{3.41}$$

l. δy_p:

$$\delta y_p = l_2\cos\phi\,\delta\phi + l_p\cos(\psi+\alpha)\left(\frac{\delta\psi}{\delta\phi}\right)\delta\phi \tag{3.42}$$

Conservative potential:

g. Hood:

$$V_h = W y_h \tag{3.43}$$

$$\delta V_h = W\delta y_h \tag{3.44}$$

h. Spring:

$$V_s = -\frac{1}{2}k[l_s - l_0]^2 \tag{3.45}$$

$$\delta V_s = -k[l_s - l_0]\delta l_s \tag{3.46}$$

Virtual work and force:

$$\delta W = P_a \delta y_p - \delta V_s - \delta V_h \tag{3.47}$$

$$= P_a \delta y_p - [-k(l_s - l_0)\delta l_s] - W \delta y_h \tag{3.48}$$

$$\Big[P_a\{l_2 \cos\phi + l_p \cos(\psi + \alpha)F_\psi\} - W\{l_2 \cos\phi + a\cos(\psi + \alpha)F_\psi\} + k(l_s - l_0)\frac{b}{l_s}\{(b_x - x_s)\sin(\theta + \beta) + (b_y - y_s)\cos(\theta + \beta\}\Big] = 0 \tag{3.49}$$

$$P_a = \frac{W\{l_2 \cos\phi + a\cos(\psi + \alpha)\, F_\psi\} - k(l_s - l_0)\frac{b}{l_s}\{(b_x - x_s)\sin(\theta + \beta) + (b_y - y_s)\cos(\theta + \beta)\}}{l_2 \cos\phi + l_p \cos(\psi + \alpha)F_\psi} \tag{3.50}$$

3.3.2. Results

The parameters, k, β, l_0, b, x_s and y_s were taken as an arbitrary values in Equation (3.50). This means that the optimization of those values begin with some initial values for compression spring application on second link [DE] in system which is shown in Figure 3.4. The goal of this that compare differences between results with a compression spring on second link [DE] with initial values, with the optimum results after optimization.

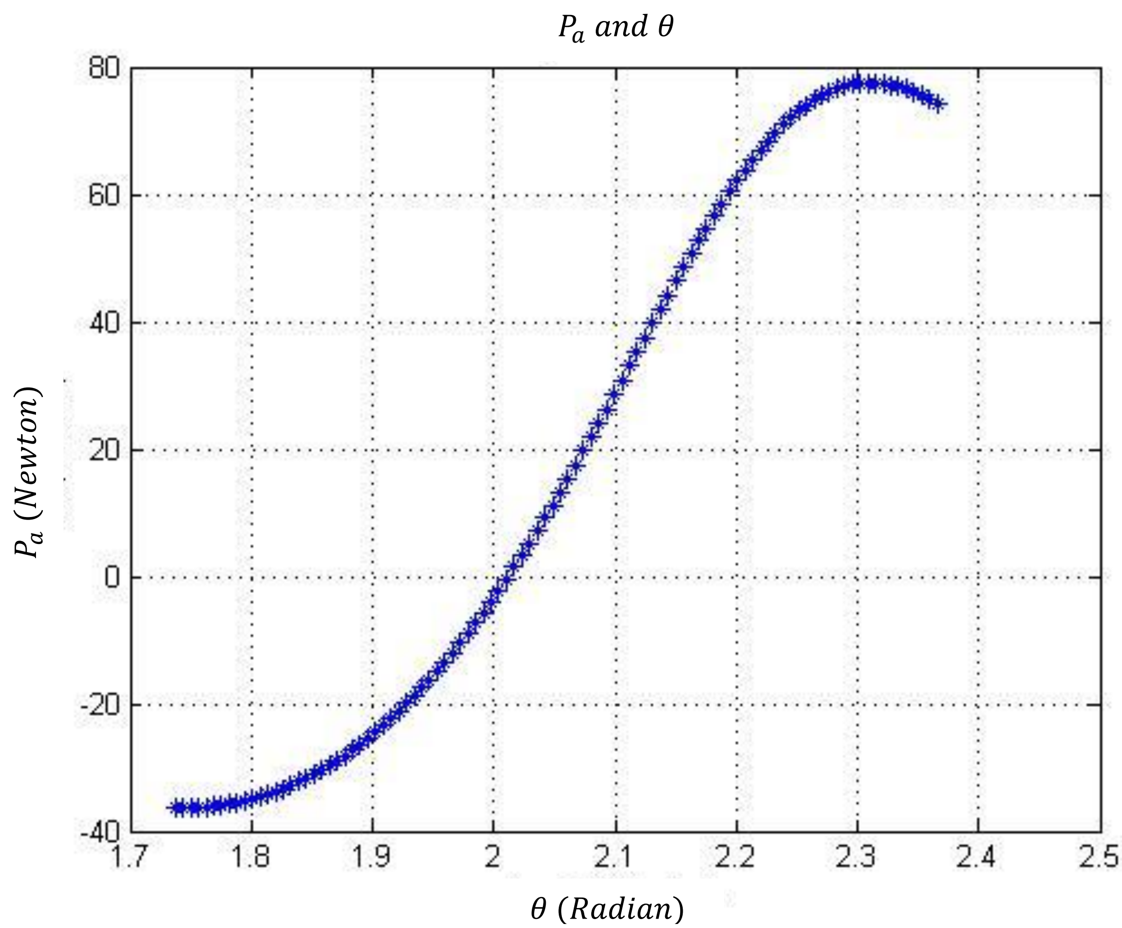

Figure 3-5 Force with θ with initial values compression spring on the second link

Before and after the optimization process, all design variables are shown in below in Table 3.2. Also, the results are discussed in Chapter 5.

Table 3-2 Design variables results before and after optimization for compression

	$k\ (N/m)$	$\beta\ (radian)$	$l_0\ (m)$	$b\ (m)$	$x_s\ (m)$	$y_s\ (m)$
Before Optimization	80	0	0.15	0.216	0.15	-0.20
After Optimization	40	0.274	0.3	0.25	-0.1	-0.15

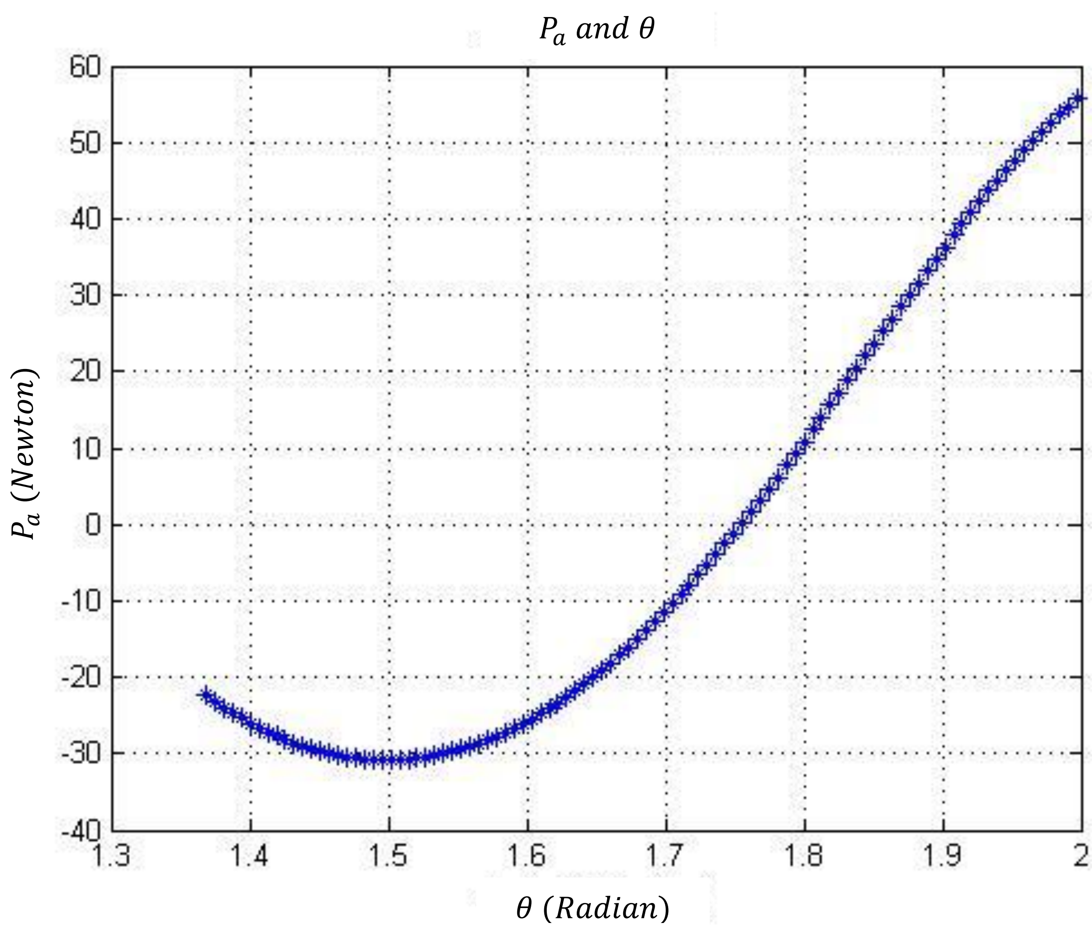

Figure 3-6 Force with θ after optimization compression spring

CHAPTER 4: TORSION SPRING APPLICATION

4.1. Relative Background Information

A torsion spring is a spring that works by torsion or twisting; that is, a flexible elastic object that stores mechanical energy when it is twisted. When it is twisted, it exerts a torque in the opposite direction and proportional to the angle it is twisted. [7]. The torsion spring configuration is created for the purpose of storing and releasing angular energy or for the purpose of statically holding a mechanism in place by deflecting the legs about the body centerline axis. A torsion spring is normally supported by a rod (mandrel) that is coincident with the theoretical hinge line of the final product. [8].

There are a number of torsion spring balance assembly patents and different applications. One example of a prior spring balance assembly is shown in U.S. Pat. No. 3,038,714 to Klaus, et al. [9]. The spring tension can therefore be adjusted (e.g., increased or decreased) by either counter-clockwise or clockwise rotational movement of a spring regulator. This design allows the coil spring to be either tightened or loosened to the degree necessary to sufficiently support or position a loading arm.

Another prior spring assembly unit, for example, as seen in U.S. Pat. No. 4,537,233 to Vroonland, et al. [10], had a protectively covered spring torsion unit with an assembly of a threaded adjustment screw and barrel nut. In use, the bolt theoretically could be rotated to adjust (e.g., increase or decrease) the spring tension, as the barrel nut moves laterally and alters its effective length along the bolt.

All of these applications show that helical extension springs are used in mechanical systems for static balancing or dynamic balancing and have a number of application fields.

4.2. First Link Application

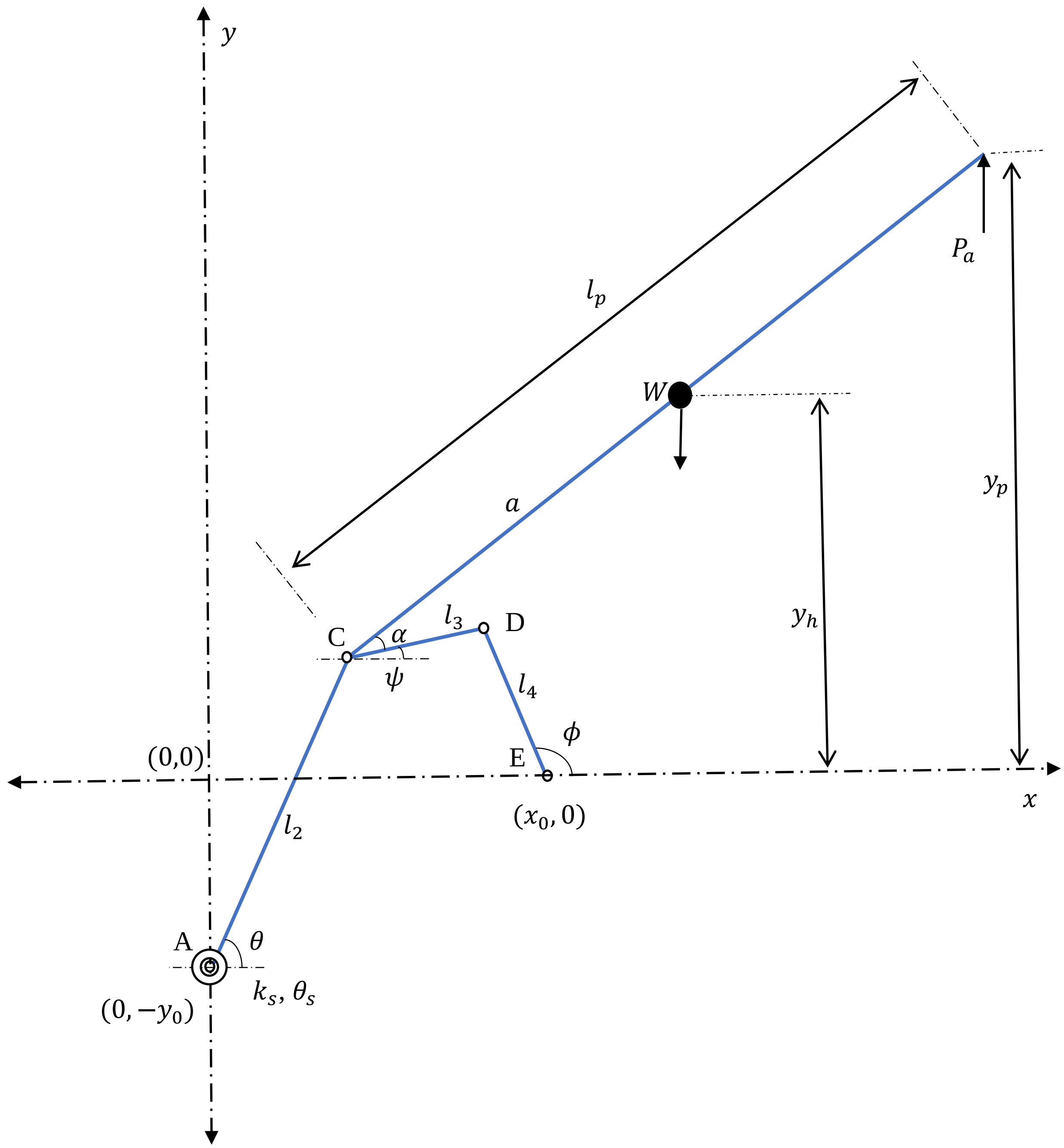

Figure 4-1 Torsion spring first link application

The linkage system is shown in Figure 4.1. Torsion spring is applied to the pivot point of the first link [AC] of system. During the torsion spring application, the torsion spring stores mechanical energy that is given up by the closing of the hood and releases it back when the hood is open. In this configuration, the second link [DE] is left without any attachment to any other mechanical assistance.

During optimization process the parameters k_s and θ_s were taken as design variables. k_s is a constant of torsion spring. θ_s is an angle which is a constant for torsion spring. Both of them are shown Figure 4.1.

4.2.1. Formulization of Torsion Spring for First Link Application

All of variables are shown in Figure 4.1. In addition, F_ψ and F_ϕ are velocity coefficients of system.

Loop equations:

$$l_2 e^{i\theta} + l_3 e^{i\psi} + \left(-l_4 e^{i\phi}\right) - x_0 - i y_0 = 0 \tag{4.1}$$

a. Reel Part:

$$l_2 \cos\theta + l_3 \cos\psi - l_4 \cos\phi - x_0 = 0 \tag{4.2}$$

b. Imaginary Part:

$$l_2 \sin\theta + l_3 \sin\psi - l_4 \sin\phi - y_0 = 0 \tag{4.3}$$

Velocity coefficients:

$$-l_2 \sin\theta\, \delta\theta - l_3 \sin\psi F_\psi\, \delta\theta + l_4 \sin\phi\, F_\phi \delta\theta = 0 \tag{4.4}$$

$$l_2 \cos\theta\, \delta\theta + l_3 \cos\psi\, F_\psi \delta\theta - l_4 \cos\phi\, F_\phi \delta\theta = 0 \tag{4.5}$$

Y position of hood: $\boldsymbol{y_h}$

$$y_h = -y_0 + l_2 \sin\theta + a \sin(\psi + \alpha) \tag{4.6}$$

Y position of applied force:$\boldsymbol{y_p}$

$$y_p = -y_0 + l_2 \sin\theta + l_p \sin(\psi + \alpha) \tag{4.7}$$

Virtual displacement

m. δy_h:

$$\delta y_h = l_2 \cos\theta\, \delta\theta + a \cos(\psi + \alpha) \left(\frac{\delta\psi}{\delta\theta}\right) \delta\theta \tag{4.8}$$

n. δy_p:

$$\delta y_p = l_2 \cos\theta\, \delta\theta + l_p \cos(\psi + \alpha) \left(\frac{\delta\psi}{\delta\theta}\right) \delta\theta \tag{4.9}$$

Conservative potential

a. Hood:

$$V_h = W y_h \tag{4.10}$$

$$\delta V_h = W \delta y_h \tag{4.11}$$

b. Spring:

$$V_s = \frac{1}{2} k_s[\theta - (\theta_o - \theta_s)]^2 \tag{4.12}$$

$$\delta V_s = k_s[\theta - (\theta_o - \theta_s)] \tag{4.13}$$

Virtual work

$$\delta W = P_a \delta y_p - \delta V_s - \delta V_h \tag{4.14}$$

$$= P_a \delta y_p - k_s[\theta - (\theta_o - \theta_s)] - W \delta y_h \tag{4.15}$$

$$\left[P_a\left\{l_2 \sin\theta + l_p \cos(\psi + \alpha) F_\psi\right\} - W\left\{l_2 \cos\theta + a \cos(\psi + \alpha) F_\psi\right\} - k_s[\theta - (\theta_o - \theta_s)]\right] = 0 \tag{4.16}$$

$$P_a = \frac{W\left\{l_2 \cos\theta + a \cos(\psi + \alpha) F_\psi\right\} + k_s[\theta - (\theta_o - \theta_s)]}{l_2 \sin\theta + l_p \cos(\psi + \alpha) F_\psi} \tag{4.17}$$

4.2.2. Results

The parameters, k_s and θ_s were taken as an arbitrary values in Equation (4.17). This means that the optimization of those values begin with some initial values for torsion spring application on first link [AC] in system which is shown in Figure 4.1. The goal of this that compare differences between results with a torsion spring on first link [AC] with initial values, with the optimum results after optimization.

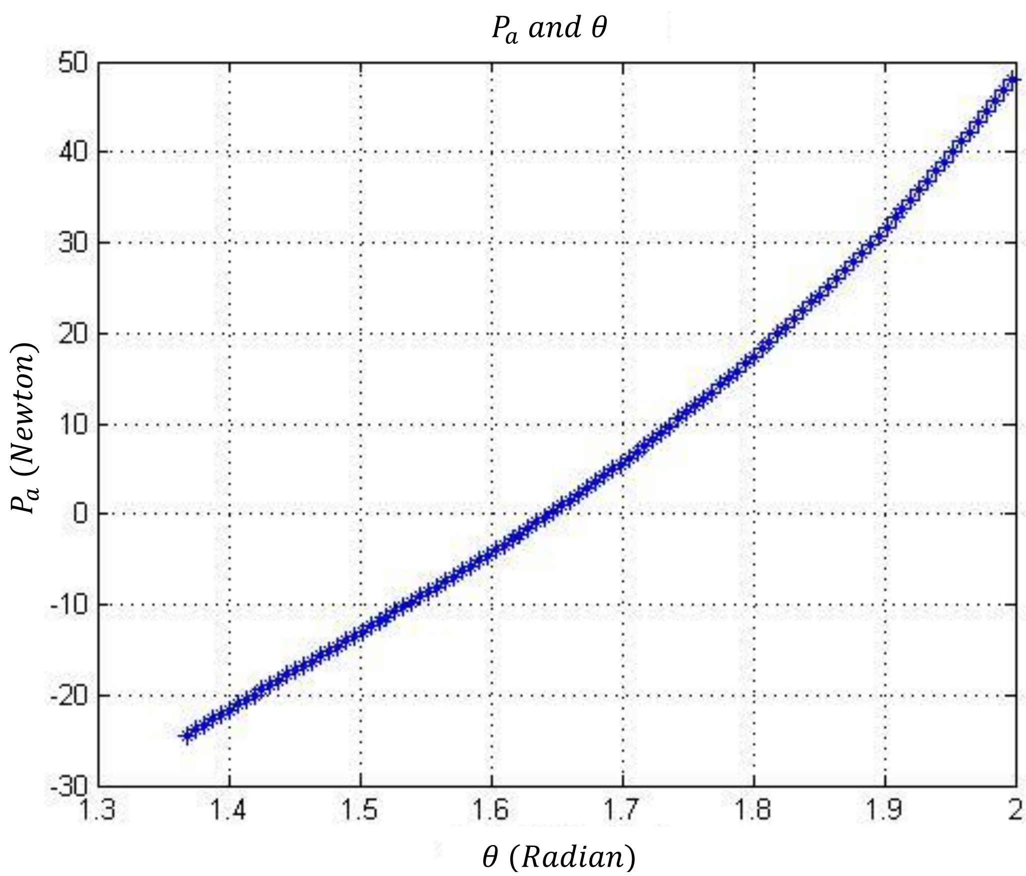

Figure 4-2 Force with θ with initial values torsion spring on the first link

As initial values, k_s was taken 100 N/m and θ_s was taken 45°. After optimization process; MATLAB is given below graph which is Figure 4.3.

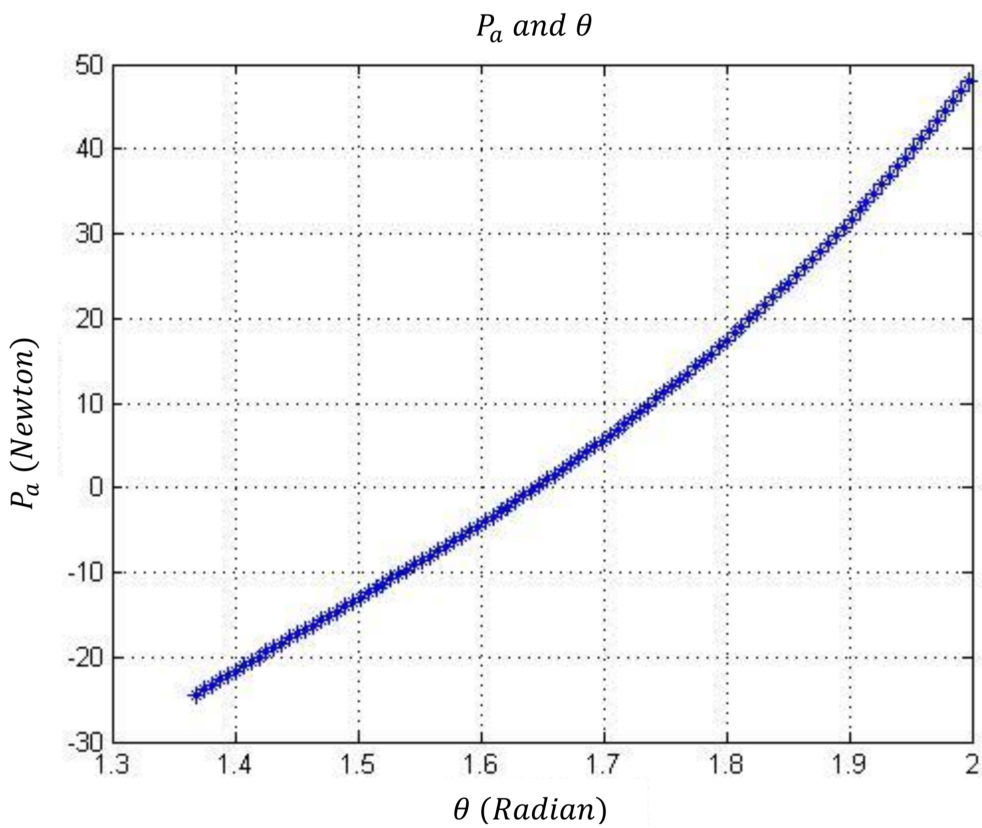

Figure 4-3 Force with θ after optimization torsion spring on the first link

After optimization process k_s is found 102.597 N/m and θ_s is found 30.093°. The results are discussed in Chapter 5. Both results are shown in Table 4.1 in below:

Table 4-1 Design variables results before and after optimization for torsion spring

	$k_s\ (N/m)$	$\theta_s(degree)$
Before Optimization	100	45
After Optimization	102.597	30.093

4.3. Second Link Application

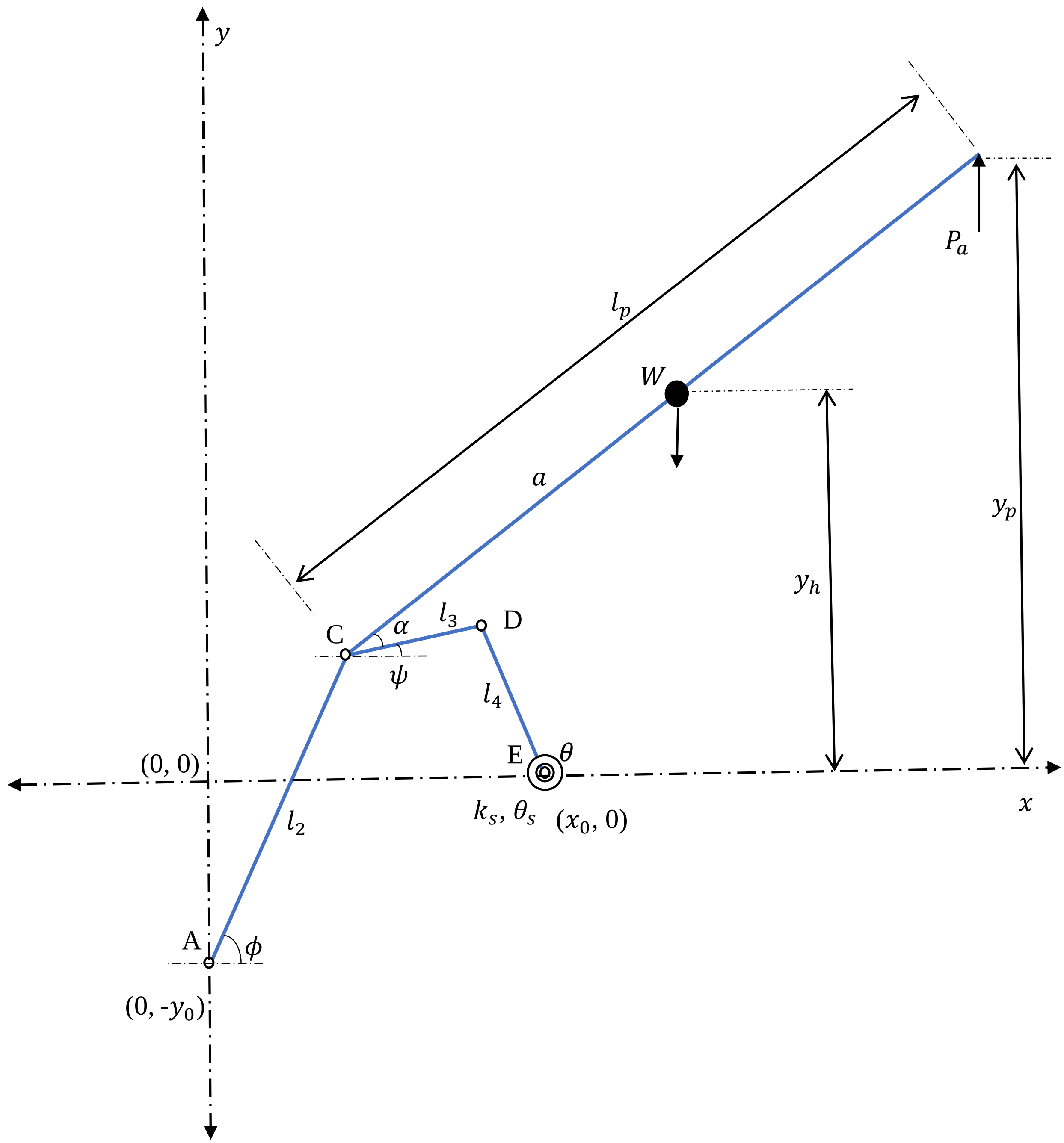

Figure 4-4 Torsion spring second link application

The linkage system is shown in Figure 4.4. Torsion spring is applied to the pivot point of the second link [DE] of system. During the torsion spring application, the torsion spring stores mechanical energy that is given up by the closing of the hood and releases it back when the hood is open. In this configuration, the first link [AC] is left without any attachment to any other mechanical assistance.

During optimization process the parameters k_s and θ_s were taken as design variables. k_s is a constant of torsion spring. θ_s is an angle which is a constant for torsion spring. Both of them are shown Figure 4.4.

4.3.1. Formulization of Torsion Spring for Second Link Application

All of variables are shown in Figure 4.4. In addition, F_ψ and F_ϕ are velocity coefficients of system.

Loop equations:

$$l_2 e^{i\phi} + l_3 e^{i\psi} + \left(-l_4 e^{i\theta}\right) - x_0 - i y_0 = 0 \tag{4.18}$$

Reel Part:

$$l_2 \cos\phi + l_3 \cos\psi - l_4 \cos\theta - x_0 = 0 \tag{4.19}$$

Imaginary Part:

$$l_2 \sin\phi + l_3 \sin\psi - l_4 \sin\theta - y_0 = 0 \tag{4.20}$$

Velocity coefficients:

$$-l_2 \sin\phi\, \delta\phi - l_3 \sin\psi F_\psi\, \delta\phi + l_4 \sin\theta\, F_\theta \delta\phi = 0 \tag{4.21}$$

$$l_2 \cos\phi\, \delta\phi + l_3 \cos\psi\, F_\psi \delta\phi - l_4 \cos\theta\, F_\theta \delta\phi = 0 \tag{4.22}$$

Y position of c. g. of hood: $\boldsymbol{y_h}$

$$y_h = -y_0 + l_2 \sin\phi + a \sin(\psi + \alpha) \tag{4.23}$$

Y position of applied force: $\boldsymbol{y_p}$

$$y_p = -y_0 + l_2 \sin\phi + l_p \sin(\psi + \alpha) \tag{4.24}$$

Virtual displacement

a. δy_h:

$$\delta y_h = l_2 \cos\phi\, \delta\phi + a \cos(\psi + \alpha) \left(\frac{\delta\psi}{\delta\phi}\right) \delta\phi \tag{4.25}$$

b. δy_p:

$$\delta y_p = l_2 \cos\phi\, \delta\phi + l_p \cos(\psi + \alpha) \left(\frac{\delta\psi}{\delta\phi}\right) \delta\phi \tag{4.26}$$

Conservative potential

c. Hood:

$$V_h = Wy_h \tag{4.27}$$

$$\delta V_h = W\delta y_h \tag{4.28}$$

d. Spring:

$$V_s = \frac{1}{2}k_s[\theta - (\theta_o - \theta_s)]^2 \tag{4.29}$$

$$\delta V_s = k_s[\theta - (\theta_o - \theta_s)] \tag{4.30}$$

Virtual work

$$\delta W = P_a\delta y_p - \delta V_s - \delta V_h \tag{4.31}$$

$$= P_a\delta y_p - k_s[\theta - (\theta_o - \theta_s)] - W\delta y_h \tag{4.32}$$

$$\begin{aligned}\Big[P_a\big\{l_2\cos\phi + l_p\cos(\psi+\alpha)F_\psi\big\} - W\big\{l_2\cos\phi + a\cos(\psi+\alpha)F_\psi\big\}\\ - k_s[\theta - (\theta_o - \theta_s)]\Big] = 0\end{aligned} \tag{4.33}$$

$$P_a = \frac{W\big\{l_2\cos\phi + a\cos(\psi+\alpha)\,F_\psi\big\} + k_s[\theta - (\theta_o - \theta_s)]}{l_2\cos\phi + l_p\cos(\psi+\alpha)F_\psi} \tag{4.34}$$

4.3.2. Results

The parameters, k_s and θ_s were taken as an arbitrary values in Equation (4.34). This means that the optimization of those values begin with some initial values for torsion spring application on second link [DE] in system which is shown in Figure 4.4. The goal of this that compare differences between results with a torsion spring on second link [DE] with initial values, with the optimum results after optimization.

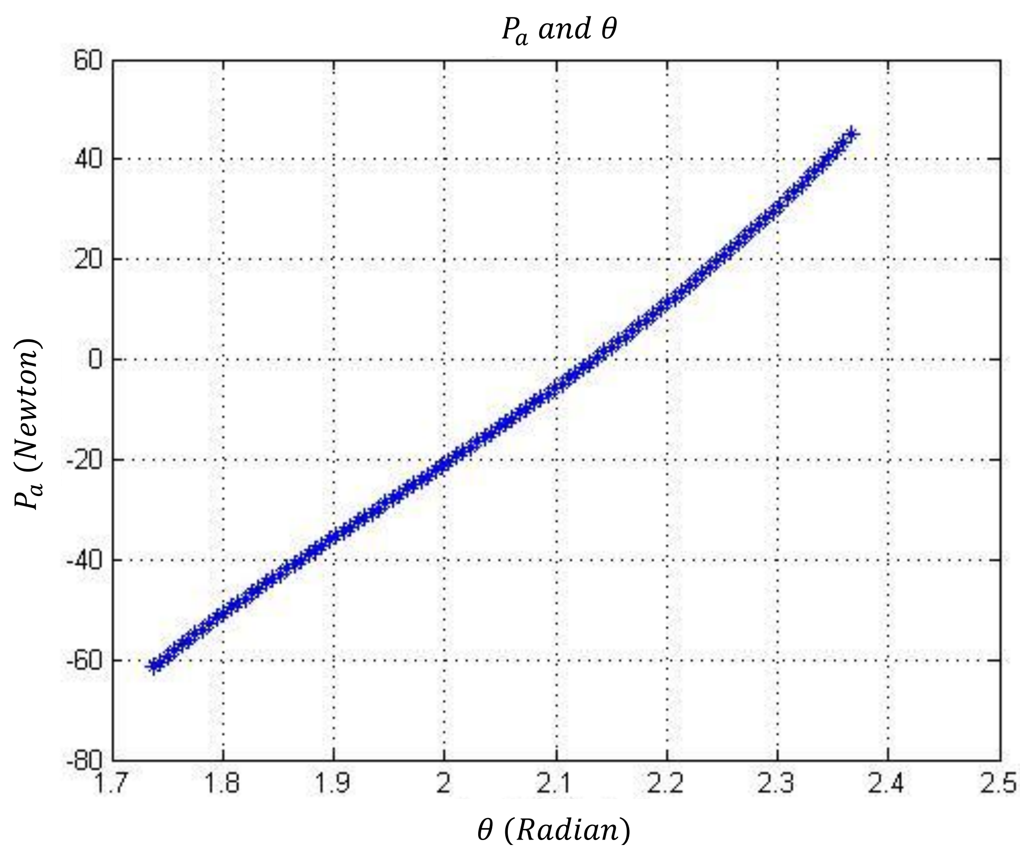

Figure 4-5 Force with θ with initial values torsion spring on the second link

As initial values, k_s is taken 100 N/m and θ_s is taken 45°. After optimization process; MATLAB is given below graph which is Figure 4.6.

After optimization process k_s is found 73.645 N/m and θ_s is found 44.98°. Both results are shown in Table 4.2 in below, Also, the results are discussed in Chapter 5.

Table 4-2 Design variables results before and after optimization for torsion spring

	k_s (N/m)	$\theta_s(degree)$
Before Optimization	100	45
After Optimization	73.645	44.98

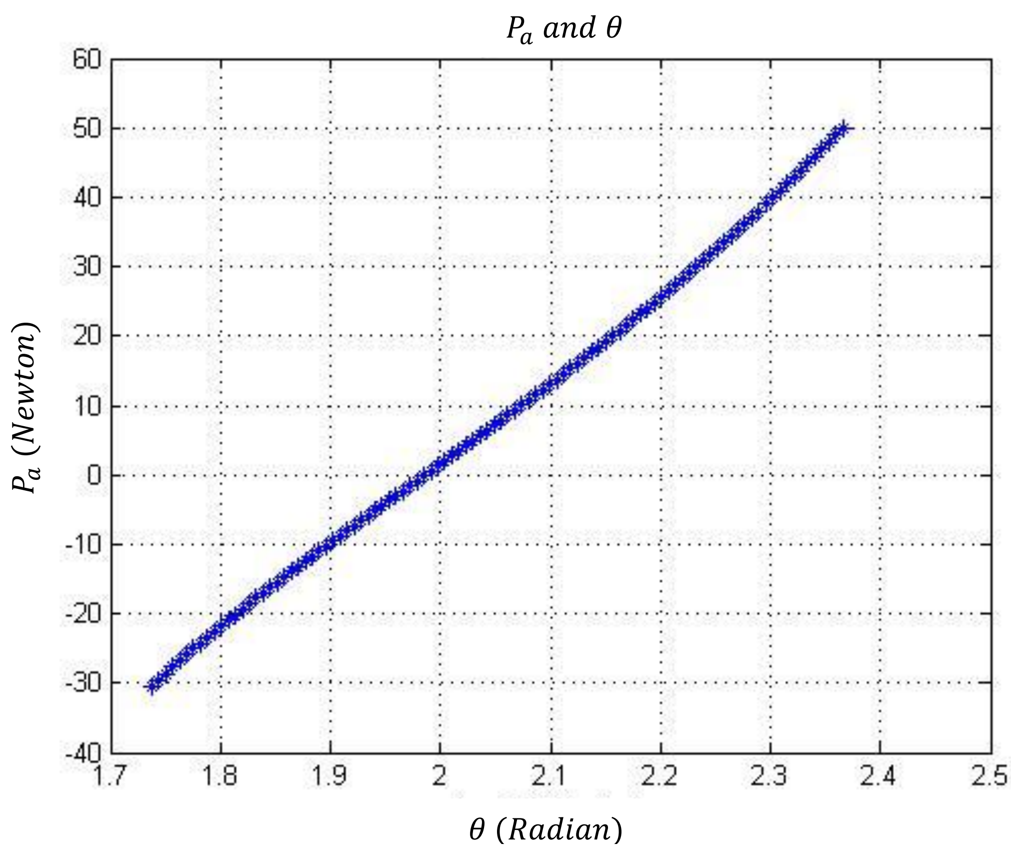

Figure 4-6 Force with θ after optimization torsion spring on the second link

CHAPTER 5: CONCLUSION AND POTENTIAL FUTURE WORK

5.1 Conclusion

5.1.1. First Link Applications

All figures are shown in Chapter 2, 3 and 4. Figure 2.3 and Figure 4.3 show that when engine hood is open, engine hood does not stay open position without the presence. In addition, Figure 2.3 and Figure 4.3 also show that to close the engine hood, a force is also needed. On the other hand, Figure 3.3 shows that the engine hood can stay open position without any force or prop when compression springs are used on first link because the figure shows that force which is needed to apply for open position is zero. It means that engine hood can stay in the open position all by itself. Moreover, the closing force required is also lower than for extension springs and torsion springs configurations. These results show that compression spring the best applied on first link, and is the best of all the different configurations.

Finally, after Lagrange multipliers and sensitivity analysis, y_s (design variable x_5) position was found as $-0.472\,m$ instead of $-0.484\,m$. There is only 0.012 m between after and before sensitivity analysis. Thus, with Lagrange multipliers and sensitivity analysis, optimal values were found for all of design variables.

The design variables are shown in Table 5.1 below but a decision cannot be made just by examining these values because all these values affect opening force on the hood.

Table 5-1 Compression spring, extension spring and torsion spring design variables after optimization for first link

	$k\ (N/m)$	$\beta\ (radian)$	$l_0\ (m)$	$b\ (m)$	$x_s\ (m)$	$y_s\ (m)$
Extension Spring	666.93	-0.018	0.05	0.2	0.4	0
Compression Spring	650	-0.175	0.38	0.1	-0.1	-0.484
	$k_s\ (N/m)$	$\theta_s\ (degree)$				
Torsion Spring	102.597	30.093				

5.1.2 Second Link Applications

All figures are shown in Chapter 2, 3 and 4. Figure 3.6 and Figure 4.6 show that when engine hood is open, engine hood does not stay open position without the presence. In addition, Figure 3.6 and Figure 4.6 also show that to close the engine hood, a force is also needed. On the other hand, Figure 2.6 shows that the engine hood can stay open position without any force or prop when extension springs are used on second link because the figure shows that force which is needed to apply for open position is zero. It means that engine hood can stay in the open position all by itself. Moreover, the closing force required is also lower than for compression springs and torsion springs configurations. These results show that extension spring the best applied on second link, and is the best of all the different configurations. Furthermore, Lagrange multipliers and sensitivity analysis isn't done for extension spring because all design variables have boundary values for second link.

The design variables are shown in Table 5.2 below but a decision cannot be made just by examining these values because all these values affect opening force on the hood.

Table 5-2 Compression spring, extension spring and torsion spring design variables after optimization for second link

	$k\ (N/m)$	$\beta\ (radian)$	$l_0\ (m)$	$b\ (m)$	$x_s\ (m)$	$y_s\ (m)$
Extension Spring	100	-0.175	0.05	0.2	0.283	0
Compression Spring	40	0.274	0.3	0.25	-0.1	-0.15
	$k_s\ (N/m)$	$\theta_s\ (degree)$				
Torsion Spring	73.645	44.98				

5.2. Potential Future Work

For the future work, it is anticipated the same as a different kind of spring (extension, compression or torsion) can be applied on first link and second link at the same time. It means that one kind of spring can be applied on first link, while the same as different kind of spring can be applied on second link of the four-bar linkage. Thus, both of springs can work together when engine hood is opened or closed and with that kind of application, it is anticipated the engine hood can be opened or closed more readily.

To illustrate, while a torsion spring can be applied on first link, a compression spring can be applied on second link at the same time. That kind of application can be more helpful for opening or closing the hood.

CHAPTER 6: REFERENCES

[1]. Deb, Kalyanmoy and Mayank Goyal. "A Flexible Optimization Procedure for Mechanical Component Design Based on Genetic Adaptive Search." *Journal of Mechanical Design 120(2)* (1998): 162-164.

[2]. Kannan, B. K. and S. N. Kramer. "An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design." *Journal of Mechanical Design 116(2)* (June 1994): 405-411.

[3]. Sandgren, E. "Nonlinear Integer and Discrete Programming in Mechanical Design Optimization." *Journal of Mechanical Design 112* (June 1990): 223-229.

[4]. M., Parades, M. Sartor and C. Masclet. "An Optimization Process for Extension Spring Design." *Computer Methods in Applied Mechanics and Engineering (Volume 191, Issues 8-10)* (21 December 2001): 783-797.

[5]. Nathan, R. H. "A Constant Force Generation Mechanism." *Journal of Mechanisms, Transmissions, and Automation in Design (107)* (December 1985): 508-512.

[6]. Company, Lee Spring Limited. *Designing & Specifying Compression, Extension and Torsion Springs*. Guide. Wokingham: Lee Spring Limited, 2002.

[7]. Shigley, Joseph E., Charles R. Mischke and Richard G. Budynas. *Mechanical Engineering Design*. New York: McGraw Hill,, 2003. 542.

[8]. Company, Century Springs. *Torsion Springs*. Engineers Guide. Bristol: Century Springs, 2003.

[9]. Schwieterman, Mark A. and Dale E. Hutchinson. Torsion Spring Balance Assembly and Adjustment Method. United States: Patent 5,983,936. 16 November 1996.

[10]. Vroonland, Evart J. and M. Curtis. Spring Balance Assembly. United States: Patent 4,537,233. 27 August 1985.

[11]. Marsh, Gerald Leonard. Compression Spring. United States: Patent 3,892,398. 1 July 1975.

[12]. Genova, P. E. *Mechanics of Plane Mechanisms with Spring Links*. VI Lenin, Sofia, Bulgaria: Mechanical and Electrofechnical Institute, 1972.

[13]. Matthew, G. K. and Tezar, D. "Synthesis of Spring Parameters to Balance General Forcing Functions in Planar Mechanisms." *Journal of Engineering for Industry* (May 1977): 347-352.

[14]. Hain, K. "Spring Mechanisms-Point Balancing." Chironis, Nicholas Paul. *Spring Design and Application*. New York: McGraw-Hill, 1961. 268-275.

[15]. Norton, Robert L. *Design of Machinery* . McGraw-Hill, 1999.

[16]. MathWorks. *Optimization Toolbox™ User's Guide*. Natick, MA: The MathWorks, Inc., 2014.

[17]. P.Venkataraman. *Applied Optimization with MATLAB Programming*. New Jersey: Wiley, 2009.

[18]. Gockenbach, Mark S. *Lagrance Multipliers and Sensitivity*. Boston, 2013. Class Notes, MIT.

CHAPTER 7: APPENDICES

Appendix A: Automotive Engine Hood Three-Position Four-Bar Linkage Mechanism Synthesis MATLAB Codes

```
%ENGINE HOOD FOUR-BAR MECHANISM-THREE POSITION SYNTHESIS MATLAB
CODES

clc
clear all

%Arbitrary Taken Angle Values

Betha2=pi/10;       %18
Betha3=pi/5;        %36
Alpha2=-pi/9;       %-20
Alpha3=-5*pi/17;    %-53
Gamma2=pi/12;       %15
Gamma3=pi/5;        %36

Dis2=-pi/3;         %-60
Dis3=-5*pi/12;      %-75

%Distance between positions

p21=30;
p31=90;

%positions of P points
P21=p21*exp(1i*Dis2);
P31=p31*exp(1i*Dis3);

a(1,1)=exp(1i*Betha2)-1;
a(1,2)=exp(1i*Alpha2)-1;
a(2,1)=exp(1i*Betha3)-1;
a(2,2)=exp(1i*Alpha3)-1;



b(1,1)=P21;
b(1,2)=exp(1i*Alpha2)-1;
b(2,1)=P31;
b(2,2)=exp(1i*Alpha3)-1;

W=det(b)/det(a)              %Position of first link

c(1,1)=exp(1i*Betha2)-1;
c(1,2)=P21;
c(2,1)=exp(1i*Betha3)-1;
c(2,2)=P31;
```

```
Z=det(c)/det(a)                %Position of [PC]

A0=0-(1i*30);                %position of A point

P1=A0+Z+W;
A1=A0+W;

Wx=real(W);
Wy=imag(W);
P1x=real(P1);
P1y=imag(P1);
A1x=real(A1);
A1y=imag(A1);
A0x=real(A0);
A0y=imag(A0);

line([P1x,A1x],[P1y,A1y]);
hold on
line([A1x,A0x],[A1y,A0y]);
hold on
grid on

t(1,1)=exp(1i*Gamma2)-1;
t(1,2)=exp(1i*Alpha2)-1;
t(2,1)=exp(1i*Gamma3)-1;
t(2,2)=exp(1i*Alpha3)-1;

U=det(b)/det(t)                %Position of [DE]

n(1,1)=exp(1i*Gamma2)-1;
n(1,2)=P21;
n(2,1)=exp(1i*Gamma3)-1;
n(2,2)=P31;

S=det(n)/det(t)        %Position of [PD]

B0=P1-S-U;
B1=B0+U;

B1x=real(B1);
B1y=imag(B1);
B0x=real(B0);
B0y=imag(B0);

U1x=real(U);
U1y=imag(U);
S0x=real(S);
S0y=imag(S);

line([P1x,B1x],[P1y,B1y]);
hold on
line([B1x,B0x],[B1y,B0y]);
hold on
```

```
%SECOND AND THIRD POSITION

AngleW=atan((A1y-A0y)/(A1x-A0x))         %angle of [AC]
w=(((A1y-A0y)^2)+((A1x-A0x)^2))^(1/2);     %length of [AC]

W2=w*exp(1i*(AngleW+Betha2))        %second position of [AC]
W3=w*exp(1i*(AngleW+Betha3))        %third position of [AC]

AngleZ=atan((P1y-A1y)/(P1x-A1x))          %angle of [PC]
z=(((P1y-A1y)^2)+((P1x-A1x)^2))^(1/2);    %lenght of [PC]

Z2=z*exp(1i*(AngleZ+Alpha2))   %second position of [PC]
Z3=z*exp(1i*(AngleZ+Alpha3))   %third position of [PC]

P2=A0+W2+Z2;
A2=A0+W2;

P3=A0+W3+Z3;
A3=A0+W3;

AngleU=atan((B1y-B0y)/(B1x-B0x))    %angle of [DE]
u=(((B1y-B0y)^2)+((B1x-B0x)^2))^(1/2);   %length of [DE]

U2=u*exp(1i*(AngleU+Gamma2))      %second position of [DE]
U3=u*exp(1i*(AngleU+Gamma3))      %third position of [DE]

AngleS=atan((P1y-B1y)/(P1x-B1x))       %Angle of [PD]
s=(((P1y-B1y)^2)+((P1x-B1x)^2))^(1/2);   %legnth of [PD]

S2=s*exp(1i*(AngleS+Alpha2))   %second position of [PD]
S3=s*exp(1i*(AngleS+Alpha3))   %third position of [PD]

B2=P2-S2;
B3=P3-S3;
%
% B2=B0+U2;
% B3=B0+U3;

P2x=real(P2);
P2y=imag(P2);
A2x=real(A2);
A2y=imag(A2);
B2x=real(B2);
B2y=imag(B2);

P3x=real(P3);
P3y=imag(P3);
A3x=real(A3);
A3y=imag(A3);
B3x=real(B3);
B3y=imag(B3);

AngleZZ2=atan((P2y-A2y)/(P2x-A2x));  %Find angles
```

```
%Grahs

line([P2x,B2x],[P2y,B2y]);
hold on
line([B2x,B0x],[B2y,B0y]);
hold on
line([P3x,B3x],[P3y,B3y]);
hold on
line([B3x,B0x],[B3y,B0y]);
hold on

line([P2x,A2x],[P2y,A2y]);
hold on
line([A2x,A0x],[A2y,A0y]);
hold on
line([P3x,A3x],[P3y,A3y]);
hold on
line([A3x,A0x],[A3y,A0y]);
hold on
xlabel('x axis');
ylabel('y axis');
title('Three Position Synthesis');
```

Appendix B: Extension Spring Optimization MATLAB Codes for First Link

```
%EXTENSION SPRING OBJECTIVE FUNCTION FOR FIRST LINK

function  F=objective(x)

n=100;  %Steps

%Design Variables

% x(1)=k;
% x(2)=AngleBetha;
% x(3)=Ls0meter;
% x(4)=bmeter;
% x(5)=Sxmeter;
% x(6)=Symeter;

%Angles
Betha2=pi/10;      %18
Betha3=pi/5;       %36
Alpha2=-pi/9;      %-20

Alpha3=-5*pi/17;   %-53
Gamma2=pi/12;      %15
Gamma3=pi/5;       %36

Dis2=-pi/3;        %-60
Dis3=-5*pi/12;     %-75

%Link Positions
p21=30;
p31=90;

P21=p21*exp(1i*Dis2);
P31=p31*exp(1i*Dis3);

a(1,1)=exp(1i*Betha2)-1;
a(1,2)=exp(1i*Alpha2)-1;
a(2,1)=exp(1i*Betha3)-1;
a(2,2)=exp(1i*Alpha3)-1;



b(1,1)=P21;
b(1,2)=exp(1i*Alpha2)-1;
b(2,1)=P31;
b(2,2)=exp(1i*Alpha3)-1;

W=det(b)/det(a)

c(1,1)=exp(1i*Betha2)-1;
c(1,2)=P21;
c(2,1)=exp(1i*Betha3)-1;
c(2,2)=P31;
```

```
Z=det(c)/det(a)

A0=0-(1i*30);

P1=A0+Z+W;
A1=A0+W;

Wx=real(W);
Wy=imag(W);
P1x=real(P1);
P1y=imag(P1);
A1x=real(A1);
A1y=imag(A1);
A0x=real(A0);
A0y=imag(A0);

t(1,1)=exp(1i*Gamma2)-1;
t(1,2)=exp(1i*Alpha2)-1;
t(2,1)=exp(1i*Gamma3)-1;
t(2,2)=exp(1i*Alpha3)-1;

U=det(b)/det(t)

n(1,1)=exp(1i*Gamma2)-1;
n(1,2)=P21;
n(2,1)=exp(1i*Gamma3)-1;
n(2,2)=P31;

S=det(n)/det(t)

B0=P1-S-U;
B1=B0+U;

B1x=real(B1);
B1y=imag(B1);
B0x=real(B0);
B0y=imag(B0);

U1x=real(U);
U1y=imag(U);
S0x=real(S);
S0y=imag(S);

%SECOND AND THIRD POSITION

AngleW=atan((A1y-A0y)/(A1x-A0x))
w=(((A1y-A0y)^2)+((A1x-A0x)^2))^(1/2)

W2=w*exp(1i*(AngleW+Betha2))
W3=w*exp(1i*(AngleW+Betha3))

AngleZ=atan((P1y-A1y)/(P1x-A1x))
z=(((P1y-A1y)^2)+((P1x-A1x)^2))^(1/2);
```

```
Z2=z*exp(1i*(AngleZ+Alpha2))
Z3=z*exp(1i*(AngleZ+Alpha3))

P2=A0+W2+Z2;
A2=A0+W2;

P3=A0+W3+Z3;
A3=A0+W3;

AngleU=atan((B1y-B0y)/(B1x-B0x))
u=(((B1y-B0y)^2)+((B1x-B0x)^2))^(1/2)

U2=u*exp(1i*(AngleU+Gamma2))
U3=u*exp(1i*(AngleU+Gamma3))

AngleS=atan((P1y-B1y)/(P1x-B1x))
s=(((P1y-B1y)^2)+((P1x-B1x)^2))^(1/2);

S2=s*exp(1i*(AngleS+Alpha2))
S3=s*exp(1i*(AngleS+Alpha3))

B2=P2-S2;
B3=P3-S3;

P2x=real(P2);
P2y=imag(P2);
A2x=real(A2);
A2y=imag(A2);
B2x=real(B2);
B2y=imag(B2);

P3x=real(P3);
P3y=imag(P3);
A3x=real(A3);
A3y=imag(A3);
B3x=real(B3);
B3y=imag(B3);


%Coupler Link Angles

AngleCoupler1=atan((B1y-A1y)/(B1x-A1x))
AngleCoupler2=atan((B2y-A2y)/(B2x-A2x))
AngleCoupler3=atan((B3y-A3y)/(B3x-A3x))

CouplerLength1=(((B1y-A1y)^2)+((B1x-A1x)^2))^(1/2)
CouplerLength2=(((B2y-A2y)^2)+((B2x-A2x)^2))^(1/2)
CouplerLength3=(((B3y-A3y)^2)+((B3x-A3x)^2))^(1/2)

n=100; %Steps
```

```
%Angles During the Hood Motions

AngleCoupler=linspace(AngleCoupler1,AngleCoupler3,n);
AngleDW=linspace(AngleW,AngleW+Betha3,n);
AngleDU=linspace(AngleU,AngleU+Gamma3,n);
AngleDZ=linspace(AngleZ,AngleZ+Alpha3,n);

for i=1:n

A(1,1)=-CouplerLength1*sin(AngleCoupler(i));
A(1,2)=u*sin(AngleDU(i));
A(2,1)=CouplerLength1*cos(AngleCoupler(i));
A(2,2)=-u*cos(AngleDU(i));

B1(i)=w*sin(AngleDW(i));
B2(i)=-w*cos(AngleDW(i));

C(1,1)=B1(i);
C(1,2)=u*sin(AngleDU(i));
C(2,1)=B2(i);
C(2,2)=-u*cos(AngleDU(i));

D(1,1)=-CouplerLength1*sin(AngleCoupler(i));
D(1,2)=B1(i);
D(2,1)=CouplerLength1*cos(AngleCoupler(i));
D(2,2)=B2(i);

FW(i)=det(C)/det(A)                                  %Velocity
Coefficients

FQ(i)=det(D)/det(A)


AngleBetha=5*pi/180;   %Angle of spring link

%Point Y positions

Yh(i)=A0y+(w*sin(AngleDW(i)))+((z/2)*sin(AngleDZ(i)))
Yp(i)=A0y+(w*sin(AngleDW(i)))+(z*sin(AngleDZ(i)))

end


b=22; %cm. lenth of [AB]

for i=1:n


Kx(i)=[b*cos(AngleDW(i)+AngleBetha)];
Ky(i)=[b*sin(AngleDW(i)+AngleBetha)];

bx(i)=(A0x+Kx(i))/100;
by(i)=(A0y+Ky(i))/100;
```

```
Sx=0.17;     %positions of(x,y)
Sy=-0.24;

%Spring length
Ls(i)=[(((bx(i)-Sx)^2+(by(i)-Sy)^2))^(1/2)];

end

%Total weight of hood

G=126.3295;

%% Virtual Displacemet

zmeter=z/100;
wmeter=w/100;

Ls0=0.15;   %first spring length
% Ls0meter=0.05;

F=0;
%Position changes during the motion

for i=1:n

%(a) DLs

DLs(i)=[(((-x(4)*(bx(i)-x(5))*sin(AngleDW(i)+x(2)))+(x(4)*(by(i)-
x(6))*cos(AngleDW(i)+x(2))))/Ls(i))]

%(b) Dyh

Dyh(i)=[(wmeter*cos(AngleDW(i)))+((zmeter/2)*cos(AngleDZ(i))*FW(i))]

%(c) Dyp

Dyp(i)=[((wmeter*cos(AngleDW(i)))+(zmeter*cos(AngleDZ(i))*FW(i)))]

end

for i=1:n

Pa(i)=[(G*Dyh(i))+(x(1)*(Ls(i)-x(3)))*DLs(i)]/(Dyp(i))    %Main
Objective Function, Force

F=F+(Pa(i))^2;

end
```

```
%EXTENSION SPRING FIRST LINK APPLICATION SUBJECTIVE FUNCTION FOR
%OPTIMIZATION

function [c,ceq]=SubjectiveForce(x)

%Design variables
% x(1)=k;
% x(2)=AngleBetha;
% x(3)=Ls0meter;
% x(4)=bmeter;
% x(5)=Sxmeter;
% x(6)=Symeter;

%Boundry Values

c(1)=0-x(1);                %k
c(2)=x(1)-1000;

c(3)=x(2)-18*pi/180;        %AngleBetha
c(4)=-10*pi/180-x(2);

c(5)=0.05-x(3);             %Ls0meter
c(6)=x(3)-0.3;

c(7)=5-x(4);               %b
c(8)=x(4)-25;

c(9)=x(5)-0.4;             %Sx
c(10)=0-x(5);

c(11)=-x(6)-0.4;           %Sy
c(12)=x(6)-0;

ceq=[];

end
```

Appendix C: Compression Spring Optimization MATLAB Codes for First Link

```
%COMPRESSION SPRING OPTIMIZATION OBJECTIVE FUNCTION FOR FIRST LINK

function  F=objective(x)

n=100;  %Steps

%Design Variables
% x(1)=k;
% x(2)=Ls0meter;
% x(3)=bmeter;
% x(4)=Sxmeter;
% x(5)=Symeter;
% x(6)=AngleBetha;

%Angles
Betha2=pi/10;       %18
Betha3=pi/5;        %36
Alpha2=-pi/9;       %-20

Alpha3=-5*pi/17;    %-53
Gamma2=pi/12;       %15
Gamma3=pi/5;        %36

Dis2=-pi/3;         %-60
Dis3=-5*pi/12;      %-75

%Link Positions
p21=30;
p31=90;

P21=p21*exp(1i*Dis2);
P31=p31*exp(1i*Dis3);

a(1,1)=exp(1i*Betha2)-1;
a(1,2)=exp(1i*Alpha2)-1;
a(2,1)=exp(1i*Betha3)-1;
a(2,2)=exp(1i*Alpha3)-1;


b(1,1)=P21;
b(1,2)=exp(1i*Alpha2)-1;
b(2,1)=P31;
b(2,2)=exp(1i*Alpha3)-1;

W=det(b)/det(a)

c(1,1)=exp(1i*Betha2)-1;
c(1,2)=P21;
c(2,1)=exp(1i*Betha3)-1;
c(2,2)=P31;
```

```
Z=det(c)/det(a)

A0=0-(1i*30);

P1=A0+Z+W;
A1=A0+W;

Wx=real(W);
Wy=imag(W);
P1x=real(P1);
P1y=imag(P1);
A1x=real(A1);
A1y=imag(A1);
A0x=real(A0);
A0y=imag(A0);

t(1,1)=exp(1i*Gamma2)-1;
t(1,2)=exp(1i*Alpha2)-1;
t(2,1)=exp(1i*Gamma3)-1;
t(2,2)=exp(1i*Alpha3)-1;

U=det(b)/det(t)

n(1,1)=exp(1i*Gamma2)-1;
n(1,2)=P21;
n(2,1)=exp(1i*Gamma3)-1;
n(2,2)=P31;

S=det(n)/det(t)

B0=P1-S-U;
B1=B0+U;

B1x=real(B1);
B1y=imag(B1);
B0x=real(B0);
B0y=imag(B0);

U1x=real(U);
U1y=imag(U);
S0x=real(S);
S0y=imag(S);

%SECOND AND THIRD POSITION

AngleW=atan((A1y-A0y)/(A1x-A0x))
w=(((A1y-A0y)^2)+((A1x-A0x)^2))^(1/2)

W2=w*exp(1i*(AngleW+Betha2))
W3=w*exp(1i*(AngleW+Betha3))

AngleZ=atan((P1y-A1y)/(P1x-A1x))
z=(((P1y-A1y)^2)+((P1x-A1x)^2))^(1/2);
```

```
Z2=z*exp(1i*(AngleZ+Alpha2))
Z3=z*exp(1i*(AngleZ+Alpha3))

P2=A0+W2+Z2;
A2=A0+W2;

P3=A0+W3+Z3;
A3=A0+W3;

AngleU=pi+atan((B1y-B0y)/(B1x-B0x))
u=(((B1y-B0y)^2)+((B1x-B0x)^2))^(1/2)

U2=u*exp(1i*(AngleU+Gamma2))
U3=u*exp(1i*(AngleU+Gamma3))

AngleS=atan((P1y-B1y)/(P1x-B1x))
s=(((P1y-B1y)^2)+((P1x-B1x)^2))^(1/2);

S2=s*exp(1i*(AngleS+Alpha2))
S3=s*exp(1i*(AngleS+Alpha3))

B2=P2-S2;
B3=P3-S3;

P2x=real(P2);
P2y=imag(P2);
A2x=real(A2);
A2y=imag(A2);
B2x=real(B2);
B2y=imag(B2);

P3x=real(P3);
P3y=imag(P3);
A3x=real(A3);
A3y=imag(A3);
B3x=real(B3);
B3y=imag(B3);

%Coupler Link Angles
AngleCoupler1=atan((B1y-A1y)/(B1x-A1x))
AngleCoupler2=atan((B2y-A2y)/(B2x-A2x))
AngleCoupler3=atan((B3y-A3y)/(B3x-A3x))

CouplerLength1=(((B1y-A1y)^2)+((B1x-A1x)^2))^(1/2)
CouplerLength2=(((B2y-A2y)^2)+((B2x-A2x)^2))^(1/2)
CouplerLength3=(((B3y-A3y)^2)+((B3x-A3x)^2))^(1/2)

n=100;  %Steps
```

```
%Angles During the Hood Motions
AngleCoupler=linspace(AngleCoupler1,AngleCoupler3,n);
AngleDW=linspace(AngleW,AngleW+Betha3,n);
AngleDU=linspace(AngleU,AngleU+Gamma3,n);
AngleDZ=linspace(AngleZ,AngleZ+Alpha3,n);


for i=1:n

A(1,1)=-CouplerLength1*sin(AngleCoupler(i));
A(1,2)=u*sin(AngleDU(i));
A(2,1)=CouplerLength1*cos(AngleCoupler(i));
A(2,2)=-u*cos(AngleDU(i));

B1(i)=w*sin(AngleDW(i));
B2(i)=-w*cos(AngleDW(i));

C(1,1)=B1(i);
C(1,2)=u*sin(AngleDU(i));
C(2,1)=B2(i);
C(2,2)=-u*cos(AngleDU(i));

D(1,1)=-CouplerLength1*sin(AngleCoupler(i));
D(1,2)=B1(i);
D(2,1)=CouplerLength1*cos(AngleCoupler(i));
D(2,2)=B2(i);

FW(i)=det(C)/det(A)                          %Velocity Coefficients

FQ(i)=det(D)/det(A)

%Point Y positions

Yh(i)=A0y+(w*sin(AngleDW(i)))+((z/2)*sin(AngleDZ(i)))
Yp(i)=A0y+(w*sin(AngleDW(i)))+(z*sin(AngleDZ(i)))

end

AngleBetha=0;    %Angle of Spring Link

b=w/2;      %cm. length of [AB]

for i=1:n

% bx(i)=[b*cos(AngleDW(i)+AngleBetha)];
% by(i)=[b*sin(AngleDW(i)+AngleBetha)];
```

```
Kx(i)=(b)*cos(AngleDW(i)+AngleBetha);
Ky(i)=(b)*sin(AngleDW(i)+AngleBetha);

bx(i)=(A0x+Kx(i))/100;
by(i)=(A0y+Ky(i))/100;

Sx=-0.1;  %Positions of (x,y)
Sy=-0.2;

%Spring length
Ls(i)=[(((bx(i)-Sx)^2+(by(i)-Sy)^2))^(1/2)];

end

%Weight of Hood
G=126.3295;

%% Virtual Displacemet

zmeter=z/100;
wmeter=w/100;

Ls0=0.2; %First Link length
% Ls0meter=0.05;

F=0;

 %Position changes during the motion
for i=1:n

%(a) DLs

DLs(i)=[(((-x(3)*(bx(i)-x(4))*sin(AngleDW(i)+x(6)))+(x(3)*(by(i)-
x(5))*cos(AngleDW(i)+x(6))))/Ls(i))]

%(b) Dyh

Dyh(i)=[(wmeter*cos(AngleDW(i)))+((zmeter/2)*cos(AngleDZ(i))*FW(i))]

%(c) Dyp

Dyp(i)=[((wmeter*cos(AngleDW(i)))+(zmeter*cos(AngleDZ(i))*FW(i)))]

end

for i=1:n

Pa(i)=[(G*Dyh(i))-(-x(1)*(Ls(i)-x(2)))*DLs(i)]/(Dyp(i))    %Main
Objective Function, Force

F=F+(Pa(i))^2;
```

```
%COMPRESSION SPRING FIRST LINK APPLICATION SUBJECTIVE FUNCTION FOR
%OPTIMIZATION

function [c,ceq]=SubjectiveForce(x)

%Design variables
% x(1)=k;
% x(2)=Ls0meter;
% x(3)=bmeter;
% x(4)=Sxmeter;
% x(5)=Symeter;
% x(6)=AngleBetha;

%Boundary values
c(1)=650-x(1);                  %k
c(2)=x(1)-10000;

c(5)=0.38-x(2);                %Ls0meter
c(6)=x(2)-0.6;

c(7)=10-x(3);                 %b
c(8)=x(3)-36.42;

c(9)=x(4)+0.1;                  %Sx
c(10)=-0.5-x(4);

c(11)=-x(5)-0.7;               %Sy
c(12)=x(5)+0.4;

c(13)=x(6)-18*pi/180;        %AngleBetha
c(14)=-10*pi/180-x(6);

ceq=[];

end
```

Appendix D: Torsion Spring Optimization MATLAB Codes for First Link

```
%TORSION SPRING OPTIMIZATION OBJECTIVE FUNCTION FOR FIRST LINK

function  F=objective(x)

n=100;   %Steps

%Design variables
% x(1)=k;
% x(2)=AngleBetha;
% x(3)=Ls0meter;
% x(4)=bmeter;
% x(5)=Sxmeter;
% x(6)=Symeter;

%Angles
Betha2=pi/10;      %18
Betha3=pi/5;       %36
Alpha2=-pi/9;      %-20

Alpha3=-5*pi/17;   %-53
Gamma2=pi/12;      %15
Gamma3=pi/5;       %36

Dis2=-pi/3;        %-60
Dis3=-5*pi/12;     %-75

%Link Positions
p21=30;
p31=90;

P21=p21*exp(1i*Dis2);
P31=p31*exp(1i*Dis3);

a(1,1)=exp(1i*Betha2)-1;
a(1,2)=exp(1i*Alpha2)-1;
a(2,1)=exp(1i*Betha3)-1;
a(2,2)=exp(1i*Alpha3)-1;



b(1,1)=P21;
b(1,2)=exp(1i*Alpha2)-1;
b(2,1)=P31;
b(2,2)=exp(1i*Alpha3)-1;

W=det(b)/det(a)

c(1,1)=exp(1i*Betha2)-1;
c(1,2)=P21;
c(2,1)=exp(1i*Betha3)-1;
c(2,2)=P31;
```

```
Z=det(c)/det(a)

A0=0-(1i*30);

P1=A0+Z+W;
A1=A0+W;

Wx=real(W);
Wy=imag(W);
P1x=real(P1);
P1y=imag(P1);
A1x=real(A1);
A1y=imag(A1);
A0x=real(A0);
A0y=imag(A0);

t(1,1)=exp(1i*Gamma2)-1;
t(1,2)=exp(1i*Alpha2)-1;
t(2,1)=exp(1i*Gamma3)-1;
t(2,2)=exp(1i*Alpha3)-1;

U=det(b)/det(t)

n(1,1)=exp(1i*Gamma2)-1;
n(1,2)=P21;
n(2,1)=exp(1i*Gamma3)-1;
n(2,2)=P31;

S=det(n)/det(t)

B0=P1-S-U;
B1=B0+U;

B1x=real(B1);
B1y=imag(B1);
B0x=real(B0);
B0y=imag(B0);

U1x=real(U);
U1y=imag(U);
S0x=real(S);
S0y=imag(S);

%SECOND AND THIRD POSITION

AngleW=atan((A1y-A0y)/(A1x-A0x))
w=(((A1y-A0y)^2)+((A1x-A0x)^2))^(1/2)

W2=w*exp(1i*(AngleW+Betha2))
W3=w*exp(1i*(AngleW+Betha3))

AngleZ=atan((P1y-A1y)/(P1x-A1x))
z=(((P1y-A1y)^2)+((P1x-A1x)^2))^(1/2);
```

```
Z2=z*exp(1i*(AngleZ+Alpha2))
Z3=z*exp(1i*(AngleZ+Alpha3))

P2=A0+W2+Z2;
A2=A0+W2;

P3=A0+W3+Z3;
A3=A0+W3;

AngleU=atan((B1y-B0y)/(B1x-B0x))
u=(((B1y-B0y)^2)+((B1x-B0x)^2))^(1/2)

U2=u*exp(1i*(AngleU+Gamma2))
U3=u*exp(1i*(AngleU+Gamma3))

AngleS=atan((P1y-B1y)/(P1x-B1x))
s=(((P1y-B1y)^2)+((P1x-B1x)^2))^(1/2);

S2=s*exp(1i*(AngleS+Alpha2))
S3=s*exp(1i*(AngleS+Alpha3))

B2=P2-S2;
B3=P3-S3;

P2x=real(P2);
P2y=imag(P2);
A2x=real(A2);
A2y=imag(A2);
B2x=real(B2);
B2y=imag(B2);

P3x=real(P3);
P3y=imag(P3);
A3x=real(A3);
A3y=imag(A3);
B3x=real(B3);
B3y=imag(B3);

%Coupler Link Angles
AngleCoupler1=atan((B1y-A1y)/(B1x-A1x))
AngleCoupler2=atan((B2y-A2y)/(B2x-A2x))
AngleCoupler3=atan((B3y-A3y)/(B3x-A3x))

CouplerLength1=(((B1y-A1y)^2)+((B1x-A1x)^2))^(1/2)
CouplerLength2=(((B2y-A2y)^2)+((B2x-A2x)^2))^(1/2)
CouplerLength3=(((B3y-A3y)^2)+((B3x-A3x)^2))^(1/2)

n=100;  %steps
```

```
%Angles during the hood motions
AngleCoupler=linspace(AngleCoupler1,AngleCoupler3,n);
AngleDW=linspace(AngleW,AngleW+Betha3,n);
AngleDU=linspace(AngleU,AngleU+Gamma3,n);
AngleDZ=linspace(AngleZ,AngleZ+Alpha3,n);

for i=1:n

A(1,1)=-CouplerLength1*sin(AngleCoupler(i));
A(1,2)=u*sin(AngleDU(i));
A(2,1)=CouplerLength1*cos(AngleCoupler(i));
A(2,2)=-u*cos(AngleDU(i));

B1(i)=w*sin(AngleDW(i));
B2(i)=-w*cos(AngleDW(i));

C(1,1)=B1(i);
C(1,2)=u*sin(AngleDU(i));
C(2,1)=B2(i);
C(2,2)=-u*cos(AngleDU(i));

D(1,1)=-CouplerLength1*sin(AngleCoupler(i));
D(1,2)=B1(i);
D(2,1)=CouplerLength1*cos(AngleCoupler(i));
D(2,2)=B2(i);

FW(i)=det(C)/det(A)                                %Velocity
Coefficients

FQ(i)=det(D)/det(A)


AngleBetha=5*pi/180;   %Angle of spring links

%Point Y positions
Yh(i)=A0y+(w*sin(AngleDW(i)))+((z/2)*sin(AngleDZ(i)))
Yp(i)=A0y+(w*sin(AngleDW(i)))+(z*sin(AngleDZ(i)))

end


b=0.25;   %cm. lenght of spring link

for i=1:n

bx(i)=[b*cos(AngleDW(i)+AngleBetha)];
by(i)=[b*sin(AngleDW(i)+AngleBetha)];

Sx=0.17;   %position of (x,y)
Sy=-0.24;

%Spring lenght
Ls(i)=[(((bx(i)-Sx)^2+(by(i)-Sy)^2))^(1/2)];

end
```

```
%Weight of hood
G=126.3295;

%% Virtual Displacemet

zmeter=z/100;
wmeter=w/100;

Ls0=0.15;   %spring lengt
% Ls0meter=0.05;

F=0;

%Position changes during the motion
for i=1:n

%(a) DLs

% DLs(i)=((((-b*(bx(i)-Sx)*sin(AngleDW(i)+AngleBetha))+(b*(by(i)-
Sy)*cos(AngleDW(i)+AngleBetha)))/Ls(i)))

%(b) Dyh

Dyh(i)=((wmeter*cos(AngleDW(i)))+((zmeter/2)*cos(AngleDZ(i))*FW(i)
))

%(c) Dyp

Dyp(i)=(((wmeter*cos(AngleDW(i)))+(zmeter*cos(AngleDZ(i))*FW(i))))

end


ks=100;     %N.m/radian
Qs=45*pi/180;
Ts=ks*Qs;

%% Force
n=100;

for i=1:n

Pa(i)=[(G*Dyh(i))+(x(1)*(AngleDW(i)-((AngleW)-x(2))))]/(Dyp(i));
%Main Objective Function, Force

F=F+(Pa(i))^2;

end
```

```
%TORSION SPRING FIRST LINK APPLICATION SUBJECTIVE FUNCTION FOR
%OPTIMIZATION

function [c,ceq]=SubjectiveForce(x)

%Design Variables

% x(1)=ks;
% x(2)=Qs

%Boundary Values

c(1)=0-x(1);              %ks
c(2)=x(1)-1000;

c(3)=x(2)-150*pi/180;      %Qs
c(4)=30*pi/180-x(2);

ceq=[];

end
```

Appendix E: Extension Spring Optimization MATLAB Codes for Second Link

```
%EXTENSION SPRING OPTIMIZATION OBJECTIVE FUNCTION FOR SECOND LINK

function  F=objective(x)

n=100; %Steps

%Design variables
% x(1)=k;
% x(2)=AngleBetha;
% x(3)=Ls0meter;
% x(4)=bmeter;
% x(5)=Sxmeter;
% x(6)=Symeter;

%Angles
Betha2=pi/10;       %18
Betha3=pi/5;        %36
Alpha2=-pi/9;       %-20

Alpha3=-5*pi/17;    %-53
Gamma2=pi/12;       %15
Gamma3=pi/5;        %36

Dis2=-pi/3;         %-60
Dis3=-5*pi/12;      %-75

%Link positions
p21=30;
p31=90;

P21=p21*exp(1i*Dis2);
P31=p31*exp(1i*Dis3);

a(1,1)=exp(1i*Betha2)-1;
a(1,2)=exp(1i*Alpha2)-1;
a(2,1)=exp(1i*Betha3)-1;
a(2,2)=exp(1i*Alpha3)-1;



b(1,1)=P21;
b(1,2)=exp(1i*Alpha2)-1;
b(2,1)=P31;
b(2,2)=exp(1i*Alpha3)-1;

W=det(b)/det(a)

c(1,1)=exp(1i*Betha2)-1;
c(1,2)=P21;
c(2,1)=exp(1i*Betha3)-1;
c(2,2)=P31;
```

```
Z=det(c)/det(a)

A0=0-(1i*30);

P1=A0+Z+W;
A1=A0+W;

Wx=real(W);
Wy=imag(W);
P1x=real(P1);
P1y=imag(P1);
A1x=real(A1);
A1y=imag(A1);
A0x=real(A0);
A0y=imag(A0);

t(1,1)=exp(1i*Gamma2)-1;
t(1,2)=exp(1i*Alpha2)-1;
t(2,1)=exp(1i*Gamma3)-1;
t(2,2)=exp(1i*Alpha3)-1;

U=det(b)/det(t)

n(1,1)=exp(1i*Gamma2)-1;
n(1,2)=P21;
n(2,1)=exp(1i*Gamma3)-1;
n(2,2)=P31;

S=det(n)/det(t)

B0=P1-S-U;
B1=B0+U;

B1x=real(B1);
B1y=imag(B1);
B0x=real(B0);
B0y=imag(B0);

U1x=real(U);
U1y=imag(U);
S0x=real(S);
S0y=imag(S);

%SECOND AND THIRD POSITION

AngleW=atan((A1y-A0y)/(A1x-A0x))
w=(((A1y-A0y)^2)+((A1x-A0x)^2))^(1/2)

W2=w*exp(1i*(AngleW+Betha2))
W3=w*exp(1i*(AngleW+Betha3))

AngleZ=atan((P1y-A1y)/(P1x-A1x))
z=(((P1y-A1y)^2)+((P1x-A1x)^2))^(1/2);
```

```
Z2=z*exp(1i*(AngleZ+Alpha2))
Z3=z*exp(1i*(AngleZ+Alpha3))

P2=A0+W2+Z2;
A2=A0+W2;

P3=A0+W3+Z3;
A3=A0+W3;

AngleU=pi+atan((B1y-B0y)/(B1x-B0x))
u=(((B1y-B0y)^2)+((B1x-B0x)^2))^(1/2)

U2=u*exp(1i*(AngleU+Gamma2))
U3=u*exp(1i*(AngleU+Gamma3))

AngleS=atan((P1y-B1y)/(P1x-B1x))
s=(((P1y-B1y)^2)+((P1x-B1x)^2))^(1/2);

S2=s*exp(1i*(AngleS+Alpha2))
S3=s*exp(1i*(AngleS+Alpha3))

B2=P2-S2;
B3=P3-S3;

P2x=real(P2);
P2y=imag(P2);
A2x=real(A2);
A2y=imag(A2);
B2x=real(B2);
B2y=imag(B2);

P3x=real(P3);
P3y=imag(P3);
A3x=real(A3);
A3y=imag(A3);
B3x=real(B3);
B3y=imag(B3);

%Coupler link angles
AngleCoupler1=atan((B1y-A1y)/(B1x-A1x))
AngleCoupler2=atan((B2y-A2y)/(B2x-A2x))
AngleCoupler3=atan((B3y-A3y)/(B3x-A3x))

CouplerLength1=(((B1y-A1y)^2)+((B1x-A1x)^2))^(1/2)
CouplerLength2=(((B2y-A2y)^2)+((B2x-A2x)^2))^(1/2)
CouplerLength3=(((B3y-A3y)^2)+((B3x-A3x)^2))^(1/2)

n=100; %steps
```

```
%Angles during the hood motion
AngleCoupler=linspace(AngleCoupler1,AngleCoupler3,n);
AngleDW=linspace(AngleW,AngleW+Betha3,n);
AngleDU=linspace(AngleU,AngleU+Gamma3,n);
AngleDZ=linspace(AngleZ,AngleZ+Alpha3,n);


for i=1:n

A(1,1)=-CouplerLength1*sin(AngleCoupler(i));
A(1,2)=u*sin(AngleDU(i));
A(2,1)=CouplerLength1*cos(AngleCoupler(i));
A(2,2)=-u*cos(AngleDU(i));

B1(i)=w*sin(AngleDW(i));
B2(i)=-w*cos(AngleDW(i));

C(1,1)=B1(i);
C(1,2)=u*sin(AngleDU(i));
C(2,1)=B2(i);
C(2,2)=-u*cos(AngleDU(i));

D(1,1)=-CouplerLength1*sin(AngleCoupler(i));
D(1,2)=B1(i);
D(2,1)=CouplerLength1*cos(AngleCoupler(i));
D(2,2)=B2(i);

FW(i)=det(C)/det(A)                                %Velocity
Coefficients

FQ(i)=det(D)/det(A)


AngleBetha=5*pi/180;  %Angle of spring link

%Point Y positions

Yh(i)=A0y+(w*sin(AngleDW(i)))+((z/2)*sin(AngleDZ(i)))
Yp(i)=A0y+(w*sin(AngleDW(i)))+(z*sin(AngleDZ(i)))

end


b=20;   %cm. length of [AB]

for i=1:n

% bx(i)=[b*cos(AngleDW(i)+AngleBetha)];
% by(i)=[b*sin(AngleDW(i)+AngleBetha)];
```

```
Kx(i)=(b)*cos(AngleDU(i)+AngleBetha);
Ky(i)=(b)*sin(AngleDU(i)+AngleBetha);
bx(i)=(B0x+Kx(i))/100;
by(i)=(B0y+Ky(i))/100;

Sx=0.17;    %Positions of (x,y)
Sy=-0.24;

%Spring length
Ls(i)=[(((bx(i)-Sx)^2+(by(i)-Sy)^2))^(1/2)];

end

%weight of hood
G=126.3295;

%% Virtual Displacemet

zmeter=z/100;
wmeter=w/100;

Ls0=0.15;
% Ls0meter=0.05;

F=0;

%Position changes during the motion
for i=1:n

%(a) DLs

DLs(i)=[(((-x(4)*(bx(i)-x(5))*sin(AngleDU(i)+x(2)))+(x(4)*(by(i)-
x(6))*cos(AngleDU(i)+x(2))))/Ls(i))]

%(b) Dyh

Dyh(i)=[(wmeter*cos(AngleDW(i)))+((zmeter/2)*cos(AngleDZ(i))*FW(i))]

%(c) Dyp

Dyp(i)=[((wmeter*cos(AngleDW(i)))+(zmeter*cos(AngleDZ(i))*FW(i)))]

end

for i=1:n

Pa(i)=[(G*Dyh(i))+(x(1)*(Ls(i)-x(3)))*DLs(i)]/(Dyp(i))  %Main
Objective Function, Force

F=F+(Pa(i))^2;

end
```

```
%EXTENSION SPRING SECOND LINK APPLICATION SUBJECTIVE FUNCTION FOR
OPTIMIZATION

function [c,ceq]=SubjectiveForce(x)

%Design Variables
% x(1)=k;
% x(2)=AngleBetha;
% x(3)=Ls0meter;
% x(4)=bmeter;
% x(5)=Sxmeter;
% x(6)=Symeter;

%Boundary Values
c(1)=10-x(1);                %k
c(2)=x(1)-1000;

c(3)=x(2)-18*pi/180;        %AngleBetha
c(4)=-10*pi/180-x(2);

c(5)=0.05-x(3);             %Ls0meter
c(6)=x(3)-0.3;

c(7)=5-x(4);               %b
c(8)=x(4)-20;

c(9)=x(5)-0.4;             %Sx
c(10)=0-x(5);

c(11)=-x(6)-0.4;           %Sy
c(12)=x(6)-0;

ceq=[];

end
```

Appendix F: Compression Spring Optimization MATLAB Codes for Second Link

```
%COMPRESSION SPRING OPTIMIZATION OBJECTIVE FUNCTION FOR SECOND
LINK

function  F=objective(x)

n=100; %Steps

%Design Variables
% x(1)=k;
% x(2)=Ls0meter;
% x(3)=bmeter;
% x(4)=Sxmeter;
% x(5)=Symeter;
% x(6)=AngleBetha;

%Angles
Betha2=pi/10;       %18
Betha3=pi/5;        %36
Alpha2=-pi/9;       %-20

Alpha3=-5*pi/17;    %-53
Gamma2=pi/12;       %15
Gamma3=pi/5;        %36

Dis2=-pi/3;         %-60
Dis3=-5*pi/12;      %-75

%Link Positions
p21=30;
p31=90;

P21=p21*exp(1i*Dis2);
P31=p31*exp(1i*Dis3);

a(1,1)=exp(1i*Betha2)-1;
a(1,2)=exp(1i*Alpha2)-1;
a(2,1)=exp(1i*Betha3)-1;
a(2,2)=exp(1i*Alpha3)-1;

b(1,1)=P21;
b(1,2)=exp(1i*Alpha2)-1;
b(2,1)=P31;
b(2,2)=exp(1i*Alpha3)-1;

W=det(b)/det(a)

c(1,1)=exp(1i*Betha2)-1;
c(1,2)=P21;
c(2,1)=exp(1i*Betha3)-1;
c(2,2)=P31;
```

```
Z=det(c)/det(a)

A0=0-(1i*30);

P1=A0+Z+W;
A1=A0+W;

Wx=real(W);
Wy=imag(W);
P1x=real(P1);
P1y=imag(P1);
A1x=real(A1);
A1y=imag(A1);
A0x=real(A0);
A0y=imag(A0);

t(1,1)=exp(1i*Gamma2)-1;
t(1,2)=exp(1i*Alpha2)-1;
t(2,1)=exp(1i*Gamma3)-1;
t(2,2)=exp(1i*Alpha3)-1;

U=det(b)/det(t)

n(1,1)=exp(1i*Gamma2)-1;
n(1,2)=P21;
n(2,1)=exp(1i*Gamma3)-1;
n(2,2)=P31;

S=det(n)/det(t)

B0=P1-S-U;
B1=B0+U;

B1x=real(B1);
B1y=imag(B1);
B0x=real(B0);
B0y=imag(B0);

U1x=real(U);
U1y=imag(U);
S0x=real(S);
S0y=imag(S);

%SECOND AND THIRD POSITION

AngleW=atan((A1y-A0y)/(A1x-A0x))
w=(((A1y-A0y)^2)+((A1x-A0x)^2))^(1/2)

W2=w*exp(1i*(AngleW+Betha2))
W3=w*exp(1i*(AngleW+Betha3))

AngleZ=atan((P1y-A1y)/(P1x-A1x))
z=(((P1y-A1y)^2)+((P1x-A1x)^2))^(1/2);
```

```
Z2=z*exp(1i*(AngleZ+Alpha2))
Z3=z*exp(1i*(AngleZ+Alpha3))

P2=A0+W2+Z2;
A2=A0+W2;

P3=A0+W3+Z3;
A3=A0+W3;

AngleU=pi+atan((B1y-B0y)/(B1x-B0x))
u=(((B1y-B0y)^2)+((B1x-B0x)^2))^(1/2)

U2=u*exp(1i*(AngleU+Gamma2))
U3=u*exp(1i*(AngleU+Gamma3))

AngleS=atan((P1y-B1y)/(P1x-B1x))
s=(((P1y-B1y)^2)+((P1x-B1x)^2))^(1/2);

S2=s*exp(1i*(AngleS+Alpha2))
S3=s*exp(1i*(AngleS+Alpha3))

B2=P2-S2;
B3=P3-S3;

P2x=real(P2);
P2y=imag(P2);
A2x=real(A2);
A2y=imag(A2);
B2x=real(B2);
B2y=imag(B2);

P3x=real(P3);
P3y=imag(P3);
A3x=real(A3);
A3y=imag(A3);
B3x=real(B3);
B3y=imag(B3);

%Coupler Link Angles
AngleCoupler1=atan((B1y-A1y)/(B1x-A1x))
AngleCoupler2=atan((B2y-A2y)/(B2x-A2x))
AngleCoupler3=atan((B3y-A3y)/(B3x-A3x))

CouplerLength1=(((B1y-A1y)^2)+((B1x-A1x)^2))^(1/2)
CouplerLength2=(((B2y-A2y)^2)+((B2x-A2x)^2))^(1/2)
CouplerLength3=(((B3y-A3y)^2)+((B3x-A3x)^2))^(1/2)

n=100;  %steps
```

```
%Angles during the hood motion
AngleCoupler=linspace(AngleCoupler1,AngleCoupler3,n);
AngleDW=linspace(AngleW,AngleW+Betha3,n);
AngleDU=linspace(AngleU,AngleU+Gamma3,n);
AngleDZ=linspace(AngleZ,AngleZ+Alpha3,n);


for i=1:n

A(1,1)=-CouplerLength1*sin(AngleCoupler(i));
A(1,2)=u*sin(AngleDU(i));
A(2,1)=CouplerLength1*cos(AngleCoupler(i));
A(2,2)=-u*cos(AngleDU(i));

B1(i)=w*sin(AngleDW(i));
B2(i)=-w*cos(AngleDW(i));

C(1,1)=B1(i);
C(1,2)=u*sin(AngleDU(i));
C(2,1)=B2(i);
C(2,2)=-u*cos(AngleDU(i));

D(1,1)=-CouplerLength1*sin(AngleCoupler(i));
D(1,2)=B1(i);
D(2,1)=CouplerLength1*cos(AngleCoupler(i));
D(2,2)=B2(i);

FW(i)=det(C)/det(A)                                  %Velocity
Coefficients

FQ(i)=det(D)/det(A)


AngleBetha=0;  %Angle of spring link

%Point Y Positions
Yh(i)=A0y+(w*sin(AngleDW(i)))+((z/2)*sin(AngleDZ(i)))
Yp(i)=A0y+(w*sin(AngleDW(i)))+(z*sin(AngleDZ(i)))

end


b=u/2;   %length of [EB]

for i=1:n

% bx(i)=[b*cos(AngleDW(i)+AngleBetha)];
% by(i)=[b*sin(AngleDW(i)+AngleBetha)];
```

```
Kx(i)=(b)*cos(AngleDW(i)+AngleBetha);
Ky(i)=(b)*sin(AngleDW(i)+AngleBetha);
bx(i)=(B0x+Kx(i))/100;
by(i)=(B0y+Ky(i))/100;

Sx=0.15;   %Positions of (x,y)
Sy=-0.20;

%Spring length
Ls(i)=[(((bx(i)-Sx)^2+(by(i)-Sy)^2))^(1/2)];

end

%Weight of hood
G=126.3295;

%% Virtual Displacemet

zmeter=z/100;
wmeter=w/100;

Ls0=0.15;  %spring length
% Ls0meter=0.05;

F=0;

%Positions changes during the motion
for i=1:n

%(a) DLs

DLs(i)=[(((-x(3)*(bx(i)-x(4))*sin(AngleDW(i)+x(6)))+(x(3)*(by(i)-
x(5))*cos(AngleDW(i)+x(6))))/Ls(i))]

%(b) Dyh

Dyh(i)=[(wmeter*cos(AngleDW(i)))+((zmeter/2)*cos(AngleDZ(i))*FW(i))]

%(c) Dyp

Dyp(i)=[((wmeter*cos(AngleDW(i)))+(zmeter*cos(AngleDZ(i))*FW(i)))]

end

for i=1:n

Pa(i)=[(G*Dyh(i))-(-x(1)*(x(6)-x(2)))*DLs(i)]/(Dyp(i))   %Main
Objective Function, Force

F=F+(Pa(i))^2;

end
```

```
%COMPRESSION SPRING SECOND LINK APPLICATION SUBJECTIVE FUNCTION
FOR OPTIMIZATION

function [c,ceq]=SubjectiveForce(x)

%Design Variables
% x(1)=k;
% x(2)=Ls0meter;
% x(3)=bmeter;
% x(4)=Sxmeter;
% x(5)=Symeter;
% x(6)=AngleBetha;

%Boundary Values
c(1)=40-x(1);           %k
c(2)=x(1)-10000;

c(5)=0.3-x(2);           %Ls0meter
c(6)=x(2)-0.5;

c(7)=25-x(3);           %b
c(8)=x(3)-40;

c(9)=x(4)-0.05;          %Sx
c(10)=-0.1-x(4);

c(11)=-x(5)-0.1;         %Sy
c(12)=x(5)+0.2;

c(13)=x(6)-18*pi/180;       %AngleBetha
c(14)=-10*pi/180-x(6);

ceq=[];

end
```

Appendix G: Torsion Spring Optimization MATLAB Codes for Second Link

```
%TORSION SPRING OPTIMIZATION OBJECTIVE FUNCTION FOR SECOND LINK

function  F=objective(x)

n=100; %Steps
%Design Variables
% x(1)=ks;
% x(2)=Qs

%Angles
Betha2=pi/10;       %18
Betha3=pi/5;        %36
Alpha2=-pi/9;       %-20

Alpha3=-5*pi/17;    %-53
Gamma2=pi/12;       %15
Gamma3=pi/5;        %36

Dis2=-pi/3;         %-60
Dis3=-5*pi/12;      %-75

%Link Positions
p21=30;
p31=90;

P21=p21*exp(1i*Dis2);
P31=p31*exp(1i*Dis3);

a(1,1)=exp(1i*Betha2)-1;
a(1,2)=exp(1i*Alpha2)-1;
a(2,1)=exp(1i*Betha3)-1;
a(2,2)=exp(1i*Alpha3)-1;


b(1,1)=P21;
b(1,2)=exp(1i*Alpha2)-1;
b(2,1)=P31;
b(2,2)=exp(1i*Alpha3)-1;

W=det(b)/det(a)

c(1,1)=exp(1i*Betha2)-1;
c(1,2)=P21;
c(2,1)=exp(1i*Betha3)-1;
c(2,2)=P31;
```

```
Z=det(c)/det(a)

A0=0-(1i*30);

P1=A0+Z+W;
A1=A0+W;

Wx=real(W);
Wy=imag(W);
P1x=real(P1);
P1y=imag(P1);
A1x=real(A1);
A1y=imag(A1);
A0x=real(A0);
A0y=imag(A0);

t(1,1)=exp(1i*Gamma2)-1;
t(1,2)=exp(1i*Alpha2)-1;
t(2,1)=exp(1i*Gamma3)-1;
t(2,2)=exp(1i*Alpha3)-1;

U=det(b)/det(t)

n(1,1)=exp(1i*Gamma2)-1;
n(1,2)=P21;
n(2,1)=exp(1i*Gamma3)-1;
n(2,2)=P31;

S=det(n)/det(t)

B0=P1-S-U;
B1=B0+U;

B1x=real(B1);
B1y=imag(B1);
B0x=real(B0);
B0y=imag(B0);

U1x=real(U);
U1y=imag(U);
S0x=real(S);
S0y=imag(S);

%SECOND AND THIRD POSITION

AngleW=atan((A1y-A0y)/(A1x-A0x))
w=(((A1y-A0y)^2)+((A1x-A0x)^2))^(1/2)

W2=w*exp(1i*(AngleW+Betha2))
W3=w*exp(1i*(AngleW+Betha3))

AngleZ=atan((P1y-A1y)/(P1x-A1x))
z=(((P1y-A1y)^2)+((P1x-A1x)^2))^(1/2);
```

```
Z2=z*exp(1i*(AngleZ+Alpha2))
Z3=z*exp(1i*(AngleZ+Alpha3))

P2=A0+W2+Z2;
A2=A0+W2;

P3=A0+W3+Z3;
A3=A0+W3;

AngleU=atan((B1y-B0y)/(B1x-B0x))
u=(((B1y-B0y)^2)+((B1x-B0x)^2))^(1/2)

U2=u*exp(1i*(AngleU+Gamma2))
U3=u*exp(1i*(AngleU+Gamma3))

AngleS=atan((P1y-B1y)/(P1x-B1x))
s=(((P1y-B1y)^2)+((P1x-B1x)^2))^(1/2);

S2=s*exp(1i*(AngleS+Alpha2))
S3=s*exp(1i*(AngleS+Alpha3))

B2=P2-S2;
B3=P3-S3;

P2x=real(P2);
P2y=imag(P2);
A2x=real(A2);
A2y=imag(A2);
B2x=real(B2);
B2y=imag(B2);

P3x=real(P3);
P3y=imag(P3);
A3x=real(A3);
A3y=imag(A3);
B3x=real(B3);
B3y=imag(B3);

%Coupler Link Angles
AngleCoupler1=atan((B1y-A1y)/(B1x-A1x))
AngleCoupler2=atan((B2y-A2y)/(B2x-A2x))
AngleCoupler3=atan((B3y-A3y)/(B3x-A3x))

CouplerLength1=(((B1y-A1y)^2)+((B1x-A1x)^2))^(1/2)
CouplerLength2=(((B2y-A2y)^2)+((B2x-A2x)^2))^(1/2)
CouplerLength3=(((B3y-A3y)^2)+((B3x-A3x)^2))^(1/2)

n=100;  %Steps
```

```
%Angles during the hood motion
AngleCoupler=linspace(AngleCoupler1,AngleCoupler3,n);
AngleDW=linspace(AngleW,AngleW+Betha3,n);
AngleDU=linspace(AngleU,AngleU+Gamma3,n);
AngleDZ=linspace(AngleZ,AngleZ+Alpha3,n);



for i=1:n

A(1,1)=-CouplerLength1*sin(AngleCoupler(i));
A(1,2)=u*sin(AngleDU(i));
A(2,1)=CouplerLength1*cos(AngleCoupler(i));
A(2,2)=-u*cos(AngleDU(i));

B1(i)=w*sin(AngleDW(i));
B2(i)=-w*cos(AngleDW(i));

C(1,1)=B1(i);
C(1,2)=u*sin(AngleDU(i));
C(2,1)=B2(i);
C(2,2)=-u*cos(AngleDU(i));

D(1,1)=-CouplerLength1*sin(AngleCoupler(i));
D(1,2)=B1(i);
D(2,1)=CouplerLength1*cos(AngleCoupler(i));
D(2,2)=B2(i);

FW(i)=det(C)/det(A)                                %Velocity
Coefficients

FQ(i)=det(D)/det(A)


AngleBetha=5*pi/180;     %Angle of spring

%Point Y Positons
Yh(i)=A0y+(w*sin(AngleDW(i)))+((z/2)*sin(AngleDZ(i)))
Yp(i)=A0y+(w*sin(AngleDW(i)))+(z*sin(AngleDZ(i)))

end


b=0;  %lengthe of [EB]

for i=1:n


bx(i)=[b*cos(AngleDW(i)+AngleBetha)];
by(i)=[b*sin(AngleDW(i)+AngleBetha)];
```

```
Sx=0;  %Position of (x,y)
Sy=0;

%Length of spring becomes 0
Ls(i)=[(((bx(i)-Sx)^2+(by(i)-Sy)^2))^(1/2)];

end

%Weight of hood
G=126.3295;

%% Virtual Displacemet

zmeter=z/100;
wmeter=w/100;

Ls0=0.15;  %length of spring becomes 0
% Ls0meter=0.05;

F=0;

%Position Changes during the motion
for i=1:n

%(a) DLs

% DLs(i)=[(((-x(4)*(bx(i)-
x(5))*sin(AngleDW(i)+x(2)))+(x(4)*(by(i)-
x(6))*cos(AngleDW(i)+x(2))))/Ls(i))]

%(b) Dyh

Dyh(i)=[(wmeter*cos(AngleDW(i)))+((zmeter/2)*cos(AngleDZ(i))*FW(i
))]

%(c) Dyp

Dyp(i)=[((wmeter*cos(AngleDW(i)))+(zmeter*cos(AngleDZ(i))*FW(i)))
]

end

for i=1:n

Pa(i)=[(G*Dyh(i))+(x(1)*(AngleDU(i)-((AngleU)-x(2))))]/(Dyp(i));
%Main Objective Function, Force

F=F+(Pa(i))^2;

end
```

```
%TORSION SPRING SECOND LINK APPLICATION SUBJECTIVE FUNCTION FOR
OPTIMIZATION

function [c,ceq]=SubjectiveForce(x)

%Design Variables
% x(1)=k;
% x(2)=Qs

%Boundary Values

c(1)=0-x(1);              %k
c(2)=x(1)-666.93;

c(3)=x(2)-150*pi/180;     %Qs
c(4)=45*pi/180-x(2);

ceq=[];

end
```

During all optimization process, MATLAB Optimization Toolbox was used.

BIOGRAPHY

Onur Denizhan is a Master of Science degree candidate in the Mechanical Engineering and Mechanics Department at Lehigh University. He received the Bachelor of Science degree in the Department of Mechanical Engineering from Inonu University in 2011. His research interests are optimization of dynamic systems, dynamic modeling, mechanisms synthesis and analysis, control of mechanism systems. He had joined some research teams during his undergraduate and graduate years. His team was awarded the Best Design Team Award in 2011 and he won a scholarship to study abroad for his Master's and Ph.D. studies from Turkish Ministry of Education in 2011. He will be a doctoral candidate in the Department of Mechanical Engineering at Columbia University in New York City upon graduation.