Mechanical Design for Track Robot Climbing Stairs

Homayoun Rastan

Thesis submitted to the Faculty of Graduate and Postdoctoral Studie

In partial fulfillment of the requirements of

MASTER OF APPLIED SCIENCES

In Mechanical Engineering

University of Ottawa

Ottawa Canada

July 2011

Abstract

The purpose of this study was to find the best robot configuration for climbing and descending stairs, in addition to traveling on flat surfaces. Candidate robot types were analyzed to find the most suitable one for further study, based on stability, size, and energy consumption. Based on these considerations, the non-variable configuration tracked robot type was selected.

The basic robot parameters (minimum track size, comparison of tracks with grousers vs. tracks without grousers, track angle of attack) were determined using static analysis methods and using North American standards for the stair geometry. Dynamic analysis methods were then employed to refine the geometry and ensure the stability of the robot when climbing and descending stairs. The final design was then simulated in Matlab to profile the device's velocity, acceleration, and power consumption during the stair climbing and descending phases. A prototype robot was constructed.

The results of this study show that a non-variable tracked robot can be constructed for the purpose of climbing stairs by applying static and dynamic analysis techniques to optimize a design. This study provides the groundwork for this design, which can also serve as a basis for designing robots with other configurations.

Acknowledgment

The successful completion of this research over this two-year period represents the culmination of many people's contributions. I would like to first express my sincere gratitude to my supervisor, Dr. Atef Fahim, for his support and guidance of my studies on this research project. I truly appreciate his mentorship and constructive comments to achieve my goals.

I would like to express my gratitude to the University of Ottawa, Department of Mechanical Engineering faculty members.

My warmest thanks to machine shop staff, John Perrins, James MacDermid, and Leo Denner, as well as my friends, Philippe Girault and Steven Recoskie.

I would also like to extend my gratitude to my brothers in law, Dr. Simon McCall and Scott Swain, for their support and kindness during this period.

Finally, special thanks and love to my wife, Sheida Sadeghi, for her encouraging, inspiring, and unconditional love during this period of studying.

Table of Contents

Abstract.....ii

Acknowledgments.....iii

Table of Contents.....iv

List of Figures.....viii

List of Tables.....xiii

Nomenclature.....xiv

Chapter 1 1

1.1 Motivation.....1

1.2 Chapters organization.....6

Chapter 2 8

2.1 Literature Review 8

2.2 Walking Machines (Leg Mechanisms) 9

2.2.1 Walking machines advantages.....9

2.2.2 Walking machines disadvantages 9

2.3 Wheeled Robots.....11

2.3.1 Wheeled machines advantages.....11

2.3.2 Wheeled machines disadvantages.....11

2.4 Three-wheel and Star-wheel (Walking and rolling machines).....12

2.5 Star-wheel (MSRox) 13
2.5.1 Star-Wheel machines advantages 14
2.5.2 Star-Wheel machines disadvantages 14
2.6 Tracked robots 15
2.6.1 Tracked robot machines advantages 17
2.6.2 Tracked robot machines disadvantages 17
2.7 Tracked Robot Analysis 17
2.8 Analysis of Fixed Geometry Tracked Robot when Climbing Stairs 18
2.9 Climbing Stability Analysis 21
2.10 Tracked Robot Turning Analysis 22
2.11 Other considerations 23
2.12 Conclusion 24

Chapter 3 26

3.1 Static Analysis 26
3.1.1 Introduction 26
3.1.2 Geometry of Stairs According to the Building Code 27
3.1.3 Necessary Front Geometry for a Stair Climbing TrackedRobot 27
3.1.4 Case of Smooth Track: 31
3.1.5 Case of a Track with Grousers: 34
3.1.6 Pose Analysis of the Tracked Robot as it Climbs Stairs 37
3.1.7 Necessary location of Center of Mass for successful Ascentand Descent 44
3.1.8 Track Shape and Dimensions 45
3.2 Conclusion 47

Chapter 4 48

4.1 Dynamic Analysis of the Tracked robot 48
4.2 Case of Tracks with Grousers: 49
4.2.1 Dynamic Analysis of the Robot in *Phase* 1 49
4.2.2 Dynamics of a Tracked Robot in *Phase* 2 52
4.2.3 Dynamic Analysis of the Tracked Robot in *Phases* 3 and 4 54
4.2.4 Dynamic Analysis of the Tracked Robot in *Phase* 5 56
4.2.5 Dynamic Analysis of the Tracked Robot in *Phase* 6 56
4.2.6 Dynamic Analysis of the Tracked Robot in *Phases* 7 and 8 57
4.2.7 Motion of the Tracked Robot in *Phase* 9 58
4.2.8 Dynamic Analysis of the Tracked Robot in *Phase* 10 60
4.2.9 Dynamic Analysis of the Tracked Robot in *Phase* 11 62
4.2.10 Dynamic Analysis of the Tracked Robot in *Phase* 12 63
4.2.11 Dynamic Analysis of the Tracked Robot in *Phase* 13 64
4.2.12 Dynamic Analysis of the Tracked Robot in *Phase* 14 64
4.2.13 Dynamic Analysis of the Tracked Robot in *Phase* 15 65
4.2.14 Dynamic Analysis of the Tracked Robot in *Phase* 16 65
4.2.15 Dynamic Analysis of the Tracked Robot in *Phase* 17 65
4.2.16 Dynamic Analysis of the Tracked Robot in *Phase* 18 66
4.3 Case of Tracks with no Grousers (Flat Track) 68
Chapter 5 74
5.1 Simulation 74
5.1.1 Introduction 74
5.1.2 Voltage–angular velocity line of the brushed DC motor 75
5.1.3 Speed-torque line 75
5.1.4 Acceleration Time 76

5.1.5 Angular velocity and Angular acceleration 77
5.1.6 Simulation Flow Charts 77
5.1.7 Simulation analysis conditions 80
5.1.8 Simulation analysis of the tracked robot during *Phase* 1 81
5.1.9 Simulation analysis of the tracked robot during *Phase* 2 82
5.1.10 Simulation analysis of the tracked robot during *Phase* 3 84
5.1.11 Simulation analysis of the tracked robot during *Phase* 4 85
5.1.12 Simulation analysis of the tracked robot during *Phase* 5 85
5.1.13 Simulation analysis of the tracked robot during *Phase* 6 86
5.1.14 Simulation analysis of the Tracked Robot during *Phases* 7and 8 87
5.1.15 Simulation analysis of the tracked robot during *Phase* 9 88
5.1.16 Simulation analysis of the tracked robot during *Phase*10 89
5.1.17 Simulation analysis of the tracked robot during *Phase*11 90
5.1.18 Simulation analysis of the tracked robot during *Phase* 12 91
5.1.19 Simulation analysis of the tracked robot during *Phase* 13 92
5.1.20 Simulation analysis of the tracked robot during *Phases* 14 and 15 92
5.1.21 Simulation analysis of the tracked robot during *Phase* 16 92
5.1.22 Simulation analysis of tracked robot during *Phase* 17 92
5.1.23 Simulation analysis of tracked robot during *Phase* 18 93
5.1.24 Comparing the Motion of the Tracked Robot with Grousers VS Flat Belt 95
Chapter 6 98
6.1 Conclusion 98
6.1.1 Conclusion and Recommendation for Future Work 98
6.1.2 Recommendations for Future work 102
References 104

List of Figures

Figure 1.1: A sample of robots network [11]……..3

Figure 2.1: Legged robot examples. I.: One leg hopper [15], II: Bipedal model [17], III: Four leg model (Big Dog) [18], IV: Six leg model [20]……10

Figure 2.2: Two models of wheel robots. Left: Powerboat mobile robot used for experimentation [21]. Right: WMRs, one of the wheel robot mechanisms for climbing stairs [22]……..12

Figure 2.3: Three-wheel robot while traversing stairs …. … ..…….....13

Figure 2.4: Star-wheel with r_{min} [24]…….....14

Figure 2.5: The stairs rise, Γ, the tread, B, slope of stair, γ, and distance between two noses, $A_{\Gamma;B}$……...15

Figure 2.6: Robot tracks; without and with grouser……..15

Figure 2.7: Top Left: Variable configuration robot [37]. Top Right: Non Variable configuration robot (Foster-Miller Talon Robot4) [35]. Bottom Left: Variable configuration tracked robot [36]. Bottom Right: Non variable configuration tracked robot, the MATILDA tracked [28]…….....16

Figure 2.8: Advantage of the front angled variable configuration tracked robot compared to the non-variable configuration tracked robot with circular front for traversing an obstacle [30]……..18

Figure 2.9: Climbing the wall and repulsive forces [31]..……19

Figure 2.10: Designing an angle of attack for overcoming the repulsive forces……..20

Figure 2.11: Descending of tracked robot as its center of mass crosses the pivot point [33]……..20

Figure 2.12 Tracked robot descending a stair [33]……21

Figure 2.13: Static analysis of robot on the stairs [33]…….....22

Figure 2.14: Skid steering turning on a stationary spot [28]……..23

Figure 2.15: A): Track robot without grouser. B): Track with small pitch grouser. C): Track with large pitch grouser [32]……...23

Figure 3.1: A track with circular front contacting a step. 28

Figure 3.2: Coefficient of friction versus the necessary radius of the circular track front…….. ..... …30

Figure 3.3: A Tracked robot as it starts climbing the first stair. 32

Figure 3.4: The tracked robot while climbing the first stair. 32

Figure 3.5: The geometry of tracked robot as well as the variable length of *q*. 33

Figure 3.6: Relation between the coefficient of friction and different angles of attacks as the robot progressively climbs the step..... 33

Figure 3.7: Robot pose as it starts climbing the first step. 35

Figure 3.8: Robot's pose during the climb of the first step. For static analysis, the angular acceleration, Ω, is zero and the torque, *T*, is not considered..... 35

Figure 3.9: Grouser interaction with step. 36

Figure 3.10: Relation between coefficient of friction and angles of attack as a robot with grousers climbs the first step. 36

Figure 3.11: Motion of tneTracked Robot during climbing in different phases(Climbing)…….. ….. 41

Figure 3.12: Motion of tneTracked Robot during climbing in different phases(Descending and on level surface) 43

Figure 3.13: The necessary location of the center of mass..... 44

Figure 3.14: Track grouser configuration. 45

Figure 3.15: Track/step geometry during yaw motion. 46

Figure 4.2.1: Pertinent kinematic and geometry relations of the robot in *Phase*1. 51

Figure 4.2.2: Robots pose, as it pass the first step..... 53

Figure 4.2.3: Pertinent kinematic and geometry relations of the robot in *Phase* 2. 54

Figure 4.2.4: Robot poses in *Phase* 3 while the track bottom is in contact with two step noses…….. 55

Figure 4.2.5: Robot poses in *Phase* 4 while the track bottom is in contact with three step noses……. 55

Figure 4.2.6: Robot poses when the robots' center of mass crosses the pivot point at the top step……..... 57

Figure 4.2.7: Schematic of the robot poses in *Phases* 7 and 8. ..... .58

Figure 4.2.8: Robot poses during the descending in *Phase* 9..... 59

Figure 4.2.9: Robot poses during the descending in *Phase* 9..... 60

Figure 4.2.10: Robot poses, while it tips over first step before reaching the second nose (*Phase*10)..... 61

Figure 4.2.11: Robot poses, during the descending in (Tracked Robot has two contact with stairs noses). 62

Figure 4.2.12: Robot poses, while it tips over the step and before reaching the following nose (*Phase*12). …….63

Figure 4.2.13: Robot poses, during the descending in (Tracked Robot has three contact with stairs noses). 64

Figure 4.2.14: Motion of the tracked robot in Phase17 66

Figure 4.2.15: Motion of the tracked robot in *Phase*18..... 67

Figure5.1.1: DC Motor Angular Speed, (ω), Motor Voltage and Voltage Constant (K_v)…….. ..... .75

Figure 5.1.2: The relation between the DC motor's maximum angular velocity and its maximum torque (T_{stall})..... 76

Figure 5.1.3: The time that the angular velocity reaches its maximum is called Acceleration Time [1].……..76

Figure 5.1.4: The relation between angular acceleration and angular velocity [1] 77

Figure 5.1.5: Summarized Robots simulation charts..... 78

Figure 5.1.6: Tracked Robots simulation charts (Detail) 79

Figure 5.1.7: Simulation of the Tracked Robot during *Phase* 1 81

Figure 5.1.8: Simulation of the Tracked Robot during *Phase* 2 84

Figure 5.1.9: Simulation of the Tracked Robot during *Phase* 3 85

Figure 5.1.10: Simulation of the Tracked Robot during *Phase* 6 86

Figure 5.1.11: Simulation of the Tracked Robot during *Phases* 7and 8..... 87

Figure 5.1.12: Simulation of the Tracked Robot during *Phase* 9 88

Figure 5.1.13: Simulation of the Tracked Robot during *Phases* 10 90

Figure 5.1.14: Simulation of the Tracked Robot during *Phase* 11 91

Figure 5.1.15: Simulation of the Tracked Robot during *Phases* 17 93

Figure 5.1.16: Simulation of the Tracked Robot during *Phases* 18 94

Figure 5.1.17: Top _Simulation of the 18 *Phases* of the Tracked Robot with grousers (Part1)……. 96

Figure 5.1.18: Top _Simulation of the 18 *Phases* of the Tracked Robot with grousers (Part2)…… 97

List of Tables

Table2.1: Comparison of Different Locomotion Concepts……..25

Table 3.1: Staircase specifications as per the building code [13]……..27

Table 4.1 : The illustration of the differences between the equations of Tracked Robot with Grousers (TRG) and the Tracked Robot with Flat Belt (TRFB) during 18 phase……69

Table 5.1: listing of the parameters values used in the simulation…….....80

Nomenclature

$A_{\Gamma,B}$	The distance between two step's noses.
$(a_{cm})_x$	Acceleration of center of mass in X direction
$(a_{cm})_y$	Acceleration of center of mass in Y direction
$(a_c)_x$	Acceleration of Robot's front contact point with step in X direction.
$(a_c)_y$	Acceleration of Robot's front contact point with the step in Y direction.
a_A	Acceleration of rear of tracked robot.
B	The stairs tread
C_m	Center of mass.
E_{ff}	The gear box efficiency
F_x	Forces in X direction.
F_y	Forces in Y direction
F_A	Tracked robots rear tangential reaction force.
F_C	Tracked robot's front tangential reaction force.
F_M	Provided force by Gear box
G	The gear box ratio
h	Height of center of mass with respect to flat bottom part of tracked robot.
h_1	Height of grousers

H_1 The grousers root thickness

k_V Voltage constant.

L The minimum length of the track robot's bottom

L_c The distance of center of mass to center of front bottom wheel

L_A The distance of center of mass to center of rear wheel

$L_1^{`}$ The projected length between rear contact point and center of mass on to the level ground.

$L_2^{`}$ The projected length between front contact point and center of mass on to the level ground.

M_C The applied moment at point *C*

M_{Cm} The applied moment at center of mass.

mg Robots Weight.

N_A Tracked robot's rear normal reaction force.

N_C Tracked robot's front normal reaction force.

P Pitch

P_T Power of DC motor

q Projection of the length between the nose of the step and the beginning of the angled of the track onto the ground.

q_2 Projected distance from the front of the bottom of the track to the nose of the first step.

R_{r1} The distance from the robot's center of mass to the step contact point (*Phase*1)**.**

R_{r2} The distance from the robot's center of mass to the step contact point (*Phase*2)**.**

R_{r3} The distance from the robot's center of mass to the step contact point (*Phase*17).

r The radius of the driven pulley.

$r_{cm/A}$ The distance between center of mass and rear Robot.

T Torque.

T_{stall} Maximum torque of DC motor.

t_a The acceleration time

W_1 The width of the track

Γ The stairs rise

β The angle between level ground and the bottom of track robot

$\ddot{\beta}$ Angular acceleration of β and robots center of mass

Θ Angle of attack

Φ Yaw angle

Ψ Angle of grousers

μ Coefficient of friction.

Ω The angular acceleration of the tracked robot about its center of mass.

γ Constant slope of Stairs.

ϖ Angular velocity of Robot's center of mass

ω Motor Angular Speed,

Chapter 1

1.1 Motivation

Humans have dreamed of mechanical devices to ease their chores and assist them in performing repetitive tasks for thousands of years. These dreams remained primarily theoretical for most of the past centuries. Early devices were usually novelty items, such as the automated rooster erected in Strasbourg, France in the mid-14th century [1]. By the end of the 18th century, robot-like devices were able to perform a few minor useful chores. Examples of these types of devices include the Automatic Scribe, which inventors Pierre and Henri-Louis Jacquet-Droz showed in 1774 [1]. This device was able to write a message up to 40 characters long. These same inventors also created a piano playing robot. These types of machines hinted at what machines might be able to do, but were impractical for anything other than entertainment.

Early labor-saving devices were hobbled by the lack of available power. Most labor required human or animal muscle power. Other devices, such as water mills, required either a fast moving stream, or a location with reliable wind power.

The development of the steam engine, starting in the early 18th century [2], accelerated the development of machines that could speed up and automate repetitive tasks. Machinery improved, and during the 19th century, devices such as the textile machine [1], the mechanical reaper [1], and the railroad revolutionized textile manufacturing, food production, transportation, and other manufacturing industries.

Autonomous machines that could perform repetitive manufacturing steps with minimal human intervention began to be developed and used in the mid-20th century. Devices that have a significant degree of autonomy to perform their programmed tasks are labeled as "robots". This word was first used by the Czech writer Karel Capek (1890-1938) in his 1921 play R.U.R. (Rossum's Universal Robots). The word comes from the Czech "robota," which means serf or one in subservient labor [1], [3].

Mobile robot development also began to accelerate starting in the mid-20th century. An early mobile robot was the Westinghouse Electro, an over 7 feet tall robot that had a limited ability to move and talk when given the appropriate commands. This robot was first displayed to the public at the 1939 World's Fair in New York [1], [3]. The first autonomous robots were developed by Grey Walter and his team in the late 1940s to early 1950s in England [1], [3-6],. The first of these robots (named Elmer and Elsie) were designed to move towards a light source, and could crudely maneuver around obstacles, without outside intervention. These robots used two vacuum tubes in their control circuitry that gave them a limited amount of autonomy. From these beginnings, mobile robot development has produced many robots that perform useful tasks, such as dismantling bombs, debris cleanup in radioactive locations, and space exploration [1].

Another important application of mobile robots is to guard both public and secure locations. This has become a very important field of research in recent years. Many companies, large and small, have been developing and marketing robots that can perform guard and patrol duties. Examples of these robots include the Samsung Techwin SGR-A1 Sentry Guard Robot [7] and the models available from iRobot Corporation [8].

Security robots are the broad category of robots that are addressed in this thesis. It is important to first define the jobs required of security robots. There are several broad applications that such robots could be used in. These robots could be deployed to perform site surveillance prior to sending humans into potentially dangerous situations. More sophisticated robots could completely eliminate the need to send people into hazardous locations by performing the work that a human would otherwise need to do. By adding image processing capabilities, a robot could recognize critical situations that a human might miss. Finally, there are potential long-term cost savings, where an initial investment in a set of expensive robots can reduce the number of people that would otherwise be required to perform the same duties.

Examples of security robots include nuclear power plant inspection robots, such as the Siemens SAPHIR system, that can cut inspection times in half [9]. A more recent example is the use of Oceaneering International's Millennium deep sea robots that were used to monitor and help seal the 2010 Deepwater Horizon oil spill in the Gulf of Mexico [10]. This work could only be performed by robots, due to the 5000 feet depth of the spill. Closer to home, there is a large effort being made to design robots that can patrol public buildings, such as malls, and allow operators to assess potentially dangerous situations from a safer distance [11].

Most of the current patrol robots are remotely controlled by humans, rather than being able to operate autonomously. This simplifies their design, but requires that at least one human be employed to monitor and control their operation. Figure 1.1 shows a wireless network control system that is being developed in Japan for a mall surveillance system. This system has a limited degree of autonomous operation, but is controlled by human monitors via the wireless network, which also sends the video collected by the robots back to the monitoring station. Remotely controlled robots require a robust communications path between the robot and the operator. This can be an important design constraint.

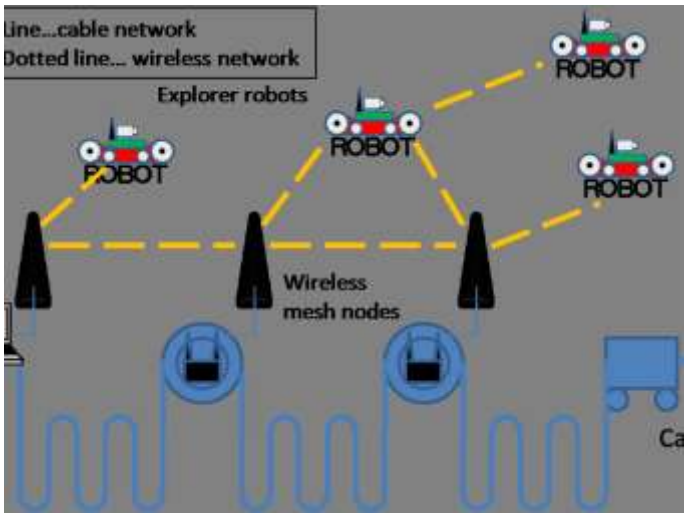


Figure 1.1 : Illustrates a sample of robots network [11]

Autonomous robots are being developed, but are not very advanced at present. Autonomous operation requires that the internal circuitry of the robot has enough built-in intelligence for the robot to accomplish its mission without any external control once the mission is started. These types of robots are much more difficult to design than robots that are tethered to

a human operator when they are in operation. One example of an autonomous system in current use is the Monterey Bay Aquarium Research Institute's autonomous underwater vehicle (AUV) [12]. These machines are programmed on board a ship, and then sent underwater to autonomously collect data before being recovered at the end of their mission.

The usefulness of mobile robots has made this field a very active area of research. Many research groups around the world are studying and developing these robots. A survey of these efforts is provided in Chapter 2. Mobile robots can be categorized in many different ways, including their field of application, the type of control system, or the method by which the robot moves from one location to another. For the purposes of this thesis, robots are primarily grouped by their method of locomotion. This is an important aspect of robot design. The locomotion method sets important limits on the type of terrain a robot can traverse, the required size of the robot for a given application, its operating range, and other important aspects. Robot locomotion methods can be broadly slotted into three main groups: walking, wheeled, and tracked robots.

The research presented in this thesis is oriented for use in robots that can perform security duties in a building or a mall. Robots that are used in building inspection or guard duty must be capable of climbing any stairs that are inside their building. The main emphasis of the research reported in this thesis was to find the best robot locomotion method for climbing stairs. An important step in this process was to specify the stair parameters, which vary around the world. The most important parameters are the stair height and width. These two variables set the stair angle, which is another important variable. A stair climbing robot must be capable of climbing and descending the stairs in its deployment location efficiently and with minimal or no slippage. If the location contains more than one set of stairs, these stairs may have different heights and/or widths, and the robot must be able to climb all sets of stairs. For this project, the North American building standards, including those for flat surfaces and stairs, were used as design guidelines.

An important step in this process was the selection of the best type of robot for this application, from among the three groups. This thesis will show that the tracked robot is usually the one best suited for stair climbing. After this selection had been made, the mechanical behavior of the robot is analyzed in order to design the best robot for the chosen application. A prototype robot was then designed, constructed, and tested to confirm the theoretical analysis.

This thesis examines several design and implementation concerns for climbing stairs, many of which have not been extensively covered in the published literature. A survey of the available literature reveals that most robot studies used static analysis for the design of these devices. Dynamic analysis of climbing and descending obstacles was seldom considered or reported on.

Many important analysis and design problems were studied as part of this project. This includes analyzing the tracked robot dynamics in climbing stairs, as well as turning on flat surfaces. The efficiency difference between tracked robots with grousers and ones without flat tracks when climbing stairs is considered. This includes their energy consumption and the types of DC motors required for this task. Determining the torque required for climbing the first step in both cases (with flat belt and with attached grips on flat belt) is also important; to make sure the selected motor can supply the needed amount.

Other important aspects of this research included studying the best track angle of attack to reduce the repulsive force of the first stair step, and analyzing the needed minimum length of stair contact for climbing. The relation of the grouser pitch to the maximum yaw angle while the tracked robot is climbing stairs was examined. The minimum dimensions of a tracked robot were also studied. The performances of the two types of robot tracks (grousered and smooth) are compared in Chapters 4 and 5, using dynamic simulation studies. The approach taken in this thesis results in a set of guidelines that can be used to find an optimal design for a tracked robot that is stable for climbing and descending stairs, and efficiently uses its available energy source.

Building or mall security robots must also be very maneuverable, be as compact as possible, and be directed by a central control system. Many of the design constraints conflict with each other. This is usually the case for engineering designs, and decisions must be made to select the best compromises consistent with creating a robot that can perform its expected duties.

The remainder of this thesis is devoted to the systematic development of a stair climbing robot. It starts with a survey of the existing literature to determine the current state of the art and to help select the best robot structure for the stated purpose. The robot is theoretically analyzed using both static and dynamic methods. The computer simulations are performed and reported. This is followed by the mechanical design of the selected robot type, including discussions of the

selections of the major mechanical components. The total design and analysis process is then summarized, and future research aspects are briefly discussed in conclusion.

1.2 Chapters Organization

Chapter 2:

Before embarking on a design of a machine, it is important to investigate similar previous designs so as to avoid reinventing the wheel. Consequently, reviewing the existing literature is an important early design step. There are many articles, theses, and books about tracked robot motion in different situations. These were very useful for analyzing and designing the robot described in this thesis. Chapter 2 reviews the articles that were deemed most relevant to the present thesis.

Chapter 3:

The shape and dimensions of the robot's tracks have a great impact on the efficiency of the robot. This thesis is primarily concerned with the case where the robot is climbing staircases where the steps are periodic with a uniform pattern.

This chapter compares the differences between the wheel sizes of different robots for climbing stairs, and studies the angled front of the tracked robot

The different phases of the tracked robot as it climbs and descends stairs are also analyzed. The location of the robot's center of mass for a successful ascending and descending stairs is another important consideration that was studied. . Furthermore, the efficiency of flat tracked robots and tracked robots with grousers is discussed.

The last part of this chapter deals with designing the tracks and the grouser pitch for efficient stair climbing.

Chapter 4:

Chapter 4 analyzes the dynamics of a tracked robot's motion during climbing and descending stairs in its different phases and compares the equation of motion of the tracked robot with attached grips (grousers) versus one with smooth belts.

Chapter 5:

This chapter applies numerical analysis techniques to study the motion of tracked robots when climbing and descending stairs. It also compares the efficiency of robots with grousers versus smooth tracks.

Chapter 6:

Chapter 6 concluded a summary of the major contributions of this work to the field of mobile robots and thoughts for future work are also discussed.

Chapter 2

2.1 Literature Review

Since the time when the first mobile robot was designed and manufactured by General Electric in 1968, a great deal of effort has been dedicated to researching, designing, implementing, and refining different types of mobile robots.

The purpose of this chapter is to summarize the available literature related to the research efforts on mobile robots that can climb stairs. This includes the mechanical advantages and disadvantages of special designs and the motivation extracted from different articles. The focus of this summery is on the mechanical aspects of robots, including their static and dynamic analysis as well as their mechanical design.

Generally, mobile robots can be broadly grouped into three categories, based on their means of locomotion and stair climbing methods. The first group comprises walking machines or legged robots. Their development is inspired by the propulsion methods of most land animals. The second category, and by far the most common, comprises wheeled robots. These machines may have as little as two wheels but can have eight or more wheels arranged on several axles. One or more of the axles may be powered to provide locomotion. The third category contains mobile robots that move by tracks or belts. Tracked robots are a modification of wheeled robots, with sets of wheels that drive at least two belts or tracks, which in turn make contact with the ground to propel the robot in the desired direction.

2.2 Walking Machines (Leg Mechanisms)

There are many types of legged robots. The minimum number of legs can be as low as one, as in the case of the one legged hopper developed by Marc Raibert at MIT in 1983 [13]. However, two-legged (Biped Model) [14], four-legged [15] and six-legged [16] robots are the most popular configurations. As the number of legs increases, the robot stability also increases, but so does the mechanical complexity. The advantages and disadvantages of legged machines are highlighted below.

2.2.1 Walking machines advantages

Walking machines are eminently adept at traversing irregular surfaces as well as ones where the surface quality and traction are low, such as bogs and slippery surfaces. This walking ability is enhanced if the machines have computer-controlled variable step height. These robots can navigate rough terrain and cross large, dispersed obstacles [14]. Because of their small contact surface with the ground, they produce considerably less damage to the ground [15]. Their motion could be roughly similar to that of humans or some animals, if they are so designed.

2.2.2 Walking machines disadvantages

The disadvantages of walking machines are discussed in increasing order of the number of legs the robot possesses. The main disadvantage of the one legged robot is its instability. These robots are inherently unstable [19]. The robot has to be in continuous hopping action in order to stay upright.

The static and dynamic stability of the two legged robot is one of its major problems. During walking, balancing the center of gravity between the different parts of the robot is a difficult challenge. The robot must continuously adjust its center of mass in two planes simultaneously as the legs change their configurations to take a step.

The four legged robot is by far the most popular type. It has very good static stability because of its four points of contact. The dynamic stability of this robot is also good because three legs are always in contact with the ground when it is walking. These robots require a complicated

system of control to synchronize the stepping action of the four legs. Six legged robots have the same complexity problems as their four legged counterparts, but depending on their construction and control mechanism, they may have better stability.

The main disadvantage of walking machines is their increased mechanical complexity compared to the other robot types. Walking machines need to synchronize all leg movements by means of mechanical or electromechanical mechanisms. In general, the movements of the legs have to be controlled by either a complex computer program or an elaborate cam system. These systems must also be able to steer the robot, and perhaps maneuver the robot around obstacles.

The complex leg mechanisms results in greater energy consumption when compared to that of wheeled or tracked robots. In addition to the inefficiency of the engine and transmission, it is necessary to power a large number of leg joints. Much of this power is wasted in stopping and starting the reciprocating movement. Walking machines also require more space for actuators, which reduces the available space for batteries, cargo, and devices that are needed for the robot to accomplish its intended mission. The increased power consumption and smaller battery capacity of legged robots reduces their working time. The complexity of the mechanical actuators and electronics systems make the walking robots expensive compared to wheeled and tracked robots.

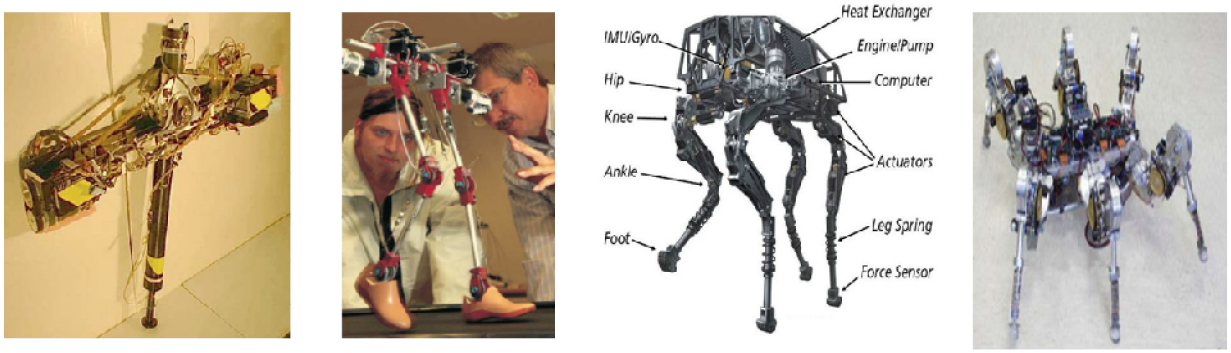


Figure 2.1 : Legged robot examples. I.: One leg hopper [15], II: Bipedal model [17], III: Four leg model (Big Dog) [18], IV: Six leg model [20]

2.3 Wheeled Robots

Wheeled robots are by far the most common type of all mobile robots, owing to the efficiency of this method of propulsion. The number of wheels used can be as few as two and can be increased as required to suit the operating conditions and the environment. However, the most common number of wheels are three and four. Based on the literature reviewed, the characteristics, advantages and disadvantages are described below.

2.3.1 Wheeled machines Advantages

Wheeled robots can typically navigate much easier and faster than robots equipped with other types of propulsion. Because of the relatively small contact area between the wheels and a flat surface, wheeled robots consume less energy for turning compared to other types of robots [21]. Furthermore, they are generally lighter than tracked or walking robots because of the simplicity of their locomotion mechanism. This simplicity, and the subsequent reduced number of actuators also results in less audible noise during operation [22].

Wheeled robots can be designed to any desired shape and size. With a slight change of design, it is possible to increase the ground contact area by adding more wheels to the robot to increase its traction force. This can reduce the possibility of slippage in certain situations. One example is the Wheeled Mobile Robot (WMRs) [22], which is designed for rough terrain and for climbing stairs. A schematic of this robot is shown in Figure 2.2 Right. The cost of design and manufacturing a wheel robot is usually less than that of legged and tracked robots because of the simpler mechanisms and control devices needed.

2.3.2 Wheeled machines Disadvantages

The wheeled robot's small contact area with the ground gives it an energy consumption advantage, but this comes with a propulsion shortcoming. The small contact area makes it difficult for wheeled robots to move on soft or mushy ground like bogs or on slippery surfaces.

Climbing over obstacles greater than the wheel radius is impossible for standard wheeled robots. The wheel-ground area of contact and the energy requirement for crossing obstacles is a major subject of study on these robots. The stair climbing difficulty is the biggest disadvantage of wheel robots from the standpoint of the goals of this research.

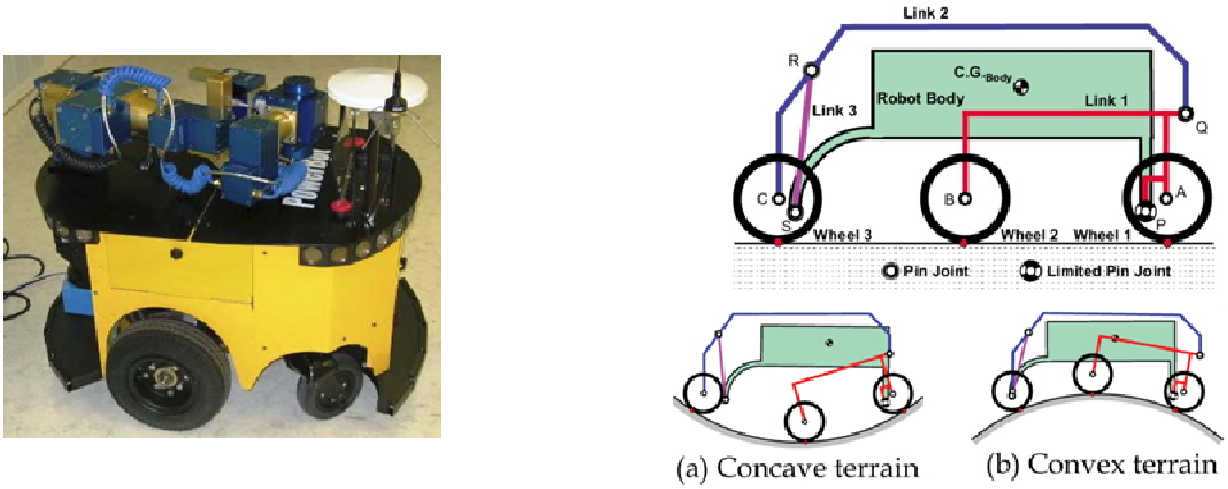


Figure 2.2 : Two models of wheel robots. Left: Powerboat mobile robot used for experimentation [21]. Right: WMRs, one of the wheel robot mechanisms for climbing stairs [22].

2.4 Three-wheel and Star-wheel (Walking and rolling machines)

The three-wheel robot concept was first used in a stair climbing wheelchair in 1962. This machine was designed by R.B. McLaughlin, at the US Department of Commerce National Inventors Council in Washington, DC. Since then, there have been many development efforts for using this type of design [23].

The basic configuration of this propulsion system is that of three wheels attached to the corners of a triangle. A motor or internal combustion engine drives a central gear, as shown in Figure 2.3; this gear in turn drives the wheels by means of six other gears. The literature dealing with this unique propulsion system is reviewed below.

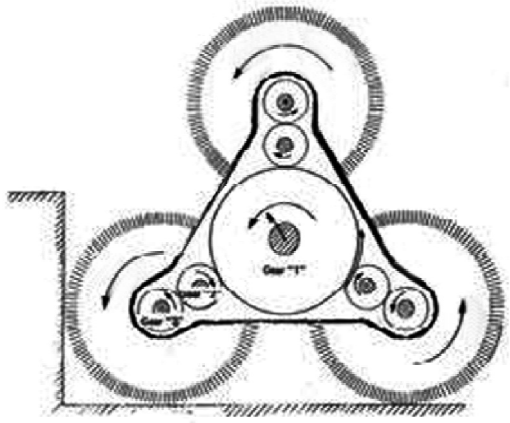


Figure 2.3 : Three-wheel robot while traversing stairs [23].

2.5 Star-wheel (MSRox)

The Star-wheel mechanism is similar to the three-wheel system discussed above. Star-wheel (MSRox) robots are designed for industries and power plants [24]. The wheel dimensions are designed to enable the robot to climb stairs, as shown in Figure 2.4.

As the robot contacts the rise of the stair, the bottom wheel behaves as a lever for the top wheel. The top wheel turns toward the flat part of the stair. If the distance between the centers of two wheels is more than the rise of the stair, the top wheel will reach the flat part of the next stair. As shown in Figure 2.4, the minimum value of the radius of regular wheel, r_{min}, to prevent the collision of the holder to the stairs is derived as follows [24]:

$$r_{min} = \frac{6Rt \mid a(3b \quad \sqrt{3a})}{(3-\sqrt{3})a + (3+\sqrt{3})b} \tag{2.1}$$

$$R = \sqrt{a^2 + b^2} \tag{2.2}$$

Where R is the radius of the star-wheels arm and is given by:

a is the height of a stair step

b is the landing width of a stair step

t is half of the thickness of the holders

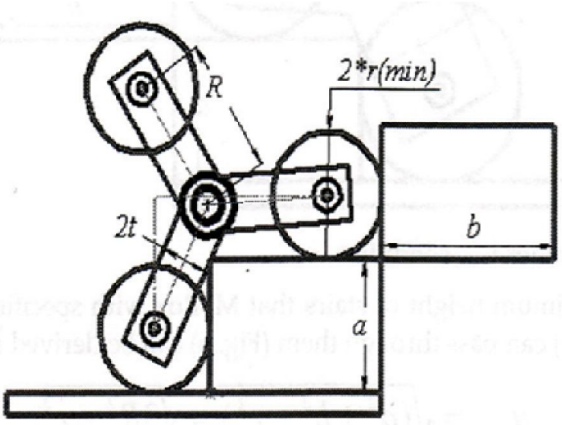


Figure 2.4 : Star-wheel with r_{min} [24].

2.5.1 Advantages

The three-wheel and Star-wheel mobile robots has all of the advantages of a wheeled robot, yet they have the ability to traverse large obstacles. In addition, they have the advantage of being able to climb stairs, if the stairs conform to the limits of the robot design.

2.5.2 Disadvantages

According to “Standards of stairs and Doors” [25], the height, width, and staircase angle can all vary for different sets of stairs. The constant angle of the star-wheel arms relative to each other, as well as their constant length, can cause problems for climbing and descending stairs.

When the robot is designed for a specific staircase dimension, the number of rotations of the star-wheel matches the number of stair steps. A particular robot can operate within a certain limit of staircase dimensions. Outside that limit, the robot is unable to climb the stairs. Table A.1 in Appendix shows the variations in tread and the risers of the steps with respect to the slope of the stairs.

When descending stairs, the robot may experience slipping, followed by smooth motion until the next slippage occurs [26]. This depends on the robot's wheel and star-wheel sizes relative to the staircase. The configuration of this propulsion mechanism results in a robot size that is larger than its simple wheel counterpart, for a given payload.

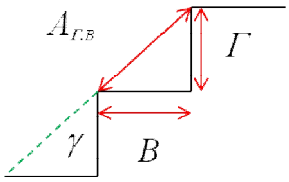


Figure 2.5 : The stairs rise, Γ, , the tread, B, slope of stair, γ,, and distance between two noses, $A_{\Gamma,B}$.

2.6 Tracked robots

Tracked robots are a modification of wheeled robots, where sets of wheels drive at least two belts or tracks, which in turn make contact with the ground and propel the robot in the desired direction. The outside of the tracks may have grousers attached at periodic intervals, to reduce slippage of the track with respect to the ground. Figure 2.6 illustrates the track belt configurations, showing belts with and without grousers. This figure also defines the grouser pitch.

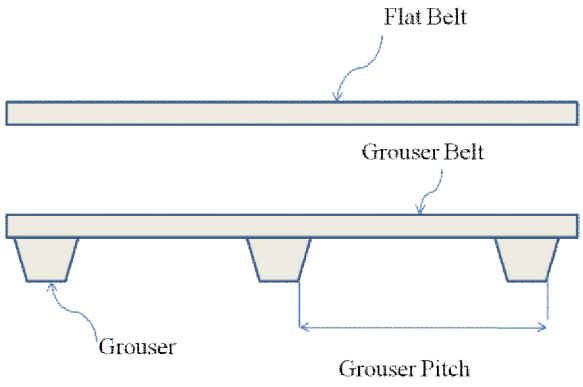


Figure 2.6 : Robot tracks; without and with grouser.

Tracked robots are currently used in many fields for diverse purposes. Examples include detecting and defusing bombs, expeditions, agriculture, protecting and guarding buildings by

means of cameras mounted on the robot body. There are two types of tracked robots: fixed geometry tracked robots and variable geometry tracked robots. Fixed geometry tracked robots have simple tracks with at least two driven sprockets and two flexible endless tracks that look like timing belts. The tracks transmit the forces from the driven sprockets to the ground. Variable geometry tracked robots have auxiliary sets of tracks that can be rotated about a fixed axis as needed to adjust their angle of contact with obstacles, angle of attack, to enhance the robot's ability to climb these obstacles. Figure 2.7 shows typical examples of two variable geometry tracked robots and two fixed geometry tracked robots.

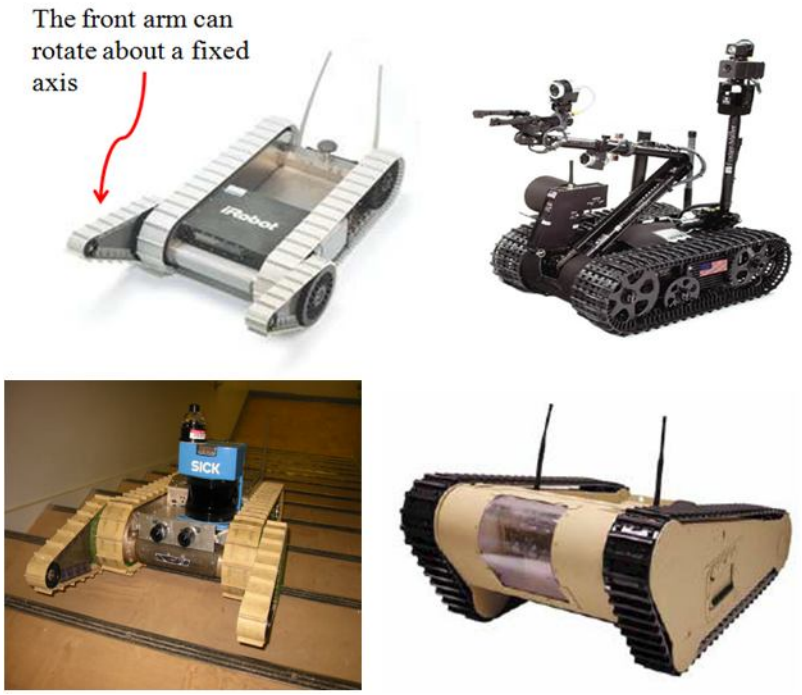


Figure 2.7 : Top Left: Variable configuration robot [37]. Top Right: Non Variable configuration robot (Foster-Miller Talon Robot4) [35]. Bottom Left: Variable configuration tracked robot [36]. Bottom Right: Non variable configuration tracked robot, the MATILDA tracked [28].

2.6.1 Tracked robot advantages

Tracked robots have many advantages over the wheel and leg types of stair climbing robots. They can navigate over very rough terrain as well as climb up and down steep staircases when appropriately designed for that purpose [29]. Tracked robots have a very large ground contact area. This provides them with increased traction and stability. This also allows them to travel over smooth and slippery surfaces [29]. Because of their propulsion mechanism, these robots use skid steering, whereby the two tracks are driven at different speeds (including different directions) to permit steering [29]. This method of steering allows the robot to have a very small turning circle. When the tracks are driven at the same speed but in different directions the robot can rotate about its center, i.e., a turning circle of zero radiuses.

2.6.2 Tracked robot disadvantages

The advantages of tracked robots are countered by some disadvantages. They are slower than wheeled robots on flat surfaces, and their energy consumption is much higher. Their mechanical complexity is greater than that of wheeled robots, but is usually less than that of walking robots. When navigating curves, tracked robots experience skidding that is accompanied by a large counter toque to the turning torque and results in a large power loss [27]. This counter turning torque also damages the ground surface and causes the tracks to wear.

2.7 Tracked Robot Analysis

Tracked robots represent a reasonable compromise between the simplicity of a wheeled robot and the complexity of a walking machine or a star-wheel machine in many applications. Their relative stability and ability to climb stairs have made them the robot of choice for a large set of situations. This thesis primarily studies tracked robot configurations. Both variable configurations (Variable joints, as shown in Figure 2.7, top left) and fixed geometry tracked robot configurations (such as those shown in Figure 2.7, top right) are analyzed, but with more emphasis on the fixed geometry ones. Fixed geometry tracked robots have a much simpler mechanism and are less expensive than their variable geometry counterparts. At the most basic

level, they commonly employ two DC motors (one for each track), whereas the variable geometry robots need at least four (the two additional ones drive the variable tracks). Variable geometry tracked robots thus must employ larger and heavier batteries in order to satisfy the higher energy demands. Non variable tracked robots have fewer degrees of freedom than variable configuration tracked robots. Therefore, their geometries must be precisely designed for the required applications.

Variable geometry tracked robots have also been the subject of many studies [29, 37]. They have several advantages over fixed geometry tracked robots, including having more clearance for crossing obstacles and rough terrain. As the terrain changes, the robot can maintain its balance by moving and controlling its arms (configurable tracks) to change its center of mass. Changing the center of mass is not only important for climbing obstacles, but is vital for descending.

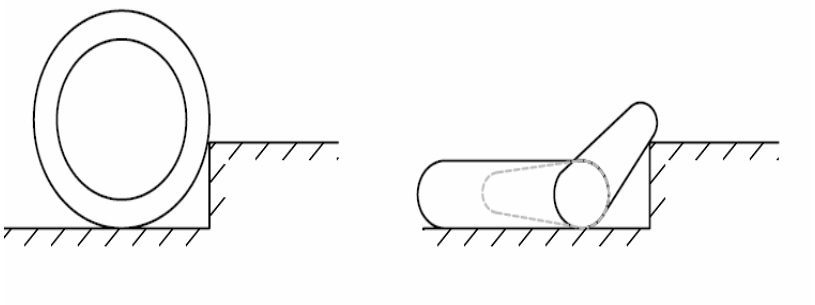


Figure 2.8 : Advantage of the front angled variable configuration tracked robot compared to the non-variable configuration tracked robot with circular front for traversing an obstacle [30].

The non-variable tracked robot's required wheel size for crossing obstacles theoretically must be two times larger than that of an equivalent variable track robot [30]. The tangential force applied by the wheel to the obstacle must have an angle of less than ninety degrees with respect to level ground.

2.8 Analysis of Fixed Geometry Tracked Robot when Climbing Stairs

This section introduces the analysis of the motion of the tracked robot while climbing stairs. When the robot encounters the stair riser, its motors apply torque to the tracks to start the robot climbing the stair riser. The most difficult part of climbing is when the angle between the robot's

center of mass and the riser, θ, is between zero and 30 degrees. As θ increases, both the coefficient of friction between the track and the ground and the driving moment increase. The robot's velocity and acceleration also increase. The best arrangement of mass distribution for optimum stair climbing is when the robot's center of mass is located close to its rear wheel [31].

As was pointed out at the end of Section 2.7, it is impossible to climb stairs with this configuration (Figure 2.9). Furthermore, finding the best location of the center of mass requires analyzing the different phases of the tracked robot during climbing and descending.

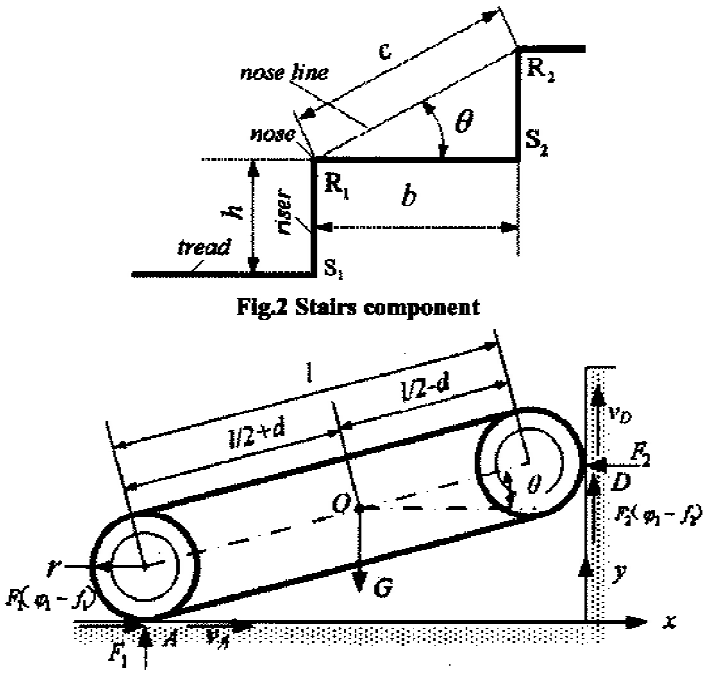


Figure 2.9 : Climbing the wall and repulsive forces [31]

The Repulsive forces applied to the robot from the stairs as the robot climbs the stairs makes climbing difficult. These forces can be reduced by configuring the track with a front angle of attack as shown in Figure2.10 [32]. This makes it easier for the robot to climb the obstacle, at the expense of a more complicated front end with more moving parts. The angle of attack is the angle between the front parts of the tracked robot with level ground. The height of the front part of the tracked robot must be taller than the obstacle.

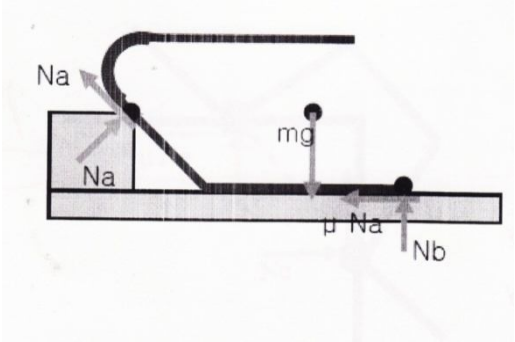


Figure 2.10 : designing an angle of attack for overcoming the repulsive forces [32].

One of the most challenging subjects of study on mobile robots is the analysis of descending from objects. Figure 2.11 illustrates the situation of a robot descending from a single obstacle. When a tracked robot is in forward motion and encounters the topmost step of a staircase to begin its descent, the robot keeps traveling until its center of gravity extends past the nose of the step (Point O). At this time, a rotational torque begins to develop.

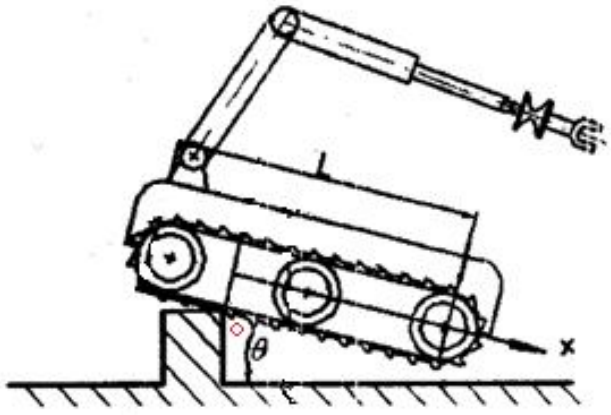


Figure 2.11 : Descending of tracked robot as its center of mass crosses the pivot point [33].

While the robot continues moving forward, the magnitude of the torque increases. This causes the angular acceleration of the robot to rapidly increase about its (moving) pivot point. The combination of the linear downhill momentum and the additional angular momentum results in a harder collision when the front of the robot hits the ground or another stair step. As shown in Figure 2.12, under certain conditions, the pivot point can shift from the first step nose to a lower

stair tread or nose (Point A of the third step in this example) [33]. If the rotational momentum is large enough, the robot could topple over.

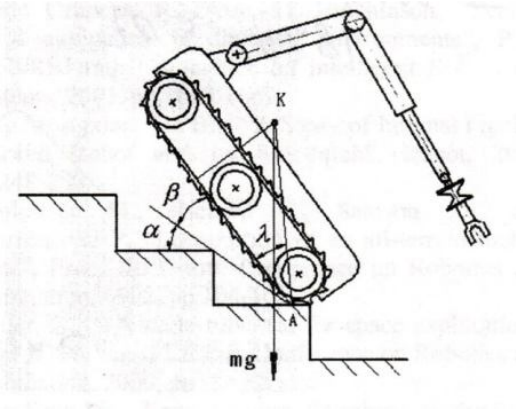


Figure 2.12 : Tracked robot descending a stair [33].

2.9 Climbing Stability Analysis

Several studies have examined the stability of tracked robots when climbing or descending obstacles. As the robot climbs the steps, it does not climb straight up the stairs, but has a deviation in the horizontal direction. This is caused by a number of factors. Among these factors are the belts' elasticity, the grouser contact points with the steps, and the variations in the coefficients of friction [34]. The belts act as dampers when the tracked robot first collides with the steps, and both belts at the points of contact cannot have the exact same elasticity. The locations of the right grousers' contact points with the steps will be slightly different than those of the left ones. These two factors cause the robot to start climbing with a horizontal angle. The tracked robot thus starts to steer and slide to the left and then to the right (or vice versa). This motion induces changes in the area of contact between the tracks and the edges of the stairs, which, coupled with changes in the coefficients of friction, also cause deviations from the horizontal. Therefore, there will be an angle of deflection, or yaw angle, between the tracks and the ideal straight line [34].

Figure 2.13 illustrates the situation. If φ is the angle of deflection, α is the angle of slope, and κ is the center of mass, it can be shown that the stability of the robot depends on φ and α. As the perpendicular to ground, *KP*, approaches *KI* or *BC*, the possibility of roll over increases [33]. The critical case is where:

$$\alpha + \gamma = 90 \tag{2.3}$$

Haijun Mo, Ping Huang and Showei Wu [33] derived an equation based on the geometrical analysis of the robot climbing stability:

$$cos\ \varphi sin\ \alpha = (L - L_1)H \tag{2.4}$$

Where: L is the length of track contact with ground, H is the height of the center of mass from the base of the track, and L_1 is the level distance of the point K to line CD.

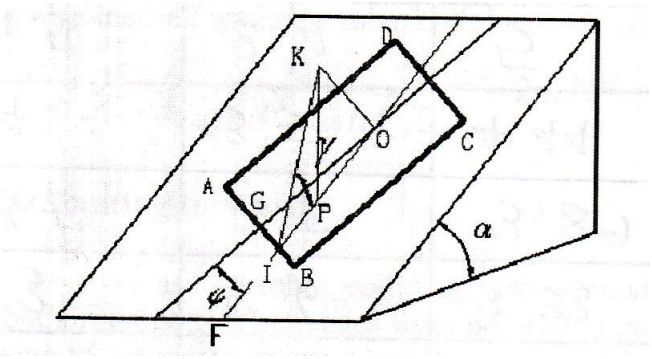


Figure 2.13 : Static analysis of robot on the stairs [33].

2.10 Tracked Robot Turning Analysis

The area of contact between the tracks and the ground is much larger than that of the wheels of an equivalent wheeled robot. Hence, with reference to Figure 2.14, when skid steering is used for turning, tracked robots require more torque than that required for a wheeled robot. The difference in speed between the two tracks imposes a very large torque on the slower track. It is estimated that the energy required by a tracked robot for turning on a flat surface is greater than that required at the start of climbing a 40° degree slope with a 0.8 coefficient of friction [27].

Figure 2.14 : Skid steering turning on a stationary spot [28]. H) Track segment movement direction. I) Relative friction forces between track and ground[28].

2.11 Other considerations

Other design subjects that must be considered are the tracks, timing belts, and, especially, grousers [32]. The grousers can significantly reduce the track robot's energy consumption in all situations. They can act as a grip for climbing obstacles, as shown in Figure 2.15 (center and right). Only one article was found in the literature that analyzes the effects of grousers, and it was written in the Korean language [32]. The authors of this article, Park Nam-Eun, Park Dong-II and Kwak Yoon-Keeum analyzed the forces acting on the grousers while the tracked robot climbs stairs. They also analyzed the effects of the grousers' pitch. They used static rather than dynamic analysis, which is the main disadvantage of their research.

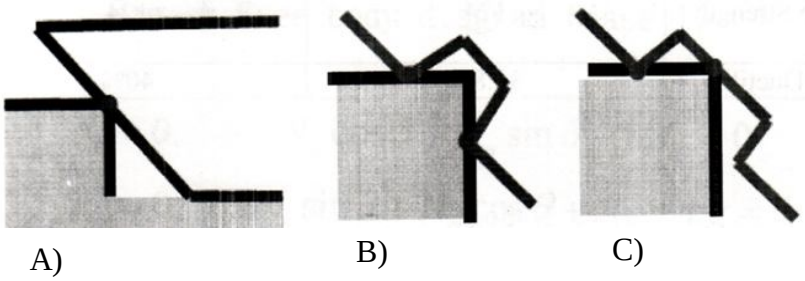


Figure 2.15 : A): Track robot without grouser. B): Track with small pitch grouser. C): Track with large pitch grouser [32].

The grousers' pitch can play a very important role in the efficiency of track robots. A very small or very large pitch does not have the best efficiency. A small pitch cannot act as a

grip and the track can slip. A large pitch can have slippage until the grouser reaches one of the stair noses. The grouser pitch can be an important subject of studies.

2.12 Conclusion

A review of the available literature, combined with the analysis summarized in Table 2.1(below), shows that there are generally three types of robots that could be used to secure the inside of buildings or malls. These are the Star-wheel robot, the variable configuration track robot, and the non variable track robot.

The Star-wheel robot might be appropriate for this task, except for its propensity to slip while climbing stairs. The size and volume of the Star-wheel robot is larger than those of track robots, and it thus needs more space for turning.

Most robots currently used in building security are the variable configuration track type. They can easily climb stairs by using their rotating arms to form an angle of attack to reduce the repulsive forces. The arms are also used to move the center of mass as required to climb or descend the stairs. The biggest disadvantages of these robots are the requirement for four DC motors and their high energy consumption.

Non variable tracked robots are not as versatile as the variable tracked type, but they have advantages that could make them attractive for building security. A major goal of this research is to show that an appropriate non variable tracked robot can be designed to reliably climb stairs. Such a design will result in a more energy efficient machine with fewer moving parts than an equivalent variable tracked robot.

Table 2.1 : Comparison of Different Locomotion Concepts.

Specification Concepts	Climbing Stairs	Energy Consumption	Speed	General Complexity	Volume	Proper space For Battery...	Robustness	Dynamic Balance	Cost	Min number of Motors
Walking Robot From[13to20]	*1	F	C	F	B	C	F	*2	F	>3
Wheel Robots [21-22]	F	A	A	A	A	A	B	B	A	2
Star Wheel [23,24,25]	B *3	A	A	B	C	A	B	A	B	2-3
Variable Configuration Tracked Robot [36-37]	A	C	B	B	B	B	B	A	B	>3
Non Variable Configuration Tracked Robot [28-35]	A *4	C	B	A	A	A	A	A	B	2

A: Excellent_ B: Very good _C: Good_ F: Not Good

*1. The climbing ability of a walking robot is highly dependent on the number of legs and the type of control system it has. A one leg robot does not have any static stability. A two leg robot also has difficulty in dynamic balancing for climbing. Four leg and six leg robots have static and dynamic stability. It is not possible to generalize walking robot characteristics.

*****2**.** As mentioned in **(**1), one leg and two leg robots have static and dynamic balance difficulties.

*3. A star wheel robot has slippage after climbing a certain number of stairs (the number depends on both the robot and the stair configurations).

*4. Non Variable configuration tracked robot could be a good stair climber if the angle of attack and its center of mass location are properly designed. This is discussed in Chapter 4.

Chapter 3

3.1 Static Analysis

3.1.1 Introduction

This chapter deals primarily with the geometric configuration selection of a tracked robot based on the permitted staircase measurements as defined in the North American Building Code. The static analysis of a tracked robot based on the best possible geometry design is then derived. The intended mission of this robot is the reconnaissance and guarding of a building facility. The optimal robot design for these purposes is one with the smallest possible size and the lightest possible weight. The purpose of this chapter is to summarize the static analysis of the applied forces on a tracked robot while it is climbing stairs. Chapter 4 (Dynamic Analysis of a Tracked Robot) contains a detailed force analysis. The main goal of this chapter is to graphically design a tracked robot, using the North American Standards for stair specifications to estimate some rough dimensions for the tracked robot. These include its length, width and height, plus its center of mass location. The minimum length of a tracked robot to make it stable during climbing stairs is also investigated.

Three other major tasks are covered in this chapter. The first of these is to determine the best angle of attack for reducing the repulsive force when the track robot starts climbing the first stair step. Finding the best pitch for the grousers that are mounted on the tracks to minimize energy losses in terms of slippage is the next task. Finally, a basic structural design of the tracked robot is developed.

3.1.2 Geometry of Stairs According to the Building Code

The Stairs Residential Building Code "Standards of stairs and Doors" [25] defines the range of permitted slopes, rises and landings. These ranges are meant to allow for safe navigation of staircases by humans, yet allow for enough variations to accommodate a range of space constraints and building styles. These stair parameters, as defined in the code, have a direct effect on the geometry and design of a stair climbing tracked robot. The stair geometry and the ranges of its parameters are shown in Figure 2.5 and Table 3.1.

Table 3: Staircase specifications as per the building code [25]

North American standards of stairs and steps			
Degree(γ)	$\Gamma; (In)$	$B\ ; (In)$	$A_{\Gamma,B} = \sqrt{\Gamma^2 + B^2}$
30 (Minimum)	6.5	11	12.78
50(Maximum)	9.5	8	12.419

Table 3.1 also shows that the maximum difference of length between the noses of two steps is about 4 millimeters.

The geometric and mass distribution elements of a stair climbing fixed geometry tracked robot can be derived from these specifications as follows.

3.1.3 Necessary Front Geometry for a Stair Climbing Tracked Robot

The following analysis is carried out for two types of tracks, namely smooth tracks and ones with grousers. While smooth tracks are rarely used for tracked robots or machines, they provide an entry point for the analysis. It should be noted that most of the literature dealing with track-ground interaction assumes that the tracks are smooth. For this analysis, it will be assumed that the track remains taught with no deflection even as it bears on the nose of a stair. It is also assumed that the robot has no yaw angle during climbing.

In Chapter 2, it was mentioned that the wheels of a wheeled or tracked robot must have radii that are greater than the rise of any stair or obstacle that the robot must climb. If the radii are less than the rise, the robot cannot cross the obstacle [18]. An analysis of the minimum

required radius of the front of the circular track to climb stairs is provided. Such a circular track shape will later be compared to the commonly used tracks that have an angled front.

Figure 3.1 shows the general case of a track with a circular front contacting a step of a stair case. As can be seen in the figure 3.1, the reaction force at the point of contact is normal to the tangent. The angle of this force relative to the level ground is the “angle of attack”. This reaction force has both horizontal and vertical components, provided that the angle of attack is acute. This will be the case when the radius of the circular track front is larger than the step height. The horizontal component must be, in the limit, less than or equal to the frictional force between the track and the ground to avoid slip. The vertical component must, in the limit, produce a moment larger or equal to that due to the weight of the robot in order to able to climb the step. If the radius of the front of the circular track is equal or less than the step height, the angle of attack is 90°, and the reaction force becomes strictly horizontal.

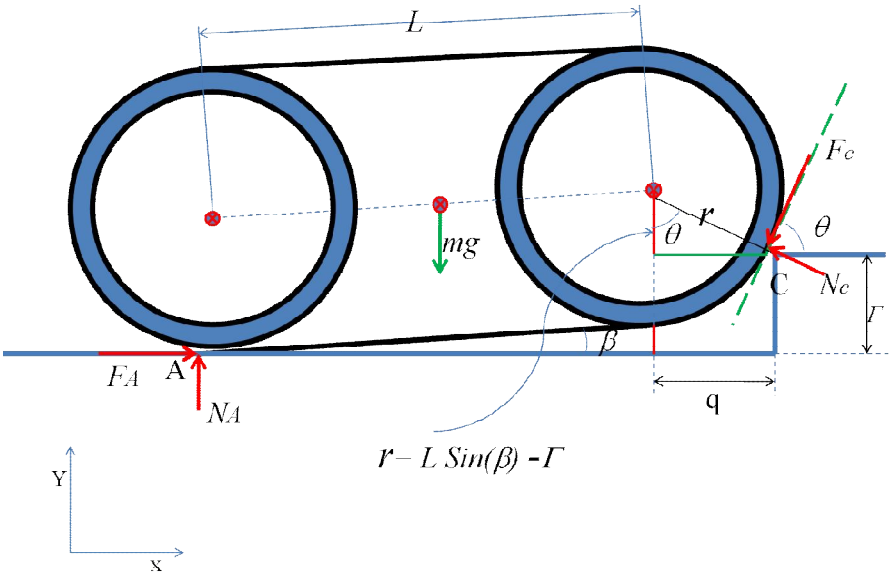


Figure 3.1 : A track with circular front contacting a step.

With reference to Figure 3.1, when the friction forces are just high enough to prevent the tracked robot from sliding, the balance of forces and moments are given as follows:

$$F_A = \mu N_A \qquad 3.1$$

$$F_C = \mu N_C \qquad 3.2$$

Where μ is the coefficient of friction of ground and stairs.

$$\sum F_X = \mu N_A - F_C \cos\theta + N_C \sin\theta = 0 \qquad 3.3$$

$$F_y = N_A - F_C \sin\theta + N_C \cos\theta - mg = 0$$

3. 4

$$\sum M_C = N_A(L\cos\theta + q) - F_A\Gamma - mg\left(\cos\theta\, {}^{L}/_{2} + q\right) = 0 \qquad 3.5$$

Where q and θ are given respectively by:

$$q = \sqrt{(r^2 - (r + \sin\beta - \Gamma)^2)} \qquad 3.6$$

$$\tan\theta = {}^{q}/_{(r - \Gamma + L\sin\beta)} \qquad 3.7$$

Solving the above set of equations for the reactions N_A and N_C gives the following:

$$N_A = \left[{}^{mg\left(\cos\theta\, L/2 + q\right)}/_{(L\cos\theta + q)} \right] \qquad 3.8$$

$$N_C = \left[{}^{mg\left(1 - \cos\theta\, L/2 + q\right)}/_{(L\cos\theta + q - \Gamma\mu)(\sin\theta - \mu\cos\theta)} \right] \qquad 3.9$$

Dividing Equation 3.8 by 3.9 and solving to express μ as a function of θ yields:

$$\mu = \frac{-(1 - L\cos\beta) - \sqrt{(1 - L\cos\beta)^2 - 4\left[\tan\theta\left(\frac{L}{2}\cos\beta + q\right)\right]\left[\tan\theta\left(1 - \frac{L}{2}\cos\beta + q\right)\right]}}{2\tan\theta\left(\frac{L}{2}\cos\beta + q\right)}$$

3. 10

By substituting for q and tan (θ) into Equation 3.10 it can be shown that the relationship between μ and θ depends only on r, L, Γ and β. It can be deduced from Figure 3.1 that as the track climbs the step the angle of attack θ decreases and the lifting force increases. Thus the limiting angle of attack occurs at the beginning of the climb when $\beta = 0$.

Using the values of Γ=0.17 m and L=0.6492 m, the relationship between μ and r for different values of β, as the robot climbs the step, is plotted in Figure 3.2

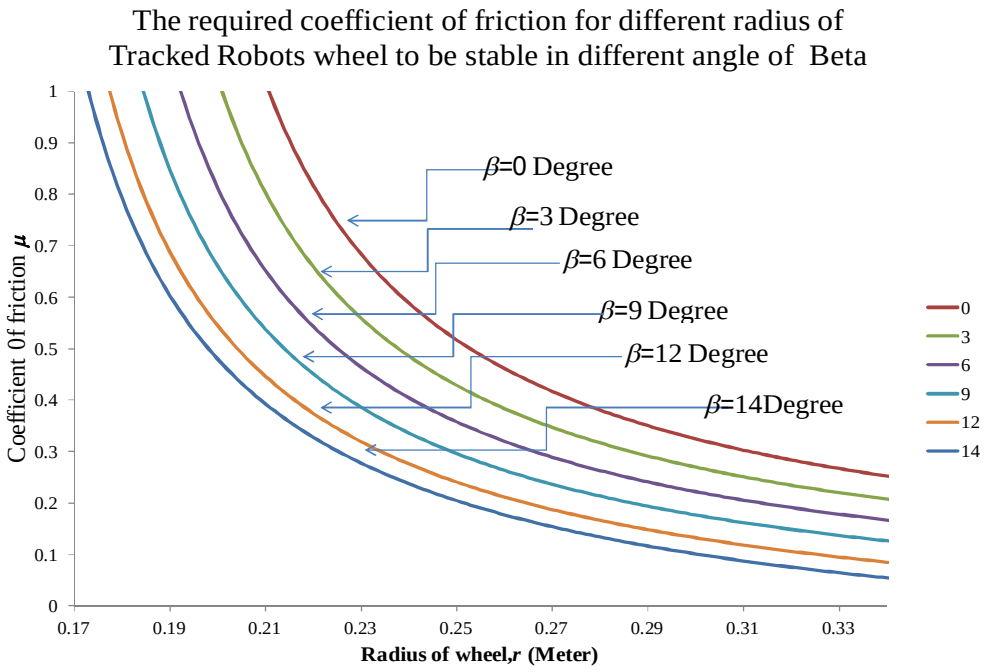


Figure 3.2 : Coefficient of friction versus the necessary radius of the circular track front.

As can be seen in Figure 3.2, as the track climbs the step, β increases, and the required radius decreases for the same coefficient of friction. The rightmost curve, corresponding to $\beta = 0$ highlights the minimum limit of r necessary for a given coefficient of friction to ensure that the track of length L can climb the step of height Γ. The figure shows that the minimum required radius for a track, at μ = 1, is 21.5 cm. The minimum radius for a practical coefficient of friction of 0.8 is 22.5 cm. Such a radius results in a larger and longer robot size that is heavier and consumes more energy.

An alternative track configuration that is more suitable for climbing objects including stairs has its front flat but at an angle. This configuration has been discussed in the literature review. The configuration of such a track front will be studied next

3.1.4 Case of Smooth Track:

Figure 3.3 and 3.4 illustrates the angled robot track and its reaction forces as it starts climbing the first step. Once the robot contacts the first step and bear on it, the reaction forces are no longer distributed along the length L but are solely concentrated at points *A* and *C*, i.e., the robot's rear track wheel and the contact with the first stair nose. In order for the robot to be able to pull itself up over the step the moment about *A* due to the upward component of the forces at *C* should be larger than that due to the downward weight at the center of mass. Furthermore, the horizontal component of the forces at *C* must me be less than or equal to the horizontal force at *A* so that the robot will not slide backward. The angle between level ground and the track robot, β, starts at zero and rises to $\beta = \tan^{-1} \Gamma/L$, where Γ is the rise of first step, and *L* is the length of the bottom of the robot track. With reference to Figure 3.4, balancing the moment and the forces in the *X* and *Y* directions gives:

$$\sum F_X = F_A - F_C \cos(\theta + \beta) - N_C \sin(\theta + \beta) \qquad 3.11$$

$$\sum F_y = N_A - F_C \sin(\theta + \beta) + N_C \cos(\theta + \beta) - mg = 0 \qquad 3.12$$

$$\sum M_C = N_A (L \cos\beta + q) - F_A \Gamma - mg(L_C \cos\beta + q) \qquad 3.13$$

Solving the set of equations above for N_A and N_C gives the following:

$$N_A = \frac{mg(L_C \cos\beta + q)}{(L \cos\beta + q - \mu\Gamma)} \qquad 3.14$$

$$N_C = \left[\frac{mg(L \cos\beta - L_C \cos\beta - \mu\Gamma)}{(-\mu \sin(\beta+\theta) + \cos(\beta+\theta))(L \cos\beta + q - \mu\Gamma)}\right] \qquad 3.15$$

Dividing Equation 3.14by 3.15 and solving to express μ as a function of θ yields:

$$\mu = \frac{-(\Gamma - L\sin\beta + \Gamma(\tan(\theta+\beta))^2)}{2(L\sin\beta\tan(\theta+\beta) - \cos\beta L_C(\tan(\theta+\beta))^2)} + \frac{\sqrt{(\Gamma - L\sin\beta + \Gamma(\tan(\theta+\beta))^2)^2}}{2(L\sin\beta\tan(\theta+\beta) - \cos\beta\, L_C(\tan(\theta+\beta))^2)} + \frac{\sqrt{4[(L_C\cos\beta(\tan(\theta+\beta))^2)(L\sin\beta\tan(\theta+\beta) - \cos\beta L_C(\tan(\theta+\beta))^2)]}}{2(L\sin\beta\tan(\theta+\beta) - \cos\beta\, L_C(\tan(\theta+\beta))^2)}$$

3. 16

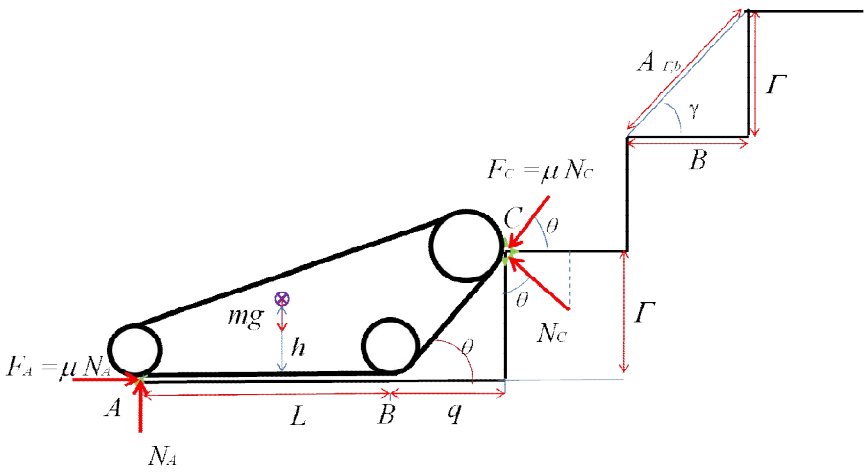


Figure 3.3 : A Tracked robot as it starts climbing the first stair

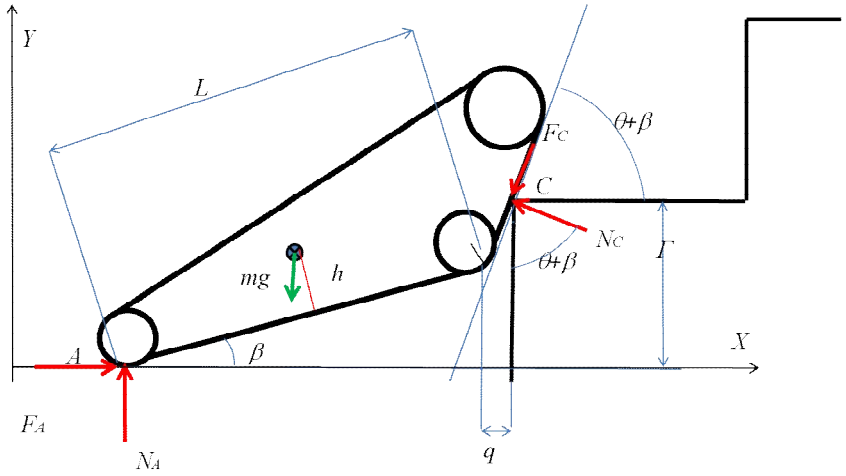


Figure 3.4 : The tracked robot while climbing the first stair.

With reference to Figure 3.5, as the tracked robot starts to climb the first step, the projection of the total length of the tracked robot ($L \cos \beta + q$) onto the ground depends on β, L, and Γ as follows:

$$q = \frac{\Gamma - L \sin \beta}{\tan(\theta + \beta)} \qquad 3.17$$

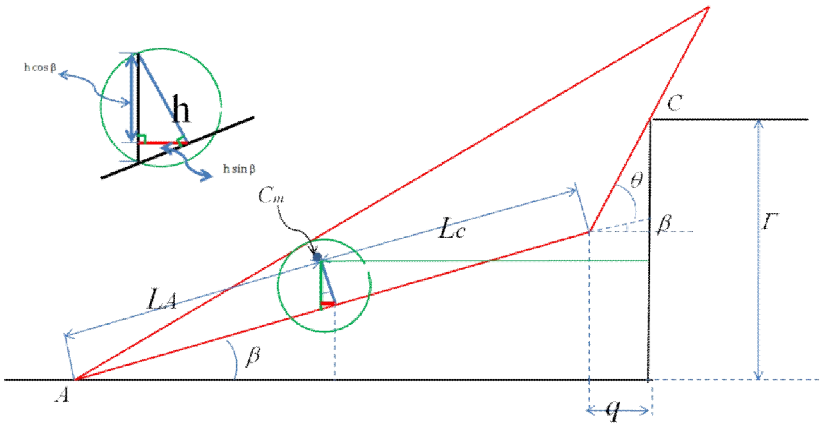


Figure 3.5 : The geometry of tracked robot as well as the variable length of q.

Using the values of Γ=0.17 m and L=0.6492 m, the relationship between the robot's angle of attack and the coefficient of friction is shown in Figure 3.6 as the robot climbs the step.

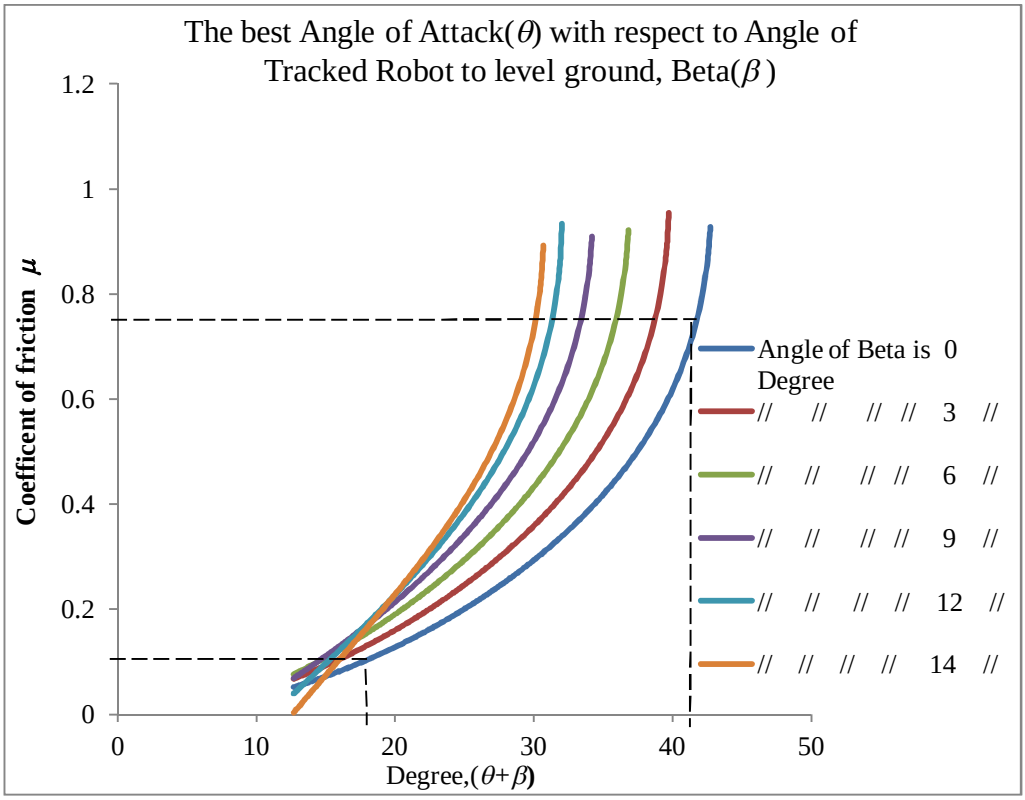


Figure 3.6 : Relation between the coefficient of friction and different angles of attacks as the robot progressively climbs the step.

The robot is horizontal with respect to the ground when it first contacts the stairs (i.e., $\beta = 0$). The rightmost curve in Figure 3.6 shows the necessary angle of attack for different ground-track coefficients of friction. This plot shows that the necessary angle of attack for the maximum possible coefficient of friction of 0.1 is 17.5^0 and for a coefficient of friction of 0.78 the angle of attack is 41.5^0. The necessary angle of attack is lower for lower coefficients of friction. As the robot climbs the step, its angle β with respect to the ground increases. This increase steepens the angle of contact between the front of the robot and the nose of the step, and causes a left shift of the plots. As shown in the graph, the maximum angle of attack decreases as β increases. Hence, for the case of a smooth track the maximum angle of attack is limited by the case where β is at its maximum value for a given step rise. It should be noted that a smaller angle of attack results in a longer robot.

3.1.5 Case of a Track with Grousers:

Most tracked vehicles use tracks with grousers. Such grousers reduce slippage when traversing uneven or rough terrain by grabbing onto protrusions in the terrain. For the case of climbing stairs, the grousers' grip the nose of a stair and pull the robot up. An analysis of the effect of adding grousers to the belt to achieve a better grip when the tracked robot climbs stairs is carried out in this section.

With reference to Figure 3.7, as the robot front contacts the first stair and starts the climb, the points of contact of the track are at the rearmost grouser (Point *A)* and the grouser near the top front of the robot (Point *C)*. (This analysis also assumes that the track does not deflect under the concentrated load at the nose of a stair.) Furthermore, while grousers are generally constructed with a small angle ψ, this analysis assumes that $\psi = 0$.

With reference to Figure 3.6, balancing the moment and the forces in the X and Y directions gives:

$$\sum F_X = F_A - F_C \qquad 3.18$$

$$\sum F_y = N_C + N_A - mg = 0 \qquad 3.19$$

$$\sum M_C = \Gamma\mu N_A + (L\cos\beta + q)\,N_A - (L_C\cos\beta + q)mg \qquad 3.20$$

Where q is given by equation 3.17

Solving the set of equations above for N_A and N_C gives the following:

$$N_A = \frac{mg(L_C \cos\beta + q)}{(L\cos\beta + q - \Gamma\mu)} \qquad 3.\ 21$$

$$N_C = mg\left(1 - \frac{(L_C \cos\beta + q)}{(L\cos\beta + q - \Gamma\mu)}\right) \qquad 3.\ 22$$

With reference to equation 3.17, the equation of coefficient of friction μ with respect to θ, and β, is as follow:

$$\mu = \frac{+L\sin\beta - \Gamma}{\tan(\theta + \beta)} \qquad 3.\ 23$$

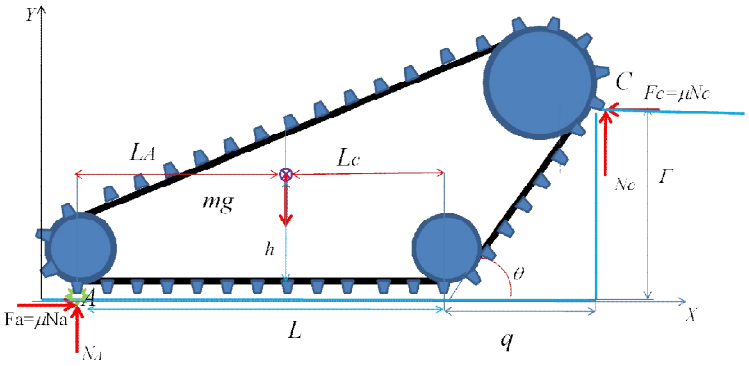


Figure 3.7 : Robot pose as it starts climbing the first step.

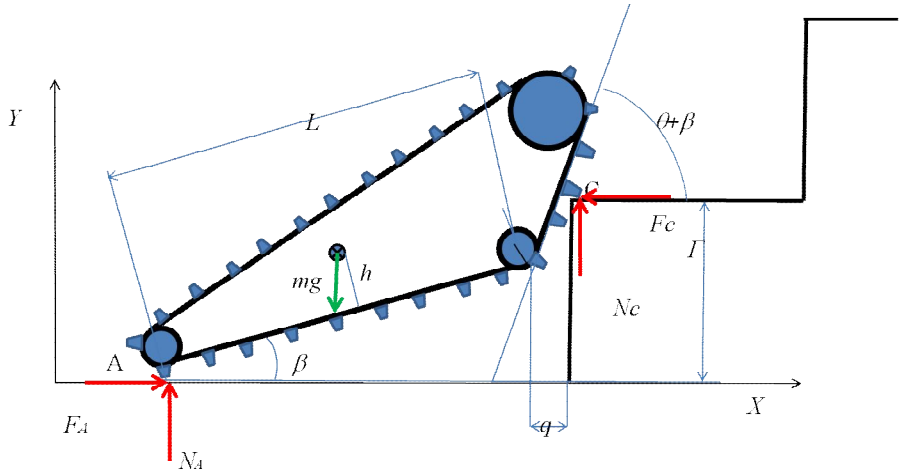


Figure 3.8 : Robot's pose during the climb of the first step. For static analysis, the angular acceleration, Ω, is zero and the torque, T, is not considered.

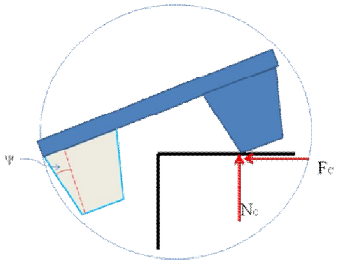


Figure 3.9 : Grouser interaction with step.

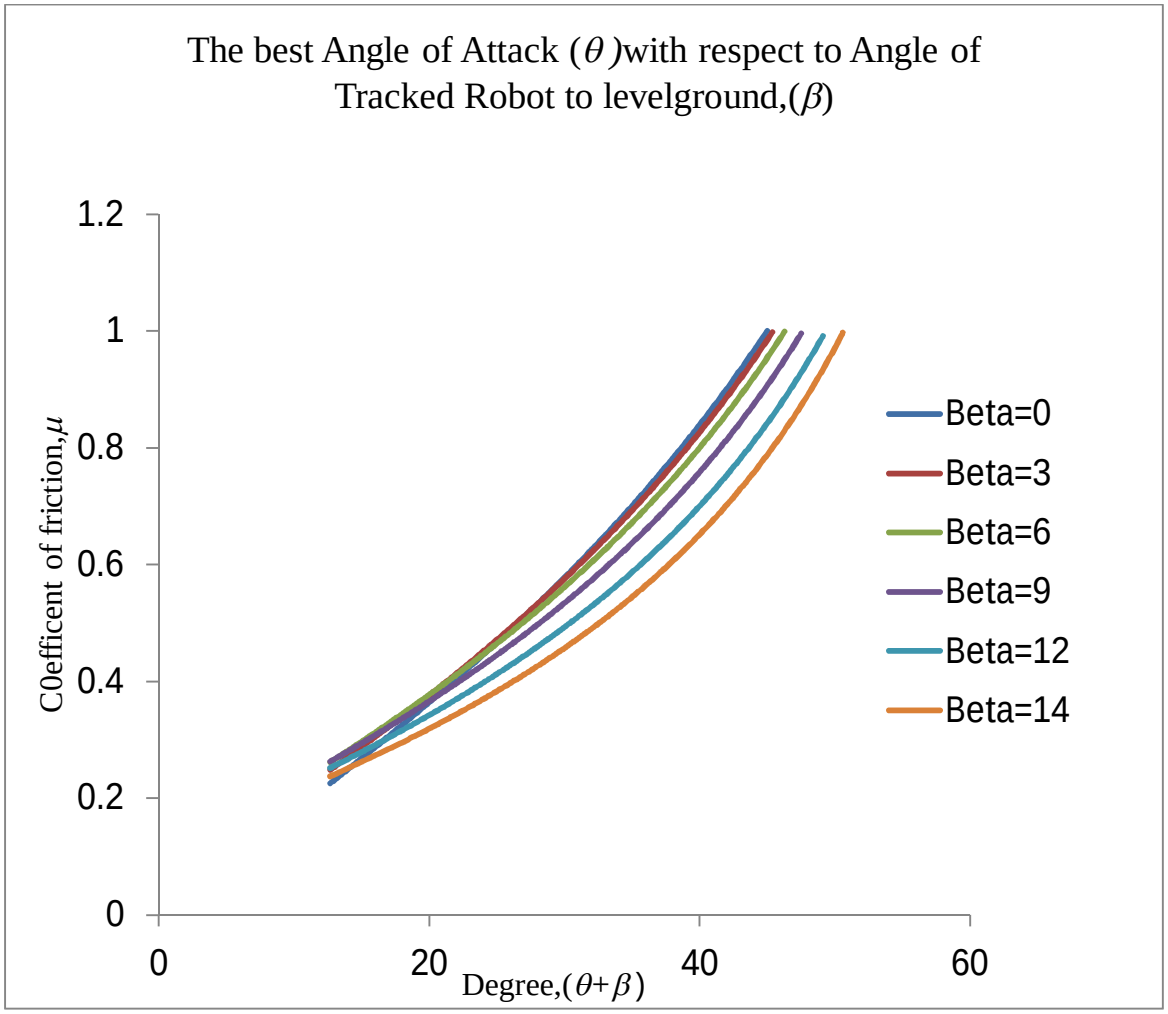


Figure 3.10 : Relation between coefficient of friction and angles of attack as a robot with grousers climbs the first step.

Again using the values of Γ=0.17m and L=0.6495m, the relationship between the robot's angle of attack and the coefficient of friction for the case of tracks with grousers is shown in Figure 3.8 as the robot climbs the step.

Figure 3.10 shows that the robot's necessary angle of attack decreases with decreasing track-ground coefficient of friction. This is similar to the smooth track case. In this case, however, as the angle of the robot with respect to the ground, β, increases the necessary angle of attack θ increases. This is the reverse of the case where the tracks do not have grousers. Thus, in this case the necessary angle of attack for a given coefficient of friction is set by the position of the robot when $\beta = 0$, i.e. at the start of climbing the first step. For a maximum possible coefficient of friction of one, the necessary angle of attack should be at or just under 45°. The above analysis shows that a tracked robot with grousers can have a steeper angle of attack than an equivalent one without grousers. This allows such a robot to be shorter and more compact than one with no grousers. The addition of grousers would also allow the use of tracks with a lower coefficient of friction with the ground, which imply harder and more durable ones. Furthermore, as the robot starts to climb the stairs, the coefficient of friction necessary to prevent slip decreases.

Minimum Required Track Length for Climbing Stairs

For continuous stability and the ability to climb staircases, the robot tracks must contact a minimum of two stairs at all times during the climb. This dictates that, from a geometric perspective and with reference to Figure 2.5 and Table 3.1, the minimum length of the robot's track bottom must be equal to or greater than the distance between two stair steps noses, i.e.

$$L \geq 2A_{\Gamma,B} \qquad 3.24$$

where for the North American Building code the maximum L would be 25.56 in (64.92 cm).

The analysis below will show that the minimum length of the track bottom must be larger than L.

3.1.6 Pose Analysis of the Tracked Robot as it Climbs Stairs

As the robot navigates up and down a staircase, it goes through several distinct pose phases that merit separate analyses. These phases are displayed graphically in Figures 3.11 and 3.12 for the stair climb and stair descend respectively. The nature of the interaction between the

robot's track and the staircase steps during each of these phases is distinct. It should be noted that the pose phases described below are developed for the case when the robot is moving at very slow speeds, i.e. a quasi-static condition. The nature of the phases changes slightly at higher speeds when inertia, momentum, and rebound effects are taken into account. However, since in practice the robot is driven at very slow speeds when navigating staircases, an analysis based on quasi-static assumptions is justified. With reference to Figure 3.11, the staircase ascend has strictly seven phases. Description of these phases follows:

Phase 1: This phase starts when the track contacts the lowest step of the staircase. During this phase the robot pulls itself up on the first step. The robot experiences both an upward and a horizontal motion coupled with a rotation about the point of contact of the track with the staircase landing. This phase ends when the front tip of the flat bottom of the robot's track contacts the nose of the same step.

Phase 2: The start of the second phase coincides with the end of the phase 1. During this phase the flat bottom of the robot's track is in constant contact with the nose of the first step as it climbs it. The rate of vertical and horizontal motion of the robot increases as its angle with the horizontal increases. The second phase ends when the front of the flat bottom of the track contacts the second step. If the length of the flat bottom of the track is equal to the minimum length as defined in Section 3.6, at the end of *Phase* 2 the robot bottom would be in contact with the staircase landing at its rear, in contact with the nose of the first step at the middle of the flat bottom, and in contact with the nose of the second step at the front of the flat bottom.

Phase 3: Again here, the beginning of *Phase* 3 coincides with the end of *Phase* 2. With reference to Figure 3.11, it can be shown that for the robot to be able to continue its climb smoothly up the staircase the line of action of the robot center of mass must be at the nose of the first step when the tip of the flat bottom contacts the second step. This means that the center of mass must be at a distance $A_{\Gamma,B}$ - h tan(γ), or less, from the tip of the flat bottom of the track. This position of the center of mass would allow a clockwise moment which causes the track to continue to be in contact with the second step. If the center of mass is forward of this position, the robot would tip forward before the tip of the flat bottom contacts the second step which would cause the angled front of the robot to contact the nose of the second step. While this is not an ideal climbing condition, nevertheless the robot would be able to continue its climb. If, however, the line of

action of the weight is backward of the nose of the first step, a counter clockwise moment develops and causes the robot to start to tip backward.

Phase 4: The start of this phase coincides with the end of the previous phase. If the length of the flat bottom of the robot's track is identically equal to $2A_{\Gamma,B}$ then *Phase* 4 occurs momentarily when the track bottom contacts three consecutive stair noses simultaneously. However, if the flat bottom of the track length is longer than $2A_{\Gamma,B}$ then *Phase* 4 continues as long as the bottom of the track continues to contact three consecutive step noses. This *Phase* ends when the utmost back point of the track bottom is about to leave contact with the lowest of the three steps.

Phase 5: This phase starts when the track bottom clears the lowest of the three steps from the previous phase and is in contact with only two consecutive step noses. The phase ends when the utmost front of the track bottom contacts a next step nose.

It should be noted that the robot pose alternate between *Phases* 4 and 5 as the robot advances up the staircase. This continues until the two topmost steps are encountered, at which point the robot poses advances from a *Phase* 5 to a *Phase* 6.

Phase 6: This phase marks the end of the staircase climb. The start of the phase starts when the rear of the flat bottom of the robot track clears the step before the last of the staircase. Because the line of action of the robot's weight is at or forward of the nose of the last step, a condition necessary for the robot to have reached this stage is explained in *Phase* 3. The clockwise moment due to the weight will cause the robot to tip forward until it lays flat on the top staircase landing. At this point in time the front of the flat bottom of the track is at a distance $A_{\Gamma,B} - h \tan(\gamma)$ from the nose of the topmost step.

Phase 7: During this phase the robot advances on the top landing from the end of the pose of *Phase* 6. *Phase* 7 ends when the rear of the track bottom clears the nose of the topmost step.

Figure 3.12 shows the pose phases of the tracked robot as it descends the staircase. For clarity and distinction from the ascending phases, the first of these will be termed *Phase* 8.

Phase 8: The start of this phase occur when the front of the flat bottom of the track encounters the nose of the topmost step. This phase ends when the center of mass coincides with the nose of that step.

Phase 9: The beginning of this phase coincide with the end of *Phase* 8. Any incremental forward motion causes the center of mass to pass the nose of the step. This causes a moment that tilts the

front of the robot downwards as it proceeds in its forward motion. This phase ends when the tip of the flat bottom of the track contacts the landing of the next step.

Phase 10: At the start of the tenth phase the robot is in a pose where it is angled forward as described at the end of the previous phase. The robot proceeds forward and this phase end when the tip of the flat track bottom touches the nose of the step. At this time the flat track bottom is in contact with two successive step noses.

Phase 11: The beginning of this phase coincides with the end of the previous one. The robot advances downwards at an angle equal to that of the staircase. This phase ends when the center of mass coincides with the nose of second step (Figure 3.12 *Phase*11).

Phase 12: The beginning of this phase coincide with the end of *Phase* 12. Any incremental forward motion causes the center of mass to pass the nose of the step. This causes a moment that tilts the front of the robot downwards as it proceeds in its forward motion. This phase ends when the tip of the flat bottom of the track contacts the landing of the next step.

Phase 13: The motion of tracked robot is similar to that of *Phase* 10.

Phase 14: This phase ends when the flat bottom of the track is in contact with three successive step noses.

Phase 15: The motion of tracked robot is similar to that of *Phase*11.

Phase 16: The motion of tracked robot is similar to that of *Phase*12.

Phase 17: This phase starts from the end of the previous phase. The robot advances downwards until the center of mass reaches the nose of the lower of the two steps, signifying the end of this phase.

In a similar fashion to what takes place during the robot ascent of the staircase, the robot pose alternates between *Phases* 9 to 13 in sequence as the robot advances down the staircase, until it reaches the bottom landing at a phase equivalent to *Phase*13.

Phase 18: This is the final phase in the descent. It starts at the end of the previous phase. The robot advances forward on the bottom landing till the back of the flat bottom of the track clears the nose of the bottom step. At this point the robot bangs on the bottom landing and lies flat on it.

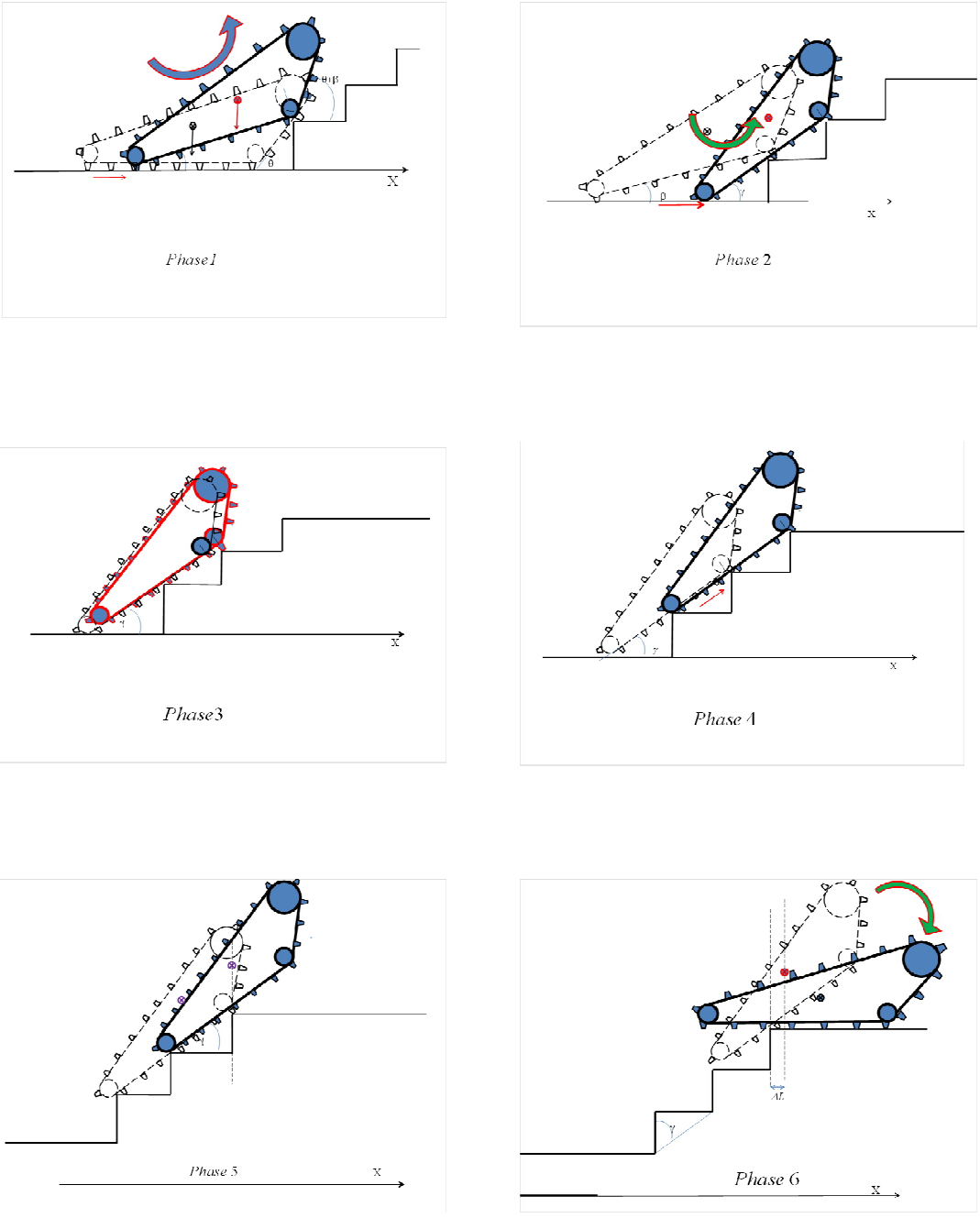
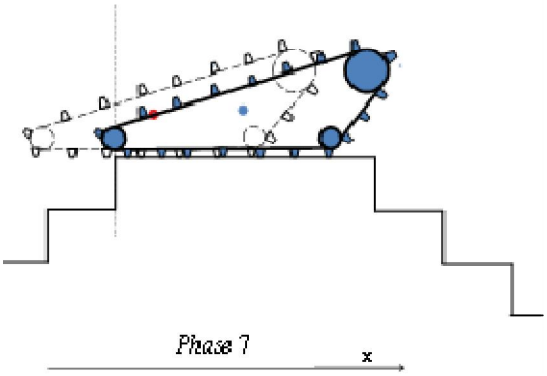
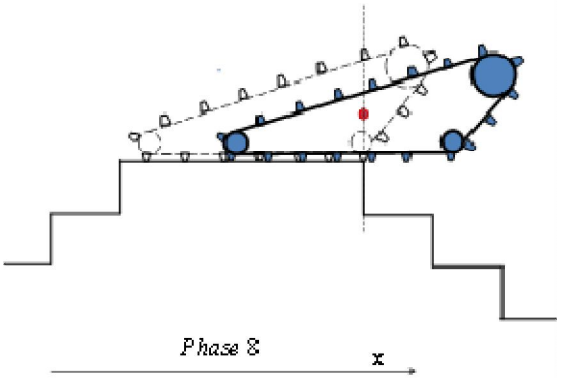
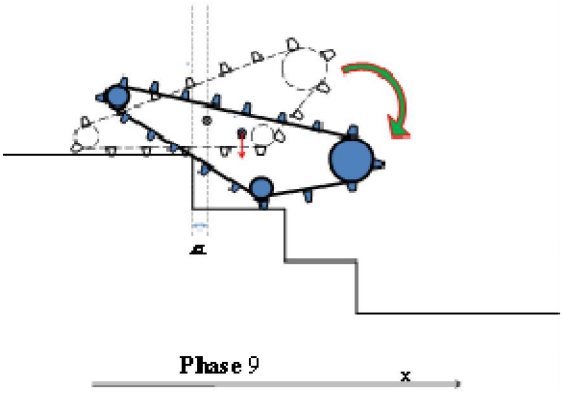
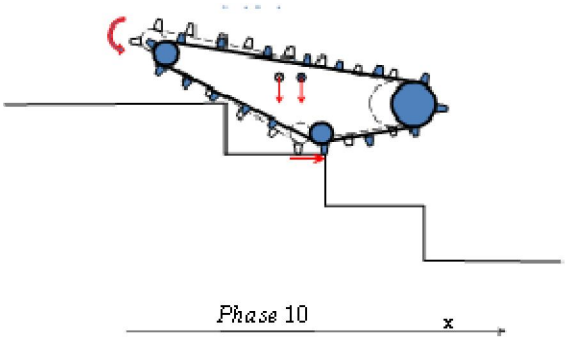


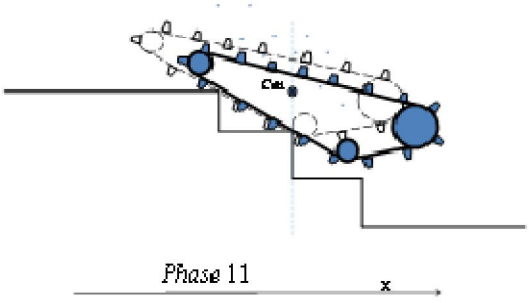
Figure 3.11 : Motion of tneTracked Robot during climbing in different phases.

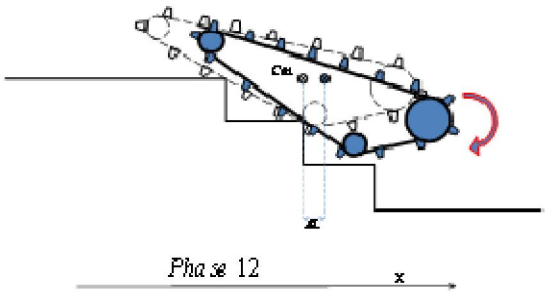

Phase 7
x


Phase 8
x


Phase 9
x


Phase 10
x


Phase 11
x


Phase 12
x

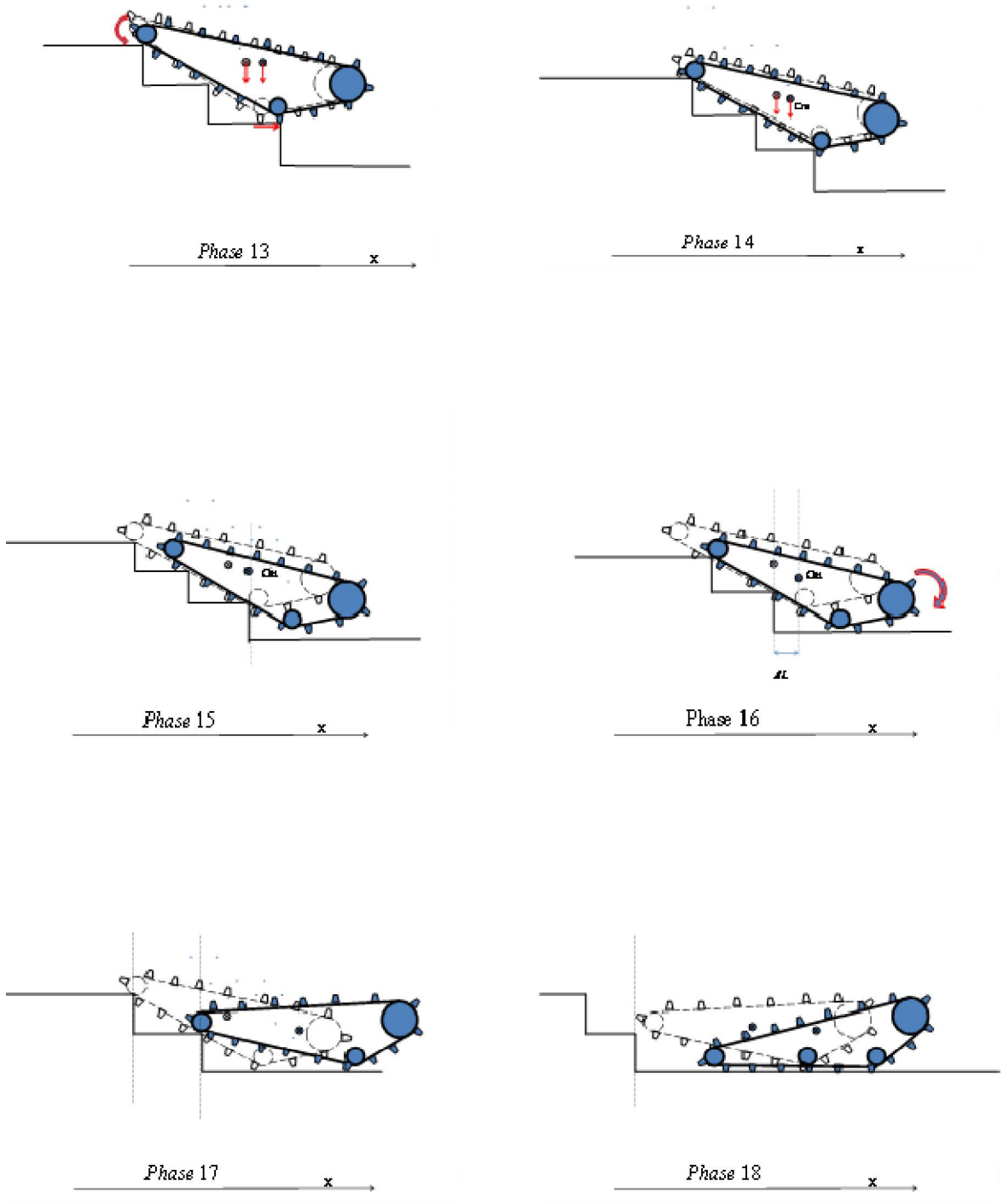


Figure 3.12 : Motion of tneTracked Robot during descending different phases.

3.1.7 Necessary location of Center of Mass for successful Ascent and Descent

In the pose analysis conducted below, Figure3.13 (*Phase 3*) showed that the location of the center of mass must be at a distance $A_{\Gamma,B}$ - h tan(γ), or less, from the tip of the flat bottom of the track. It was also seen from *Phase* 9 showed that the center of mass must be at a distance less than $A_{\Gamma,B}$ from the front of the flat bottom of the track. Based on these two conditions, it can be concluded that the center of mass must be at a location as defined in *Phase* 3, with the optimum location being at exactly $A_{\Gamma,B}$ - h tan(γ).

The tilting moments that come into play at *Phases* 3 and 9 are a function of the height of the center of mass, h, for a given staircase. This tilting moment, and the associated banging of the robot on the next step as it navigates the staircase, can be reduced by lowering the height of the center of mass.

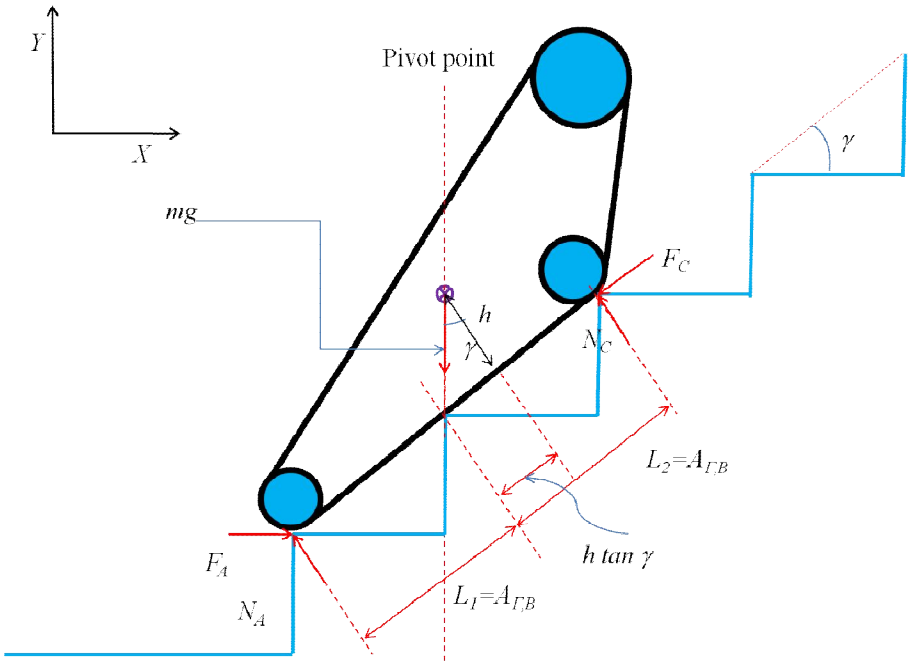


Figure 3.13 : The necessary location of the center of mass.

3.1.8 Track Shape and Dimensions

In general the track and grouser design greatly impact the efficiency of a tracked vehicle, particularly when traversing rough terrain. This is especially true when the robot is climbing staircases where the steps are periodic and with a uniform pattern. The pitch and size of grousers was briefly described in Chapter 2. The general geometry of the track with grousers is shown schematically in Figure 3.14. With reference to this figure the primary parameters are; the grousers root thickness H_1, height h_1, the angle ψ, the width of the track W_1, and the grouser's pitch (S+W_1).

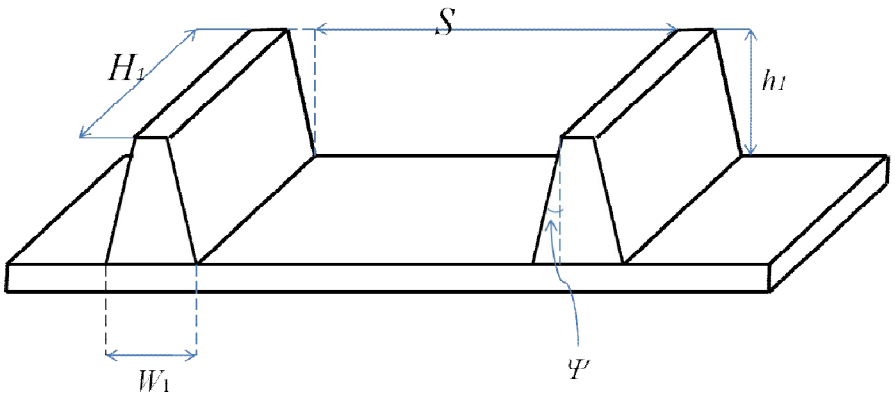


Figure 3.14 : Track grouser configuration.

The relationship between the grouser's pitch and the distance between two successive step noses has a direct impact on the ability of the robot's track to grip a step landing. This in turn affects its ability to climb the staircase, and the smoothness of the climb. In order for the grousers to grip on the landing of successive steps landings, when the tracked robot is climbing straight up the staircase, the length between two successive step noses, $A_{\Gamma, B}$, should be exactly equal to a whole multiple of the grousers pitch, or marginally longer. For better and smoother climbing ability, the grouser pitch should be as small as possible, such that the grousers are continually gripping the stair steps. If the pitch is too large, the track may have to slide with respect to a step nose until the next grouser contacts and grabs the stair nose. However, because of the track-staircase interaction, a tracked robot generally experiences alternating left and right yaw action as it climbs a staircase, as described in Chapter 2. The minimum distance between grousers needs

to take this into account. This alternating yaw action is primarily due to the small differences between the speeds of the left and right tracks, as well as the alignment of the two tracks' grousers as the robot moves. It is necessary to make sure that there is no possibility for the grousers to ride on the step noses during the climb.

Figure 3.15 shows the geometry of the track with respect to a step when the climb is at a yaw angle Φ.

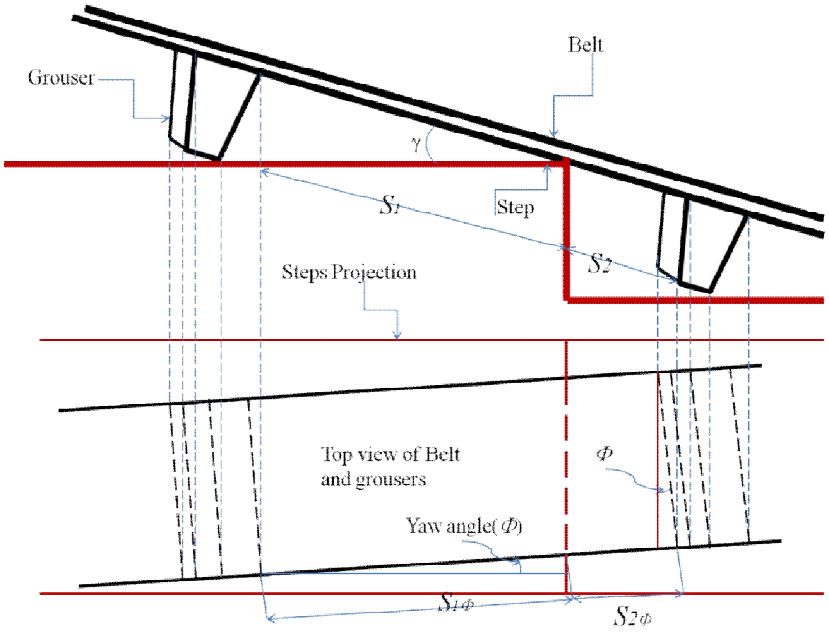


Figure 3.15 : Track/step geometry during yaw motion.

With reference to the figure, the track geometry with respect to a step nose is primarily defined by the projections of the distances S_1 and S_2 of two successive grousers from the nose of the step with respect to the yaw angle Φ. The minimum values for these two parameters sum up to the minimum pitch for the track grousers that would be necessary to prevent slip and ensure successful and efficient climb. The minimum values of the parameters are given by:

$$S_{1\phi} = h_1(\tan\gamma + \cot\gamma - \tan\psi)/\cos\Phi \qquad 3.25$$

$$S_{2\phi} = H_1 \tan\phi \qquad 3.26$$

The minimum pitch is hence given by:

$$P = \left[h_1(\tan\gamma + \cot\gamma - \tan\psi)/\cos\Phi\right] + H_1 \tan\phi \qquad 3.27$$

The exact track length and width can be defined using standard timing belt equations once the sprocket width and locations, including those forming the angled front, are known. With these parameters known and using the pitch equation, the grousers geometry, and number can be calculated.

3.2 Conclusion

Static analysis is useful for calculating the minimum size of a tracked robot capable of climbing stairs of a specified size. This analysis also shows the best track angle of attack for climbing stairs. By comparing the angles of attack of a tracked robot with a flat track to a robot with grousers, the analysis shows that the best angle of attack for the robot with grousers is larger than that of the robot without grousers. This allows the robot with grousers to be smaller than the alternative one. Further analysis can reveal the best grouser pitch to reduce stair climbing slippage while maintaining stability and efficiency.

Static analysis techniques give a good preliminary robot design, based on low speed motion and the maximum expected stair angle. This gives a reasonable estimate of the minimum required force and torque, as well as the size of gear box and the power of the DC motors.

This is a preliminary estimate for designing the track robot. These design parameters will be refined by using dynamic techniques to analyze the robot motion in more detail. This analysis is the topic of the next chapter.

Chapter 4

4.1 Dynamic Analysis of the Tracked robot

The generally low traveling speed of tracked robots allows static analysis techniques to be used to analyze their behavior in most cases. However, for those instances when the robot has to travel at high speeds or undergoes fast pose transitions, dynamic analysis yields a more accurate model of the robot motion. This is necessary in order to analyze the robot stability, and for controller design. The main purpose of this chapter is to study the dynamic behavior of a tracked robot while it is climbing and descending stairs, and while it is traveling on flat surfaces.

One of the interesting subjects of study is when vehicles encounter obstacles. When a mobile tracked robot traveling on level ground encounters stairs and starts to climb the first step, the motion changes from one direction to two directions. The vehicle now travels along a curved line instead of a straight line. As the robot starts to climb, there are two additional motions to consider: the vertical motion and the rotational motion with respect to the center of mass.

Studying the climbing and descending of obstacles is important because of the requirement for dynamic stability. This is directly related to the robot's structure, the location of its center of mass, the grouser geometry, as well as its speed and acceleration. The chapter also compares the motion dynamics of the tracked robot with grousers to that of one without grousers (flat track).

One of the main purposes of the dynamic analysis is to find the critical poses and conditions that would have adverse effects on a tracked robot while it is climbing and descending stairs. Stair descent can be more critical than stair climbing due to the greater potential for toppling. This analysis is intended to be a preliminary study of the performance characteristics and dynamics of tracked robots. The robot velocities, accelerations, and motor torque versus the robot poses during stair climbing and descent are studied and used as proxies of robot performance.

Three main assumptions are necessary to simplify this analysis: a) there is no track slippage and the grousers have a perfect grip while the tracked robot is climbing stairs, b) there is no track deflection due to the contact with the stair step nose, c) the robot climbs and descends straight with no yaw angle.

Since the pose and motion of the robot during the climb and descend phases described in Chapter 3 are distinct, the tracked robot dynamic equations are developed for each phase independently. A small number of phases do share the same dynamic equations, and the analysis is combined for these cases.

4.2 Case of Tracks with Grousers:

4.2.1 Dynamic Analysis of the Robot in *Phase* 1

The tracked robot starts to climb the stairs from a flat surface. The robot begins with a zero angle between the bottom of the robot track and the ground. As the robot starts to climb the first step, the angle between the bottom flat part of its track and level ground increases.

With reference to Figure 3.6, q is defined as the projection of the length between the nose of the step and the beginning of the angled front of the track onto the ground. The maximum value of q $is \frac{\Gamma}{tan\ \theta}$. When the tracked robot begins to climb the first step β is zero. As the robot continues its climb β increases and q decreases. When the front of the bottom of the track reaches the nose of the first step, signaling the end of this phase, the value of q is zero and $\beta = cos^{-1}\frac{\Gamma}{L}$. , where Γ is the rise of the step and L is the length of the flat bottom of the track.

Also, with reference to Figure 3.6 the distance from the robot's center of mass to the step contact point, $R_{r1,}$ is given by:

$$R_{r1} = \sqrt{(L_C + q)^2 + (\Gamma - L_C \sin\beta - h\cos\beta)^2} \qquad 4.1$$

Where h is the height of the robot's center of mass from the flat bottom of the track.

With reference to Section 3.1.5 the location of the center of mass must be less than $A_{\Gamma,B}$ - $h\ tan(\gamma)$ from the front wheel. Therefore, this parameter is defined as:

$$L_c = A_{\Gamma,B} - h\ tan\gamma \qquad 4.2$$

And the location of the center of mass with respect to the rear wheel is defined as:

$$L_A = A_{\Gamma,B} + h\ tan\gamma \qquad 4.3$$

During *Phase* 1, the robot's center of mass goes through three motions: one translation each in the x and y directions, and one rotation with respect to its center of mass.

With reference to Figure 3.9, the equations of motion of the robot are given by:

$$\sum F_x = m\frac{d^2x}{dt^2} = m(a_{cm})_x = F_M\cos(\theta + \beta + \psi) + F_A - F_C \qquad 4.4$$

$$\sum F_y = m\frac{d^2y}{dt^2} = m(a_{cm})_y = F_M\sin(\theta + \beta + \psi) + N_A + N_C - mg \qquad 4.5$$

$$\sum M_C = \left(I + m{R_{r_1}}^2\right)\Omega = + N_A(\mu\Gamma - L\cos\beta - q\ ) + (L_C\cos\beta + q)mg \qquad 4.6$$

Where F_X, and F_Y, are the forces on the center of mass, M_C is the applied moment at point C (the stair nose contact point), Ω is the angular acceleration of the tracked robot about its center of mass and is given by$\frac{d^2\beta}{dt^2}$, and $\frac{d^2x}{dt^2}$, $\frac{d^2y}{dt^2}$ are the components of linear acceleration.

The output torque T of the gear box provides the force F_M that acts on the step's nose at point C to pull the robot up and forward. They are related by:

$$F_M = \frac{G*Eff}{r} * T \qquad 4.7$$

Where G is the gear box ratio, E_{ff} is the gear box efficiency, and r is the radius of the driven pulley.

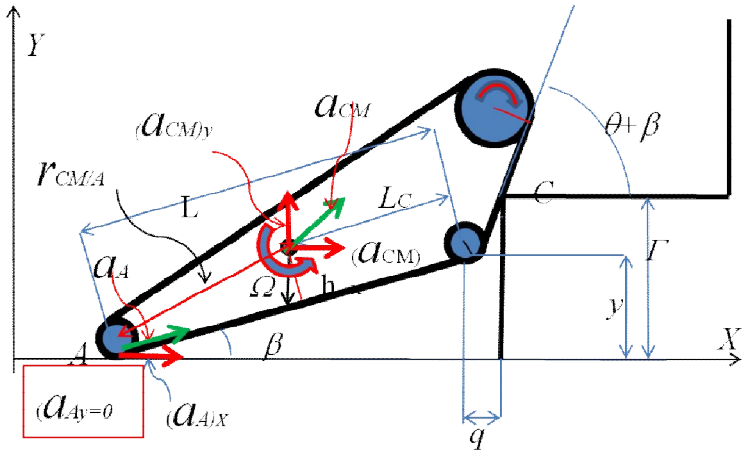


Figure 4.2.1 : Pertinent kinematic and geometric relations of the robot in *Phase*1.

The acceleration of the center of mass of the robot with respect to the acceleration of the rear pulley of the track is given by:

$$a_{CM} = a_A + \Omega \times r_{Cm/A} - \varpi^2 r_{Cm/A} \qquad 4.8$$

$$(a_{CM})_x i + (a_{CM})_y j = a_{Ai} + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_A \cos\beta - h\sin\beta) & (L_A \sin\beta + h\cos\beta) & 0 \end{vmatrix} - \begin{vmatrix} i & j & k \\ 0 & 0 & \varpi^2 \\ (L_A \cos\beta - h\sin\beta) & (L_A \sin\beta + h\cos\beta) & 0 \end{vmatrix} \qquad 4.9$$

The acceleration of the rear of the tracked robot in the Y direction is zero. The acceleration in the X direction is proportional to the angular acceleration of the driven pulley.

The angular velocity of the robot is negligibly small, and $\dot{\beta}$ will be much smaller, therefore, it is assumed zero for all phases.

The equations of acceleration in *X* and *Y* directions are:

$$\frac{d^2x}{dt^2} = (a_{CM})_x = (a_A)_x - \left[\ddot{\beta}(L_A sin\,\beta + h\,cos\,\beta)\right] \qquad 4.10$$

$$\frac{d^2y}{dt^2} = (a_{CM})_y = \left[\ddot{\beta}(L_A cos\,\beta - h\,sin\,\beta)\right] \qquad 4.11$$

As has been mentioned above, *Phase* 1 encompasses$\beta = 0\ to \beta = cos^{-1}\frac{\Gamma}{L}$. Hence, the above equations are valid only within that range.

4.2.2 Dynamics of a Tracked Robot in *Phase* 2

The dynamic analysis of *Phase* 2 is somewhat similar to that of phase 1. However, here the contact geometry between the tracks, the ground, and the stairs are different, and hence the magnitude and direction of the forces are also different.

With reference to Figure 4.2.2, as the front of the bottom of the track reaches the nose of the first step the only existing angle between the tracked robot and the steps is now the angle between level ground and the track robot, β. This angle is much less than the angle of attack of the tracked robot in the first step. The normal force of the rear wheel increases. The tracks' grousers behave like hooks on the steps.

The projected distance from the front of the bottom of the track to the nose of the first step, q_2, is given by:

$$q_2 = {(\Gamma - L\,sin\,\beta)}/{tan\,\beta} \qquad 4.12$$

Also, with reference to Figure 4.2.2, the distance from the robot's center of mass to the step contact point**,** R_{r2}, is given by:

$$R_{r2} = \sqrt{\left(L_C - {}^{q_2}/_{\cos\beta}\right)^2 + (L_C \sin\beta + h\cos\beta - \Gamma)^2} \qquad 4.13$$

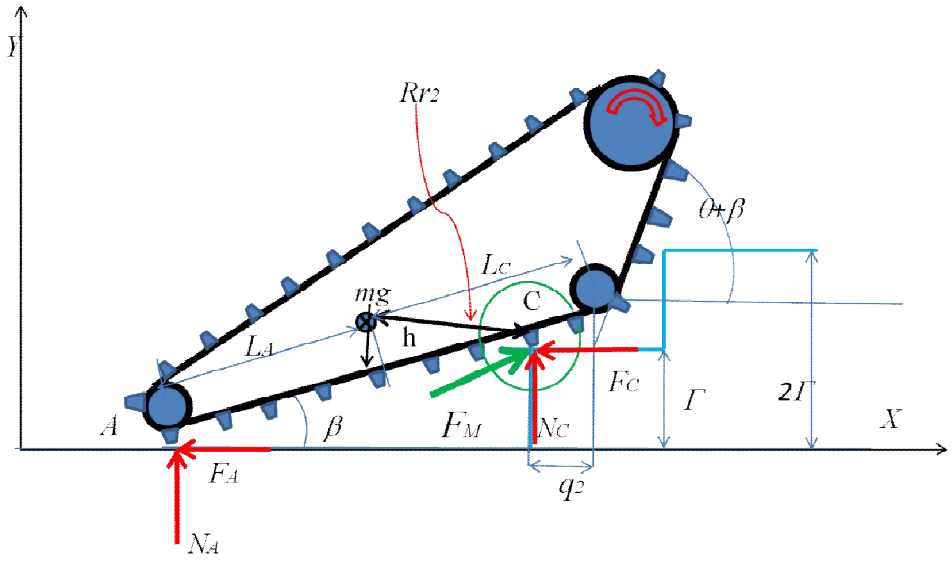


Figure 4.2.2 : Robots pose, as it pass the first step.

The equations of motions of the robot during this *Phase* are given by:

$$\sum F_x = m(a_{cm})_x = F_M \cos(\beta + \psi) + F_A - F_C \qquad 4.14$$

$$\sum F_y = m(a_{cm})_y = F_M \sin(\beta + \psi) + N_A + N_C - mg \qquad 4.15$$

$$\sum M_C = [I + (R_{r2})^2]\Omega = N_A(\mu\Gamma - L\cos\beta - q) - mg(L_C\cos\beta - q + h\sin\beta) \qquad 4.16$$

The acceleration of the center of mass with respect to the acceleration of the rear sprocket of the tracked robot is given by:

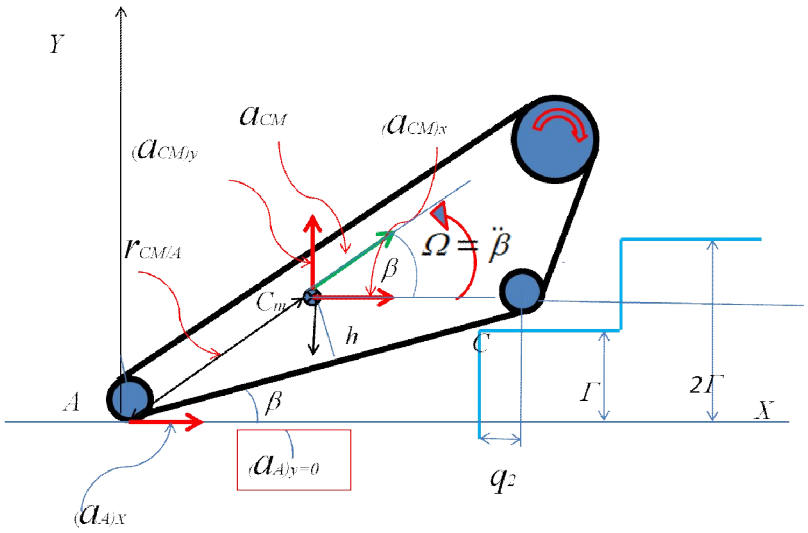


Figure 4.2.3 : Pertinent kinematic and geometry relations of the robot in *Phase* 2.

$$(a_{CM})_x i + (a_{CM})_y j = a_A i + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_A cos\,\beta - h\,sin\,\beta) & (L_A\,sin\,\beta + h\,cos\,\beta) & 0 \end{vmatrix} \qquad 4.\,17$$

As mentioned earlier, the angular velocity of the center of mass is neglected. The equations of acceleration in the *X* and *Y* directions are hence given by:

$$(a_{CM})_x = (a_A)_x - \ddot{\beta}(L_A\,sin\,\beta + h\,cos\,\beta) \qquad 4.\,18$$

$$(a_{CM})_y = \ddot{\beta}\,(L_A\,cos\,\beta - h\,sin\,\beta) \qquad 4.\,19$$

4.2.3 Dynamic Analysis of the Tracked Robot in *Phases* 3 and 4

In *Phases* 3 and 4, the bottom of the track is in contact with two or three steps noses as was discussed in Chapter 3. The robot is moving upward along the constant slope of the staircase γ. Therefore, the center of mass of the robot has two motions, in the *X* and *Y* directions.

Figure 4.2.4 and 4.2.5 show the configurations for the two stairs and three stairs contact respectively.

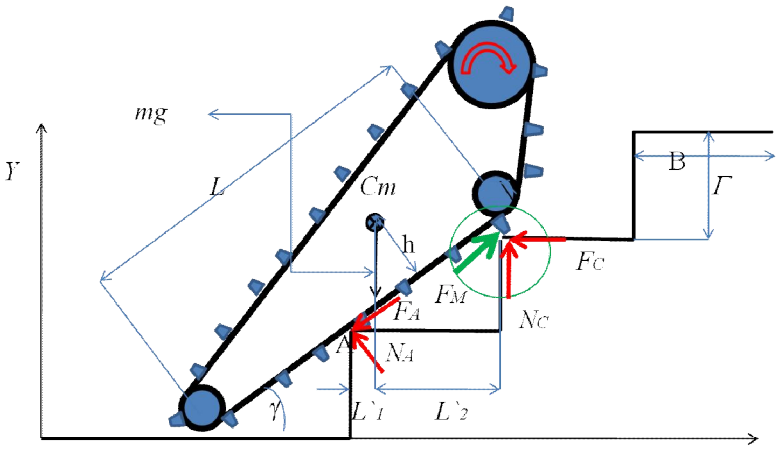


Figure 4.2.4 : Robot poses in *Phase* 3 while the track bottom is in contact with two step noses.

With reference to Figure 4.2.4, the equations of motion of the robot are given by:

$$B = \grave{L_1} + \grave{L_2} \qquad 4.\ 20$$

$$\sum F_x = m(a_{CM})_x = F_M \cos\gamma + F_A \cos\gamma - N_A \sin\gamma - F_C \qquad 4.\ 21$$

$$\sum F_y = m(a_{CM})_y = F_M \sin\gamma + F_A \sin\gamma + N_A \cos\gamma - mg + N_C \qquad 4.\ 22$$

$$\sum M_A = F_M[-(\Gamma\cos\gamma) + (B)\sin\gamma] + N_C[\Gamma\mu + (B)] - mg\grave{L_1} \qquad 4.\ 23$$

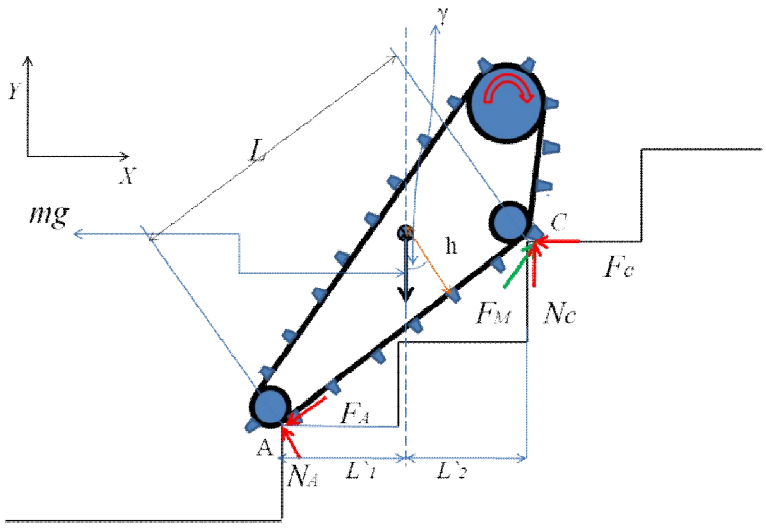


Figure 4.2.5 : Robot poses in *Phase* 4 while the track bottom is in contact with three step noses.

With reference to Figure 4.2.5, the equations of motions of the robot are given by:

$$2B = L_1^{`} + L_2^{`} \quad 4.24$$

$$\sum F_x = m(a_{CM})_x = F_M \cos\gamma + F_A \cos\gamma - N_A \sin\gamma - F_C \quad 4.25$$

$$\sum F_y = m(a_{CM})_y = F_M \sin\gamma + F_A \sin\gamma + N_A \cos\gamma - mg + N_C \quad 4.26$$

$$\sum M_A = N_C[2\Gamma\mu + (2B)\cos\gamma] - F_M[(2\Gamma\cos\gamma) - (2B)\sin\gamma] - mgL_1^{`} \quad 4.27$$

4.2.4 Dynamic Analysis of the Tracked Robot in *Phase* 5

With reference to Figures 3.11and 4.2.4 the equations of motion of the robot in *phase* 5 are similar to those in *Phase* 3.

4.2.5 Dynamic Analysis of the Tracked Robot in *Phase* 6

One of the most challenging motions of the robot is when it reaches the top of the stairs and its center of mass crosses its pivot point. This case is portrayed schematically in Figure 4.2.6. Any incremental forward motion causes the center of mass to pass the nose of the step. This causes a moment that tilts the front of the robot downwards as it continues its forward motion. The tracked robot's dynamic motion simultaneously moves the tracked robot forward. This phase ends when the tip of the flat bottom of the track contacts the landing of the next step

The acceleration of the tracked robot plays an important role in determining the time and the distance where the robot will collide with the level ground. A higher acceleration causes the distance between the last stair nose and the tracked robot's center of mass to increase.

The robot`s motion in this situation has two translation components, in the *X* and *Y* directions, and one rotation component.

With reference to Figure 4.2.6, the equations of motion of the robot are given by:

$$\sum F_x = m(a_{cm})_x = F_M \cos\beta - F_C \quad 4.28$$

$$\sum F_y = m(a_{cm})_y = F_M \sin\beta + N_C - mg \quad 4.29$$

$$\sum M_{cm} = I\Omega = -F_M \cos\beta\left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) + F_M \sin\beta\,\Delta L + F_C\left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) + N_C\Delta L \quad 4.30$$

The acceleration of the center of mass with respect to the pivot point is given by:

$$(a_{CM})_X i + (a_{CM})_y j = (a_c)_x i + (a_c)_y + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ \Delta L & \left(h/_{cos\,\beta} + \Delta L\, tan\,\beta\right) & 0 \end{vmatrix} \qquad 4.\ 31$$

$$(a_{CM})_x = (a_c)_x - \Omega\left(h/_{cos\,\beta} + \Delta L\, tan\,\beta\right) \qquad 4.\ 32$$

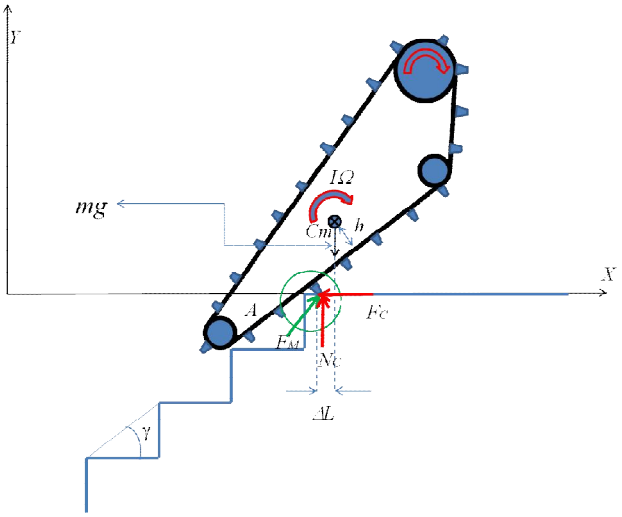


Figure 4.2.6 : Robot poses when the robots' center of mass crosses the pivot point at the top step.

4.2.6 Dynamic Analysis of the Tracked Robot in *Phases* 7 and 8

As shown in the Figures 3.12 (*Phases*7&8) and Figure 4.2.7, after the robot climbs the stairs and reaches the flat surface at the top, the weight of the robot is uniformly distributed along the grousers on the track bottom that are in contact with the flat ground. In *Phase* 7, the rear of the robot hangs over the top step in the climb, while in *Phase* 8, the front of the robot hangs over the top step of the descent. Thus, part of the robot's track is suspended above the stairs, but the center of gravity is always above the part of the track that is in contact with the ground. The robot motion in these phases is strictly along the *X* direction.

With reference to Figure 4.2.7, the equations of motion of the robot are given by:

$$Total\,R = \mu mg \qquad 4.\ 33$$

$$\sum F_x = m\ddot{x} = F_M - \mu\, mg \qquad 4.34$$

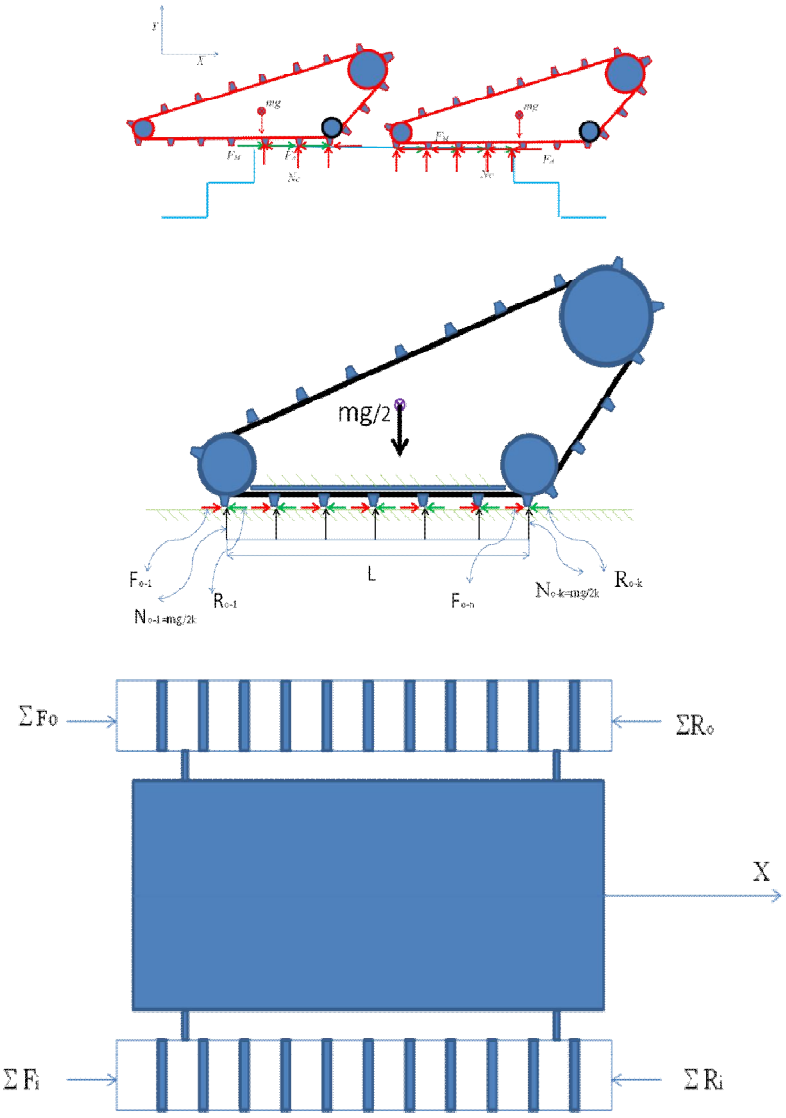


Figure 4.2.7 : Schematic of the robot poses in *Phases* 7 and 8.

4.2.7 Motion of the Tracked Robot in *Phase* 9

By far, this is the most critical phase in the robots' navigation of a staircase as it is the phase where the robot experiences the highest accelerations and impact loading.

As the robot's center of mass crosses the nose of the topmost step on its way to descend the staircase, the distance between the center of mass and the nose of the step multiplied by the robot's weight produces a torque. This causes the robot to tilt downwards towards the next step.

Its angular acceleration rapidly increases from its zero value on the flat surface. The magnitude of this increase depends on the robot's weight, velocity and linear acceleration. The increasing ΔL increases the torque, which in turn increases the angular acceleration. Under the assumption of no slip, the robot momentum will not cause an appreciable change in the contact point with the next step landing. The no slip assumption in this case is justified since the center of mass is nearer to the front of the track as has been concluded earlier, and there would be enough track length behind this point to provide frictional force as the robot starts to tip downward.

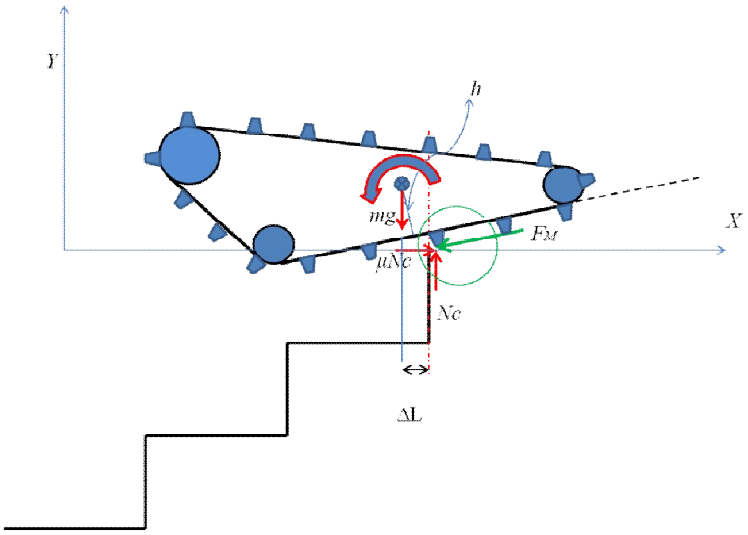


Figure 4.2.8 : Robot poses during the descending in *Phase* 9.

With reference to Figure 4.2.8, the equations of motion of the robot are given by:

$$\sum F_x = -F_M \cos\beta + F_C \qquad 4.35$$

$$\sum F_y = m(a_{cm})_y = -F_M \sin\beta - mg + N_C \qquad 4.36$$

$$\sum M_{cm} = I\Omega = -F_M \left[\sin\beta\, \Delta L + \left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right)\cos\beta\right] + N_C \Delta L + F_C \left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) \qquad 4.37$$

The acceleration of the center of mass with respect to the pivot point is given by:

$$(a_{CM})_x i + (a_{CM})_y j = (a_c)_x i + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ -\Delta L & \left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) & 0 \end{vmatrix} \qquad 4.38$$

$$(a_{CM})_x = (a_c)_x - \Omega\left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) \quad 4.39$$

$$(a_{CM})_x = \Omega\Delta L \quad 4.40$$

It should be noted that the minimum distance between the center of mass and the front of the track beyond which the robot may tip over when descending can be found as follows.

If:

$$L_M > h \quad 4.41$$

And;

$$L_M - L_S \geq \Gamma \quad 4.42$$

Under above condition the possibility of the tracked robots tipping over is low.

Where L_M is the front length of tracked robot and L_S the distance between front bottom wheel and center of mass in X direction (Figure 4.2.9)

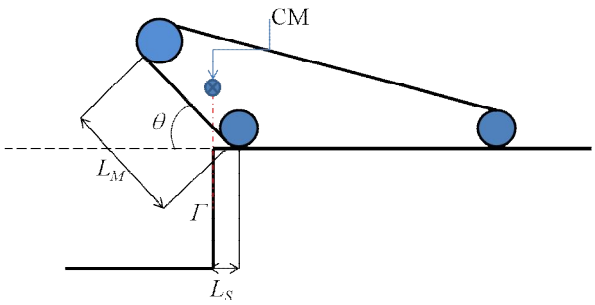


Figure 4.2.9 : Robot poses during the descending in *Phase* 9

4.2.8 Dynamic Analysis of the Tracked Robot in *Phase* 10

The angular acceleration of the center of mass of the robot, coupled with the linear accelerations at the instant of contact of the track with the next step, may cause the robot to topple over (i.e., the back of the track loses contact with the nose of the top step). The likelihood of this occurring increases as the center of mass moves closer to the front of the track. This is because the angle of inclination of the robot when it contacts the next step depends on where it contacts the next step landing. That angle becomes steeper as the center of mass is closer to the

front of the track, due to the earlier nose crossing of the center of mass. During this phase, after the tracks contact the landing of the next step, the robot crawls forward and its inclination angle decreases until it coincides with the staircase angle. This happens at the end of this phase when the front of the flat bottom of the track reaches the nose of the step.

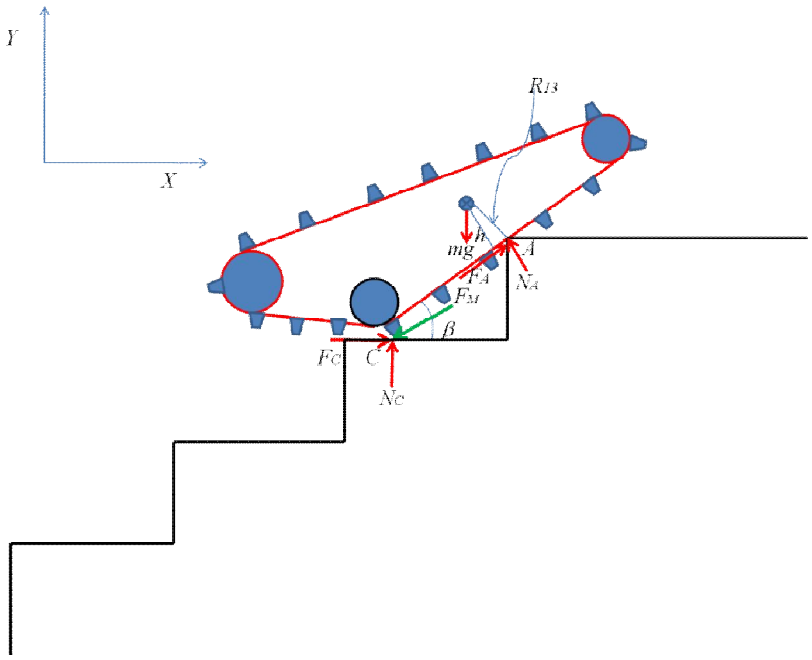


Figure 4.2.10 : Robot poses, while it tips over first step before reaching the second nose (*Phase*10).

With reference to Figure 4.2.10, the equations of motion of the robot are given by:

$$R_{r3} = \sqrt{h^2 + \left(\Gamma/_{\cos\beta} - L_C\right)^2} \qquad 4.\,43$$

$$\sum F_X = m\ddot{x} = F_C + F_A\cos\beta - N_A\sin\beta - F_M\cos\beta \qquad 4.\,44$$

$$\sum F_y = m\ddot{y} = N_c + N_A\cos\beta - F_A\sin\beta - F_M\sin\beta \qquad 4.\,45$$

$$\sum M_C = (I + m(R_{r3})^2)\,\Omega = N_A\Gamma\left[1/_{\tan\beta}(\mu\sin\beta - \cos\beta) - (\sin\beta + \mu\cos\beta)\right] + mg\left[\Gamma/_{\tan\beta} - \cos\beta(\Gamma\sin\beta - L_C) - h\sin\beta\right] \qquad 4.\,46$$

The acceleration of the center of mass with respect to the acceleration of the front of the tracked robot is given by:

$$(a_{CM})_x i + (a_{CM})_y j = a_A i + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_C \cos\beta - h\sin\beta) & (L_C \sin\beta + h\cos\beta) & 0 \end{vmatrix} -$$

$$\varpi^2\big((L_C \cos\beta - h\sin\beta)i + (L_C \sin\beta + h\cos\beta)j\big) \qquad 4.47$$

$$(a_{CM})_x = (a_A)_x - \ddot{\beta}(L_C \sin\beta + h\cos\beta) \qquad 4.48$$

$$(a_{CM})_y = \ddot{\beta}\,(L_C \cos\beta - h\sin\beta) \qquad 4.49$$

4.2.9 Dynamic Analysis of the Tracked Robot in *Phase* 11

In *Phases* 3, it was shown that the track flat bottom is in contact with two step noses. The robot is moving downward along the constant slope of γ. Therefore; the center of mass of the robot has two motions, in the *X* and *Y* directions. Figure 4.2.11 shows the robot's pose when the robot contacts two steps.

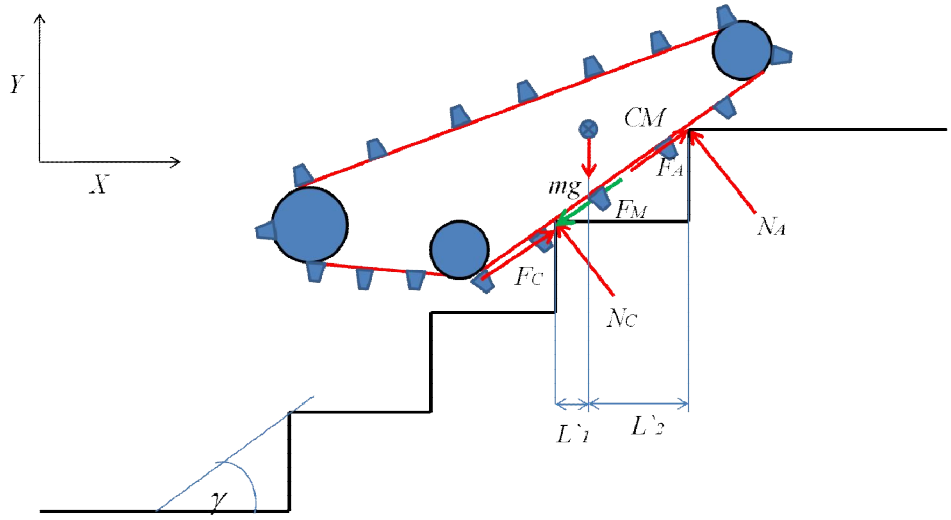


Figure 4.2.11 : Robot poses, during the descending in (Tracked Robot has two contact with stairs noses).

With reference to Figure 4.9, the equations of motions of the robot are given by:

$$\sum F_x = m(a_{CM})_x = \cos(\gamma)\,(-F_M - F_A + F_C) - \sin(\gamma)\,(N_A + N_C) \qquad 4.50$$

$$\sum F_y = m(a_{CM})_Y = sin(\gamma)\,(-F_M + N_A + N_C) + cos(\gamma)\,(N_A + N_C) - mg = 0 \qquad 4.51$$

$$\sum M_A = F_M[(2\Gamma cos\,\gamma) - (B)\,sin\,\gamma] - mgL_2^{`} + N_C[2\Gamma(sin\,\gamma - \mu cos\,\gamma) + (B)(cos\,\gamma + \mu sin\,\gamma)] \qquad 4.52$$

4.2.10 Dynamic Analysis of the Tracked Robot in *Phase* 12

This phase starts when the center of mass crosses the nose of the lower step. The robot tips while pivoting on the nose of the lower step and contacts the landing of the next step, similar to what happens in *Phase* 9.

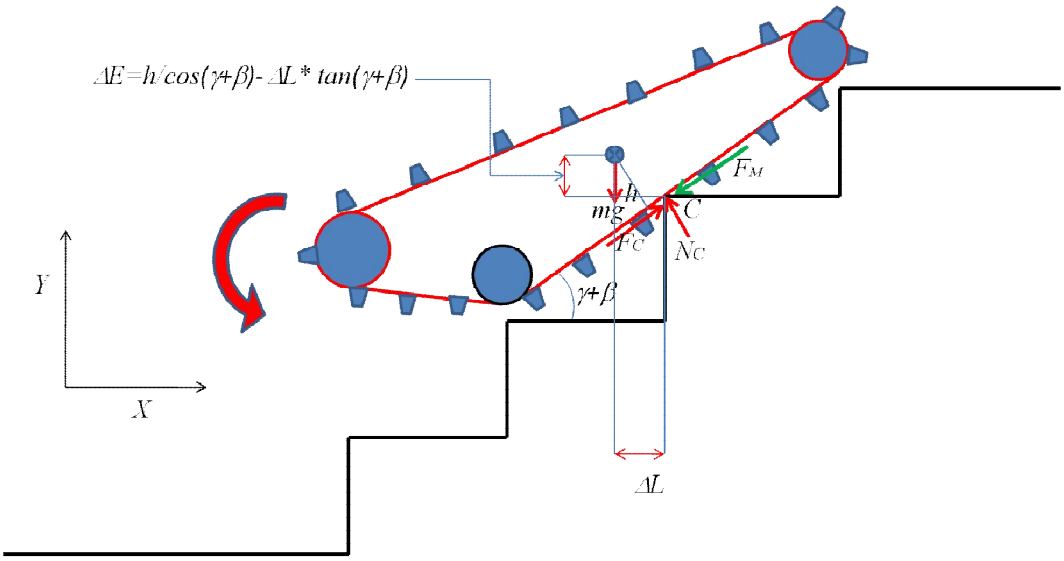


Figure 4.2.12 : Robot poses, while it tips over the step and before reaching the following nose (*Phase*12).

With reference to Figure 4.2.12, the equations of motion of the robot are given by:

$$\Delta E = {}^{h}/_{cos(\beta)} - \Delta L\,tan(\beta) \qquad 4.53$$

$$\sum F_x = m(a_{cm})_y = -F_M\,cos(\beta) + F_C\,cos(\beta) - N_C\,sin(\beta) \qquad 4.54$$

$$\sum F_y = m(a_{cm})_y = -F_M\,sin(\beta) + F_C\,sin(\beta) + -mg + N_C\,cos(\beta) \qquad 4.55$$

$$\sum M_{cm} = I\Omega = -F_M[sin(\beta)\,\Delta L + cos(\beta)\,\Delta E] + N_C\{[-sin(\beta)\,\Delta L + \mu cos(\beta)]\Delta L + \Delta E[\mu sin(\beta)\,\Delta L + cos(\beta)]\}$$

4.56

The acceleration of the center of mass with respect to the pivot point is given by:

$$(a_{CM})_x i + (a_{CM})_y j = (a_c)_x i + (a_c)_y j + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ -\Delta L & \Delta E & 0 \end{vmatrix} \qquad 4.57$$

$$(a_{CM})_x = (a_c)_x - \Omega\Delta E \qquad 4.58$$

$$(a_{CM})_y = (a_c)_y + \Omega\Delta L \qquad 4.59$$

Phases 10, 11, and 12 repeat alternately untill the robot reaches the bottom most step.

4.2.11 Dynamic Analysis of the Tracked Robot in *Phase* 13

With reference to Figure 4.2.10, the motion of the tracked robot in *Phase* 13 is similar to that of *Phase* 10.

4.2.12 Dynamic Analysis of the Tracked Robot in *Phase* 14

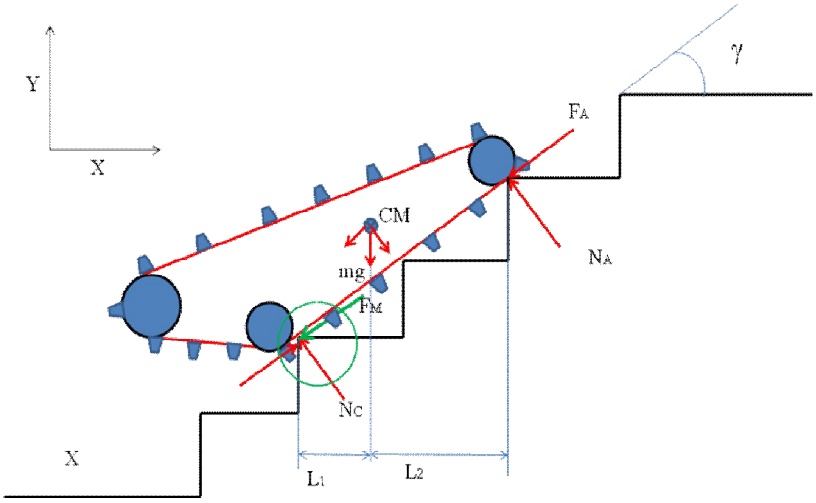


Figure 4.2.13 : Robot poses, during the descending in (Tracked Robot has three contact with stairs noses).

With reference to Figure 4.2.13, the equations of motions of the robot are given by:

$$\sum F_x = m(a_{CM})_x = cos(\gamma)\,(-F_M - F_A + F_C) - sin(\gamma)\,(N_A + N_C) \qquad 4.\,60$$

$$\sum F_y = m(a_{CM})_Y = sin(\gamma)\,(-F_M + N_A + N_C) + cos(\gamma)\,(N_A + N_C) - mg \qquad 4.\,61$$

Taking the moment with respect to point *A* yields:

$$\sum M_A = F_M[(2\Gamma\, cos\,\gamma) - 2B\, sin\,\gamma] - mgL_2^{`} + N_C[2\Gamma(sin\,\gamma - \mu\, cos\,\gamma) + 2B(cos\,\gamma + \mu\, sin\,\gamma)] \qquad 4.\,62$$

4.2.13 Dynamic Analysis of the Tracked Robot in *Phase* 15

With reference to Figure 4.2.11, the motion of the tracked robot in *Phase* 15 is similar to that of *Phase* 11.

4.2.14 Dynamic Analysis of the Tracked Robot in *Phase* 16

With reference to Figure 4.2.12, the motion of the tracked robot in *Phase* 16 is similar to that of *Phase* 12.

4.2.15 Dynamic Analysis of the Tracked Robot in Phase 17

When the front of the robot tracks contact the bottom landing the robot crawls forward and its inclination angle changes from one that is steeper than that of the staircase to an angle given by $\tan^{-1}(\Gamma/2L)$ when the back of the track flat bottom contacts the nose of the lowest step.

The robot's center of mass in this phase has three different motions, in the *X* and *Y* directions and a rotation.

With reference to Figure 4.2.14, the equations of motions of the robot are given by:

$$R_{r3} = \sqrt{h^2 + \left(\Gamma/_{cos\,\beta} - L_C\right)^2} \qquad 4.\,63$$

$$\sum F_X = m\ddot{x} = F_C - F_A\, cos\,\beta - N_A\, sin\,\beta - F_M\, cos\,\beta \qquad 4.\,64$$

$$\sum F_y = m\ddot{y} = N_c + N_A \cos\beta - F_A \sin\beta - F_M \sin\beta \qquad 4.65$$

$$\sum M_C = (I + m(R_{r3})^2)\,\Omega = N_A\Gamma\left[{}^{1}/_{\tan\beta}\,(\mu\sin\beta - \cos\beta) - (\sin\beta + \mu\cos\beta)\right] +$$

$$mg\left[{}^{\Gamma}/_{\tan\beta} - \cos\beta\,(\Gamma\sin\beta - L_C) - h\sin\beta\right] \qquad 4.66$$

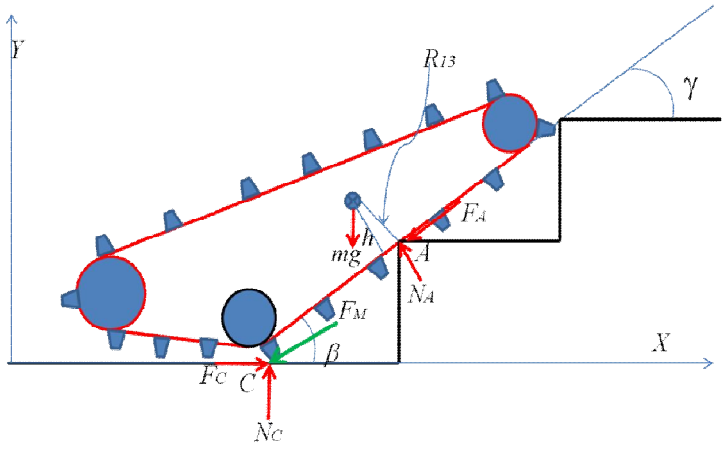


Figure 4.2.14 : Motion of the tracked robot in Phase17.

The acceleration of the center of mass with respect to the acceleration of the front of the tracked robot is given by:

$$(a_{CM})_x i + (a_{CM})_y j = a_A i + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_C\cos\beta - h\sin\beta) & (L_C\sin\beta + h\cos\beta) & 0 \end{vmatrix} -$$

$$\varpi^2\big((L_C\cos\beta - h\sin\beta)i + (L_C\sin\beta + h\cos\beta)j\big) \qquad 4.67$$

$$(a_{CM})_x = (a_A)_x - \ddot{\beta}(L_C\sin\beta + h\cos\beta) \qquad 4.68$$

$$(a_{CM})_y = \ddot{\beta}\,(L_C\cos\beta - h\sin\beta) \qquad 4.69$$

4.2.16 Dynamic Analysis of the Tracked Robot in *Phase* 18

When the back of the flat bottom of the track clears the nose of the bottom step, the robot back falls down on the bottom landing and lies flat on it.

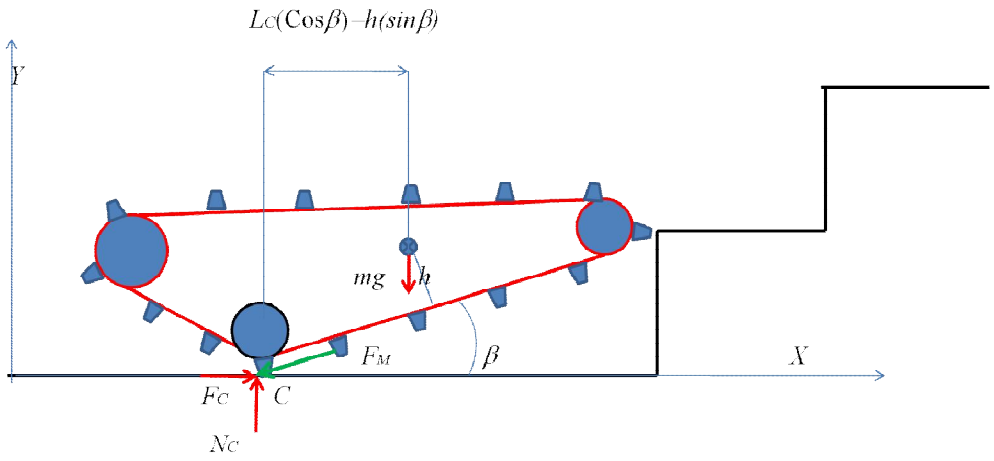


Figure 4.2.15 : Motion of the tracked robot in *Phase*18.

With reference to Figure 4.2.15, the equations of motion of the robot are given by:

$$\sum F_x = m(a_{cm})_x = F_M \cos(\beta) - F_C \qquad 4.70$$

$$\sum F_y = m(a_{cm})_y = F_M \sin(\beta) + N_C - mg \qquad 4.71$$

$$\sum M_{Cm} = I\Omega = F_M\{\cos(\beta)\,[(L_C \cos\beta - h\sin\beta)] - \sin\beta\,(L_C \sin\beta + h\cos\beta)\} + N_C\{(L_C \cos\beta - h\sin\beta) - \mu(L_C \sin\beta + h\cos\beta)\} \qquad 4.72$$

The acceleration of the center of mass with respect to the acceleration of the front of the tracked robot is given by:

$$(a_{CM})_x i + (a_{CM})_y j = (a_c)_x i + (a_c)_y + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_C \cos\beta - h\sin\beta) & (L_C \sin\beta - h\cos\beta) & 0 \end{vmatrix} \qquad 4.73$$

$$(a_{CM})_x = (a_c)_x - \Omega[(L_C \sin\beta - h\cos\beta)] \qquad 4.74$$

$$(a_{CM})_y = \Omega(L_C \cos\beta - h\sin\beta) \qquad 4.75$$

The impact force on the back roller can be calculated as follows:

$$M_C = (L_C \cos\beta - h \sin\beta) mg \qquad 4.76$$

4.3 Case of Tracks with no Grousers (Flat Track)

For the case of flat tracks with no grousers, the equations of motion of the robot differ enough from the case with grousers such that it affects the performance of the robot particularly when climbing stairs, and to a lesser degree when descending stairs. The performance difference on flat ground is not substancial, and is more apparent on loose or compliant terrain like sand or on carpets.

The dynamic equations derivations are provided in Aappendix C. Table 4.1 below lists the equations for the cases of grousers and no grousers side by side for comparative purposes. The table highylights the fact of similarity of the equations during descend. The table also highlights the importance of the grousers in assisting the robot to climb stairs.

Table 4.1: Illustration of the differences between the equations of Tracked Robot with Grousers (TRG) and the Tracked Robot with Flat Belt (TRFB) during18 Phases.

The deference of forces of Tracked Robot with Grousers (TRG) VS Flat Belt (TRFB) in *phase* 1.		
	TRG	TRFB
F_X	$F_M\,cos(\theta + \beta + \psi) + F_A - F_C$	$F_M\,cos(\theta + \beta) + F_A - F_C\,cos(\theta + \beta) - N_C\,sin(\theta + \beta)$
F_y	$F_M\,sin(\theta + \beta + \psi) + N_A + N_C - mg$	$F_M\,sin(\theta + \beta) + N_A - F_C\,sin(\theta + \beta) + N_C\,cos(\theta + \beta) - mg$
M_C	$(I + mR_{r1}{}^2)\Omega = N_A(\mu\Gamma - L\,cos\,\beta - q\ ) + (L_C\,cos\,\beta + q)mg$	$(I + mR_{r1}{}^2)\Omega = N_A(\mu\Gamma - L\,cos\,\beta - q\ ) + (L_C\,cos\,\beta + q)mg$

The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase* 2.		
	TRG	TRFB
F_X	$F_M\,cos(\beta + \psi) + F_A - F_C$	$F_M\,cos\,\beta + F_A\ - F_C\,cos\,\beta - N_C\,sin\,\beta$
F_y	$F_M\,sin(\beta + \psi) + N_A + N_C - mg$	$F_M\,sin\,\beta + N_A - F_C\,sin\,\beta\ + N_C\,cos\,\beta - mg$
M_C	$[I + (R_{r2})^2]\Omega = N_A\,(\mu\Gamma - L\,cos\,\beta - q) + mg(L_C\,cos\,\beta - q + h\,sin\,\beta)$	$[I + m(R_{r2})^2] = N_A\,(\mu\Gamma - L\,cos\,\beta - q) + mg(L_C\,cos\,\beta - q + h\,sin\,\beta)$

The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase3*.		
	TRG	TRFB
F_X	$F_M\,cos\,\gamma + F_A\,cos\,\gamma - N_A\,sin\,\gamma - F_C$	$F_M\,cos\,\gamma + F_A\,cos\,\gamma - N_A\,sin\,\gamma - F_C\,cos\,\gamma - N_c\,sin\,\gamma$
F_y	$F_M\,sin\,\gamma + F_A\,sin\,\gamma + N_A\,cos\,\gamma + N_C - mg$	$F_M\,sin\,\gamma + F_A\,sin\,\gamma + N_A\,cos\,\gamma - F_C\,sin\,\gamma + N_C\,cos\,\gamma - mg$
M_C	$F_M[-(\Gamma cos\,\gamma) + (\beta)\,sin\,\gamma] + N_C[\Gamma\mu + (\beta)] - mgL_1^{`}$	$F_M[-(\Gamma cos\,\gamma) + (\beta)\,sin\,\gamma] + N_C\{[cos\,\gamma\,(\Gamma\mu + (\beta))] + [\mu\,sin\,\gamma\,((\beta) - \Gamma)]\} - mgL_1^{`}$

The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*4.		
	TRG	TRFB
F_X	$F_M \cos\gamma + F_A \cos\gamma - N_A \sin\gamma - F_C$	$F_M \cos\gamma + F_A \cos\gamma - N_A \sin\gamma - N_c \sin\gamma - F_C \cos\gamma$
F_y	$F_M \sin\gamma + F_A \sin\gamma + N_A \cos\gamma + N_C - mg$	$F_M \sin\gamma + F_A \sin\gamma + N_A \cos\gamma + N_C \cos\gamma - F_C \sin\gamma - mg$
M_A	$F_M[-(2\Gamma\cos\gamma) + (2B)\sin\gamma] + N_C\,[2\Gamma\mu + (2B)\cos\gamma] - mgL_1^{`}$	$F_M[-(2\Gamma\cos\gamma) + (2B)\sin\gamma] + N_C\{[\cos\gamma\,(2\Gamma\mu + (2B))] + [\mu\sin\gamma\,((2B) - 2\Gamma)]\} - mgL_1^{`}$
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*5.		
The equations are the same as equations *Phase* 3		
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*6.		
	TRG	TRFB
F_X	$F_M \cos\beta - F_C$	$F_M \cos\beta - F_C \cos\beta - N_C \sin\beta$
F_y	$F_M \sin\beta + N_C - mg$	$F_M \sin\beta + N_C \cos\beta - F_C \sin\beta - mg$
M_{CM}	$I\Omega = -F_M \cos\beta\left(\frac{h}{\cos\beta} + \Delta L \tan\beta\right) + F_M \sin\beta\,\Delta L + N_C\Delta L + F_C\left(\frac{h}{\cos\beta} + \Delta L \tan\beta\right)$	$I\Omega = -F_M \cos\beta\,\Delta L + F_M \sin\beta\left(\frac{h}{\cos\beta} + \Delta L \tan\beta\right) + N_C \sin\beta + N_C \cos\beta\left(\frac{h}{\cos\beta} + \Delta L \tan\beta\right)$
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*7and8.		
	TRG	TRFB
F_X	$F_M - mg \times \mu$	$F_M - mg \times \mu$
F_y	----------	----------
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*9		

	TRG	TRFB
F_X	$-F_M \cos\beta + F_C$	$-F_M \cos\beta + F_C \cos\beta - N_C \sin\beta$
F_y	$-F_M \sin\beta + N_C - mg$	$-F_M \sin\beta + N_C \cos\beta + F_C \sin\beta - mg$
M_{CM}	$I\Omega = -F_M\left[+\left(\frac{h}{\cos\beta} + \Delta L \tan\beta\right)\cos\beta + \sin\beta\Delta L\right] + N_C\Delta L + F_C\left(\frac{h}{\cos\beta} + \Delta L \tan\beta\right)$	$I\Omega = -F_M\left[\left(\frac{h}{\cos\beta} + \Delta L \tan\beta\right)\cos\beta + \sin\beta\,\Delta L\right] + N_C\left[\cos\beta\Delta L - \sin\beta\left(\frac{h}{\cos\beta} + \Delta L \tan\beta\right)\right] + F_C\left[\cos\beta\left(\frac{h}{\cos\beta} + \Delta L \tan\beta\right) + \sin\beta\,\Delta L\right]$
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*10		
	TRG	TRFB
F_X	$F_C + F_A \cos\beta - N_A \sin\beta - F_M \cos\beta$	
F_y	$N_c + N_A \cos\beta - F_A \sin\beta - F_M \sin\beta$	The same
M_C	$N_A\Gamma\left[\frac{1}{\tan\beta}(\mu\sin\beta - \cos\beta) - (\sin\beta + \mu\cos\beta)\right] +$ mg $\left[\frac{\Gamma}{\tan\beta} - \cos\beta(\Gamma\sin\beta - L_C)h\sin\beta\right]$	
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*11.		
F_X	$-F_M\cos\gamma + \cos\gamma\,(-F_A + F_C) - \sin\gamma\,(N_A + N_C)$	
F_y	$-F_M \sin\gamma + \sin\gamma\,(-F_A + F_C) + \cos\gamma\,(N_A + N_C) - mg$	The same
	$F_M[(\Gamma\cos\gamma) - (B)\sin\gamma] + N_C[\Gamma(\sin\gamma - \mu\cos\gamma) + (B)(\cos\gamma + \mu\sin\gamma)] - mgL_2^{`}$	
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*12.		
	TRG	TRFB

F_X	$-F_M \cos(\beta) + F_C\, cos(\beta) - N_C\, sin(\beta)$	
F_y	$-F_M\, sin(\beta) + F_C\, sin(\beta) + N_C\, cos(\beta) - mg$	The same
M_{CM}	$-\mathrm{F_M}[\sin(\beta)\,\Delta\mathrm{L} + \cos(\beta)\,\Delta E] + N_C\{[-sin(\gamma + \beta)\,\Delta L + \mu\, cos(\beta)]\Delta L + \Delta E[\mu\, sin(\beta)\,\Delta L + cos(\beta)]\}$	
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*13.		
	TRG	TRFB
F_X	$-F_M \cos\beta + F_A\, cos\,\beta - N_A\, sin\,\beta + F_C$	
F_y	$-F_M \sin\beta + N_A\, cos\,\beta - F_A\, sin\,\beta + N_C$	The same
M_C	$\mathrm{N_A}\Gamma\left[\frac{1}{\tan\beta}(\mu \sin\beta - \cos\beta) - (\sin\beta + \mu\cos\beta)\right] + mg\left[\frac{\Gamma}{tan\,\beta} - cos\,\beta\,(\Gamma\, sin\,\beta - L_C) - h\, sin\,\beta\right]$	
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*14.		
	TRG	TRFB
F_X	$-F_M cos\,\gamma + cos\,\gamma\,(-F_A + F_C) - sin\,\gamma\,(N_A + N_C)$	
F_y	$-F_M\, sin\,\gamma + sin\,\gamma(-F_A + F_C) + cos\,\gamma\,(N_A + N_C) - mg$	The same
M_A	$F_M[(2\Gamma cos\,\gamma) - (2B)\, sin\,\gamma] + N_C[2\Gamma(sin\,\gamma - \mu\, cos\,\gamma) + (2B)(cos\,\gamma + \mu\, sin\,\gamma)] - mgL_2^{`}$	
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*15.		
The equations are the same as equations *Phase* 11		
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase*16.		
The equations are the same as equations *Phase* 12		

The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase* 17.		
	TRG	TRFB
F_X	$-F_M \cos\beta + F_C - F_A \cos\beta - N_A \sin\beta$	
F_y	$-F_M \sin\beta + N_c - F_A \sin\beta + N_A \cos\beta$	The same
M_C	$(I + m(R_{r3})^2)\Omega = N_A \Gamma \left[\frac{1}{\tan\beta}(\mu \sin\beta - \cos\beta) - (\sin\beta + \mu\cos\beta)\right] + mg\left[\frac{\Gamma}{\tan\beta} - \cos\beta(\Gamma \sin\beta - L_C) - h\sin\beta\right]$	
The difference of Forces of the Tracked Robot with Grousers (TRG) VS a Flat Belt Robot (TRFB) in *Phase* 18.		
	TRG	TRFB
F_X	$-F_M \cos\beta + F_C$	
F_y	$-F_M \sin(\beta) + N_C - mg$	The same
M_{CM}	$I\Omega = F_M\{\cos(\beta)[(L_C \cos\beta - h\sin\beta)] - \sin\beta(L_C \sin\beta + h\cos\beta)\} + N_C\{(L_C \cos\beta - h\sin\beta) - \mu(L_C \sin\beta + h\cos\beta)\}$	

Chapter 5

5.1 Simulation

5.1.1 Introduction

This chapter presents results from simulation and its results of the tracked robot's motion from the start of stair climb through the end of the stairs descend till it reaches level ground. This activity is broken into eighteen phases.

The dynamic equations are taken from Chapter 4. All of the assumptions from the previous chapters were used for the equations. The simulations use the Euler method with error correction for solving the differential equations. The electric motor dynamics (torque, angular acceleration, and angular velocity) where modeled using the method proposed in "A Comprehensible Guide to Servo Motor Sizing", by Wilfried Voss [38] and the "maxon DC motor and maxon EC motor, Key information" document [39].

To apply simulation, the motor's behavior has to be examined. The relation between the motor voltage, angular velocity, torque, angular acceleration, angular velocity, and acceleration time must be determined.

5.1.2 Voltage – angular velocity line of the brushed DC motor

The relation between the applied voltage and the generated angular velocity for brushed DC motors is constant, with the voltage or speed constant being labeled as k_V. Figure 5.1.1 shows the relation between different applied voltages and their generated angular velocities [38].
The related formula is:

$$\frac{\omega_{correcte\ no\ load}}{\omega_{no\ load,rated}} = \frac{V_{applied}}{V_{rated}} = K_v \qquad 5.1$$

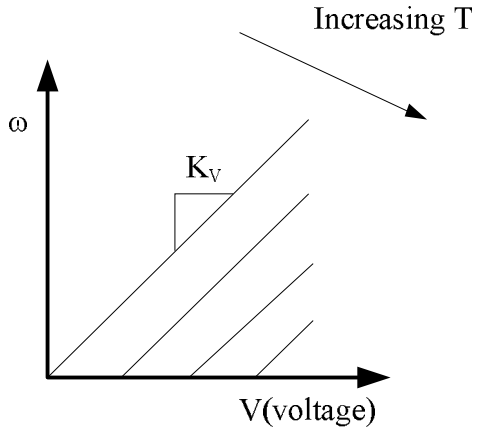


Figure 5.1.1 : DC Motor Angular Speed, (ω), Motor Voltage and Voltage Constant (K_v).

5.1.3 Speed-torque line

The DC motor's torque vs. angular speed relationship for a given constant voltage is shown in Figure 5.1.2. Speed decreases linearly with increasing torque. Figures 5.1.1 and 5.1.2 also show that as the voltage changes, the slope of ω vs. T remains constant [39].

The relationship between torque and angular velocity is given as follows:

$$T_{corrected\ Stall} / \omega_{Corrected\ noload} = -1 / Slope_{\omega,T} \qquad 5.2$$

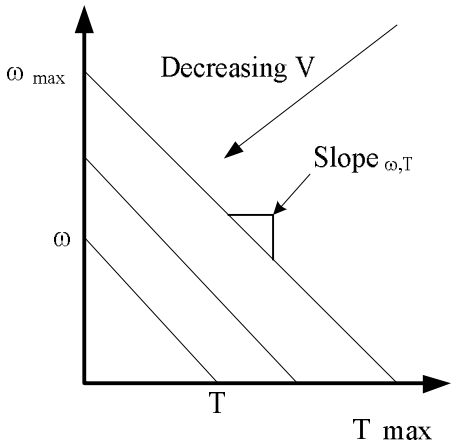


Figure 5.1.2 : The relation between the DC motor's maximum angular velocity and its maximum torque (T_{stall}).

5.1.4 Acceleration Time

The time that it takes for the angular velocity to start from zero and reach its maximum value is defined as the acceleration time (t_a).

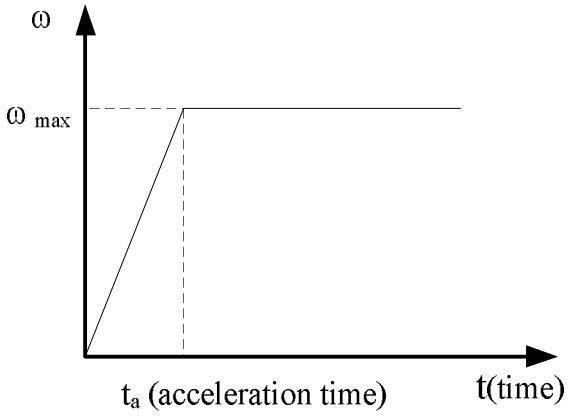


Figure 5.1.3 : The time that the angular velocity reaches its maximum is called Acceleration Time [39].

5.1.5 Angular velocity and Angular acceleration

With reference to Figures 5.1.3 and 5.1.4, the relationship between the angular acceleration and the angular velocity may be approximated as follows:

$$\alpha_{noload} = {}^{\omega_{noload}}/_{t_a} \qquad 5.3$$

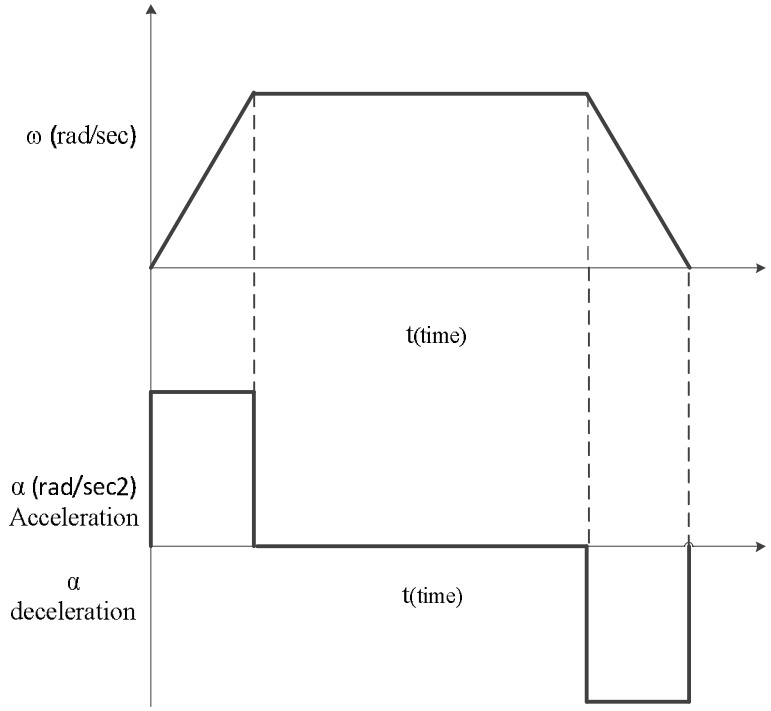


Figure 5.1.4 : The relation between angular acceleration and angular velocity [39].

5.1.6 Simulation Flow Charts

The following two flow charts provide a brief overview of the simulation of the tracked robot as it navigates up and down a staircase.

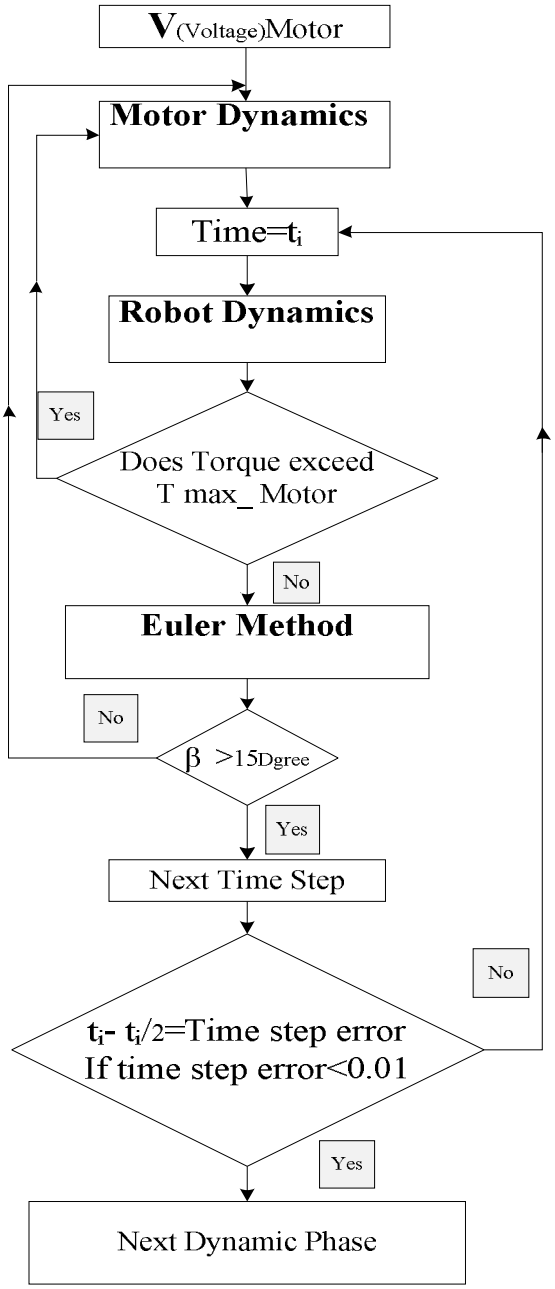


Figure 5.1.5 : Summarized Robots simulation charts

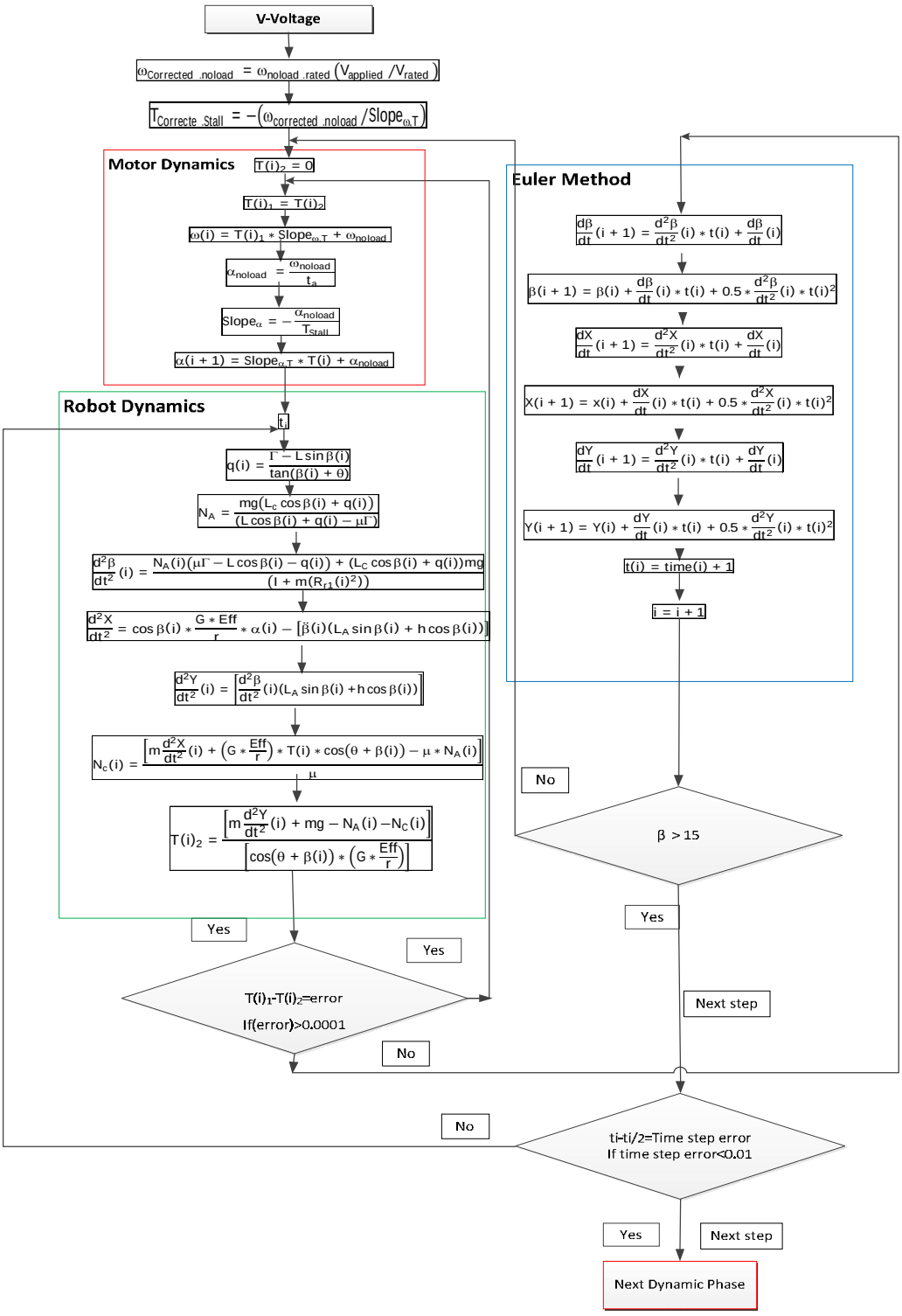


Figure 5.1.1 : Tracked Robots simulation charts (Detail)

Table 5.1 : Below Provides a listing of the parameters values used in the simulation.

Required Data for Simulation				
	Name	Symbol	Unit	Magnitude
1	DC Motor Max Voltage	V	Volt	30
2	DC Motor No load Speed	ω_no load	RPM	3380
3	DC Motor T Max	T-Stall	Newton meter N-m	1.13
4	Gear Box Ratio	G	-	1/60
5	Radius of Driven Pulley	r	Meter	0.1
6	Efficiency of Gear Box	Eff	-	0.55
7	Coefficient of friction(static)	μ	-	0.6
8	Weight of Half of Tracked Robot	mg	Newton- N	440
9	Mass Tracked Robot	m	kg	44
10	Rise of step	Γ	Meter	0.17
11	Tracked Robots` Angle of Attack	θ	Rad	$45 \times 3.14/180$
12	Angle of Stairs	γ	Rad	$30 \times 3.14/180$
13	Contact Length to Level Ground	L	Meter-	0.65
14	Height of center of mass to Level ground	h	Meter	0.065
15	Moment of Inertia	I	$kg * m^2$	44
16	Acceleration time	t_a	sec	0.2

5.1.7 Simulation analysis conditions

During the simulation run, it is assumed that the there is no track slippage on the stairs, and that the speeds of the two tracks are coordinated such that the robot navigates the stairs along a straight line with no yaw motion.

Due to the absence of yaw, the robot motion has three acceleration components that must be determined in each phase: one in the X direction, one in the Y direction, and an angular acceleration about an axis parallel to the Z axis. Since there is no slippage, the three acceleration

components are related to each other, and to the center of mass, in a strictly a geometrical manner, which is set by the robot and stair geometries

The conditions listed above hold for most of the phases analyzed below. In a few phases, particularly during the descend, the robot will tip and bang on the next step down or the staircase landing at the end of the descend. The accelerations in these cases will depend on other factors like the configurations of the staircase and the robot in addition to the geometry. The analysis of the resulting accelerations and the resulting forces at the instant of impact are beyond the scope of this thesis.

5.1.8 Simulation analysis of the tracked robot during *Phase* 1

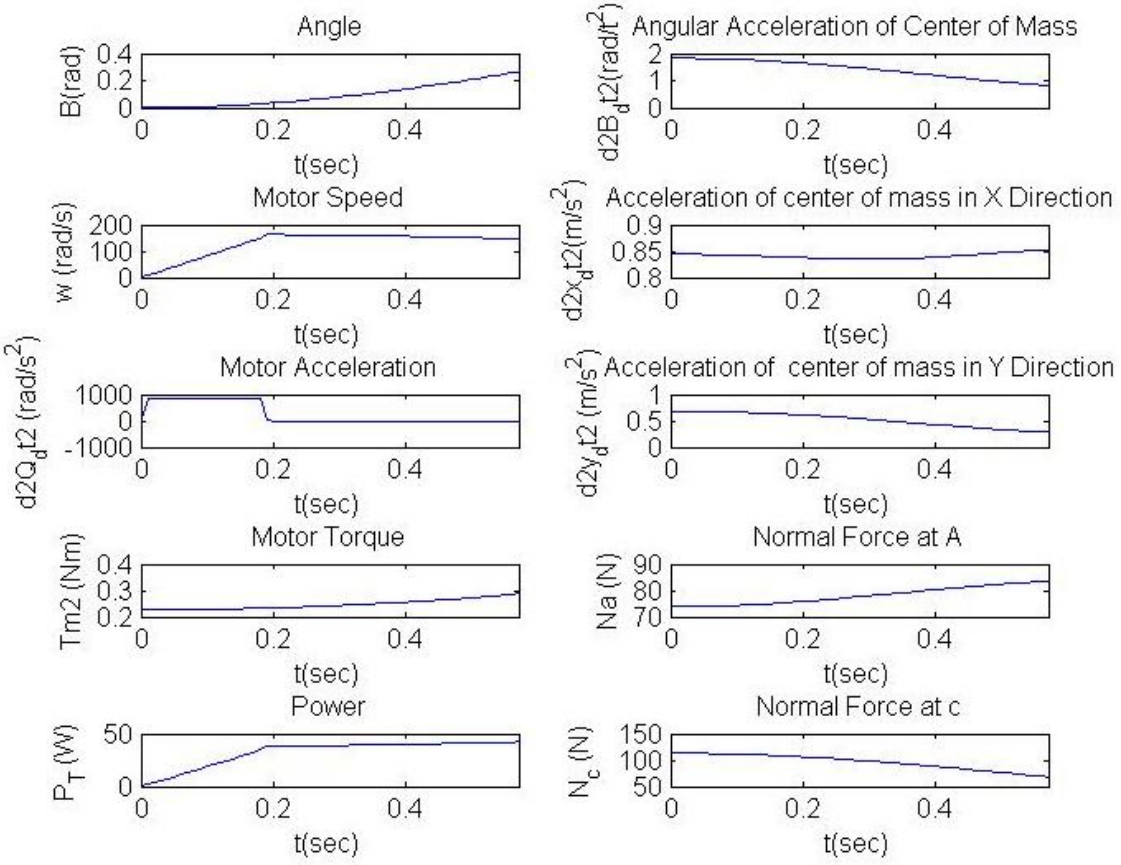


Figure 5.1.2 : Simulation of the Tracked Robot during *Phase* 1.

The robot is motionless at the start of the stair climbing (angular velocity and motor speed start are both 0).

As the front part of the robot touches the first step and starts to climb it, the following behavior occurs:

The angle of the bottom of the tracked robot with respect to the level ground, β, continuously increases. The required motor torque (T_{m2}) to overcome the tracked robot's weight and move it upward is 0.22 N-m. The magnitude of the torque slightly increases to 0.29 N-m at the end of *Phase* 1.

The weight of the robot tends to the rear wheel; therefore the normal reaction force increases at Point *A* and decreases at Point *C* (Figure 3.9). The angular acceleration of the tracked robot is related to the motor's torque. As the torque suddenly increases, the acceleration rapidly increases and then gently starts to decrease. The acceleration of the robot's center of mass in the *Y* direction is directly related to its angular acceleration; therefore, it instantaneously increases at the start of *Phase* 1 and then gently decreases. The acceleration of the center of mass in the *X* direction is also related to its angular acceleration center of mass as well as robot's angular acceleration (Eq4.9). Therefore it instantaneously increases and then remains approximately constant.

The angular velocity of the DC motor increases from zero to a maximum value of 180 rad/s in 0.2 seconds, and slightly decreases during the rest of *Phase* 1.

By increasing angular velocity, the motor's angular acceleration initially increases, and then remains constant for just under 0.2 seconds. While the motor's angular velocity reaches a constant value, its angular acceleration then drops to 0.

The required motor power depends on the motor's torque and its angular velocity. The change of torque is very small compared to the change of angular velocity at the beginning of the phase. Therefore, the power rapidly increases to overcome the inertia of the tracked robot. The slope of the power declines after 0.2 seconds.

5.1.9 Simulation analysis of the tracked robot during *Phase* 2

As the tracked robot continues to climb the first step, the following behavior occurs:

While the tracked robot transitions to *Phase* 2, the angle of attack changes from $\theta+\beta$ to β (Figure 4.2.2). This results in a change of the angle and magnitude of the tangential force caused by the motor torque. The required torque suddenly decreases to 0.26 N-m. While the tracked robot continues to move upward, the applied torque slightly increases.

The angle of the robot with respect to level ground, β, continuously increases until the front bottom part of the robot reaches the second steps nose.

At the start o*f Phase* 2, the power increases due to the spike in the angular velocity (the change of torque is very small at this stage). As the angular velocity starts to gently decline, the torque increases, and the power consumption also increases (under the effect of increasing torque).

The center of mass gradually moves to where it is directly above the first step nose and then passes beyond it. The load transfers from Point *A* to Point *C* during this transition. Consequently, the front normal force (N_C) increases and the rear normal force (N_A) decrease.

Because of the reduced torque at the beginning of the phase, there is a sudden decrease in the robot's angular acceleration at the beginning of the phase, but the acceleration gradually increases during the remainder of the phase as the torque increases. The robot's acceleration in the *Y* direction depends on the motor's torque and angular acceleration, as well as the robot's center of mass angular acceleration. These factors are all increasing or remaining constant, and the *Y* acceleration thus increases .The angular acceleration of the motor instantaneously increases for short period of time (0.2 seconds). It then drops to near zero because the change of ω is very small. The DC motor has a limited amount of torque. Most of the torque is used for moving the robot upwards (the robot's angular acceleration and *Y* acceleration are both increasing). Thus, the acceleration in the *X* direction decreases (as dictated by the geometry of the situation).

The velocity of the DC motor suddenly increases at the start of this phase and then smoothly decreases after 0.2 seconds for the remaining duration of the phase (because the require torque increases)

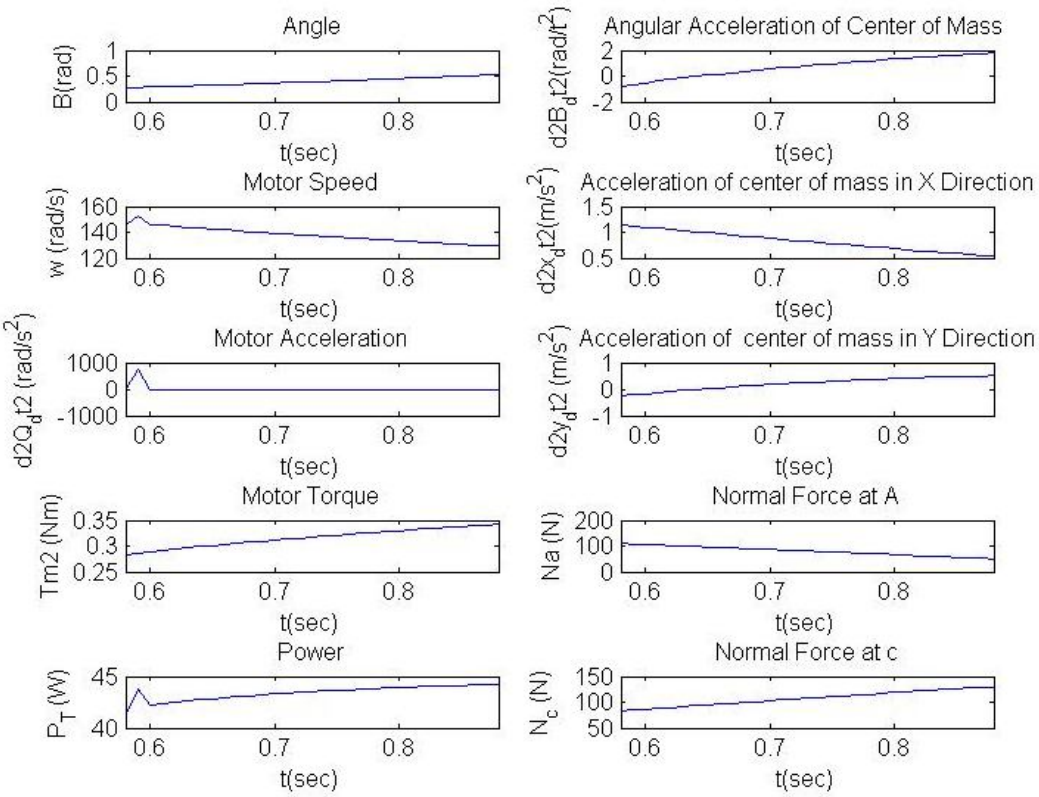


Figure 5.1.3 : Simulation of the Tracked Robot during *Phase* 2.

5.1.10 Simulation analysis of the tracked robot during *Phase* 3

The following behavior occurs:

As the front bottom part of the tracked robot touches the second nose and starts to move forward, the angle β remains constant (equal to the stairs angle). The angular acceleration of the robot will be zero. The magnitude of the torque decreases because there is no angular acceleration and the applied torque is solely used for climbing the stairs in a straight line. There will be slight changes around zero for the accelerations of the center of mass in the *X* and *Y* directions. When the front bottom part of the tracked robot touches the second step's nose (Point *C*), the center of mass will be on top of the first nose (Point *A*). Most of the robot's load thus transfers to the rear normal force, N_A (Figure 4.2.4). As the tracked robot moves forward, the center of mass gradually moves towards the front stair support (N_C). N_A gradually decreases and N_C increases.

The angular velocity of the DC motor slightly decreases for 1.75 seconds. The motor's angular acceleration slightly increases during this phase (because of the decreasing torque and increasing angular velocity of the motor), and transitions from negative to positive values. The required motor power decreases throughout the phase.

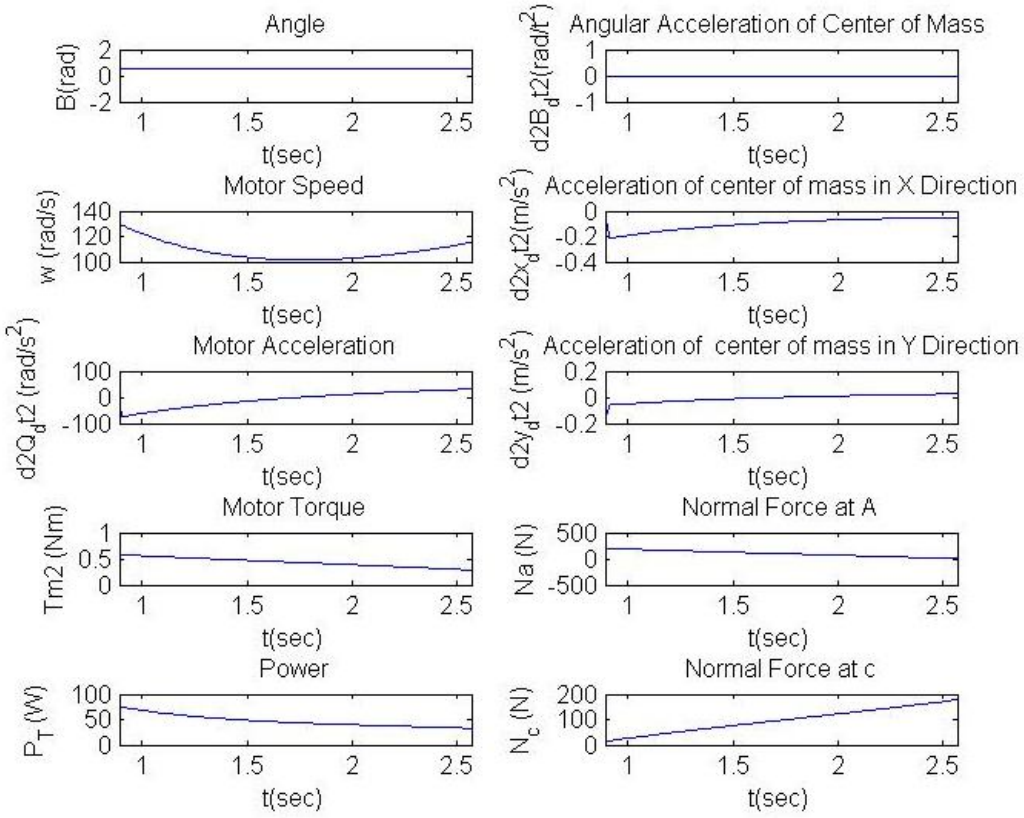


Figure 5.1.4 : Simulation of the Tracked Robot during *Phase* 3.

5.1.11 Simulation analysis of the tracked robot during *Phase* 4

The time duration of *Phase* 4 is less than two "time steps" that was chosen for simulation (one time step is 0.01 second); therefore this phase can be added to *Phases* 3 and 5.

5.1.12 Simulation analysis of the tracked robot during *Phase* 5

The simulation of the tracked robot during *Phase* 5 is similar to that of *Phase* 3.

5.1.13 Simulation analysis of the tracked robot during *Phase* 6

As the tracked robot reaches the end of the steps, it continues moving in a straight line until its center of gravity crosses the last pivot point (*Phase* 5). The tracked robot then starts to tip over (*Phase* 6, Figure 4.2.6).

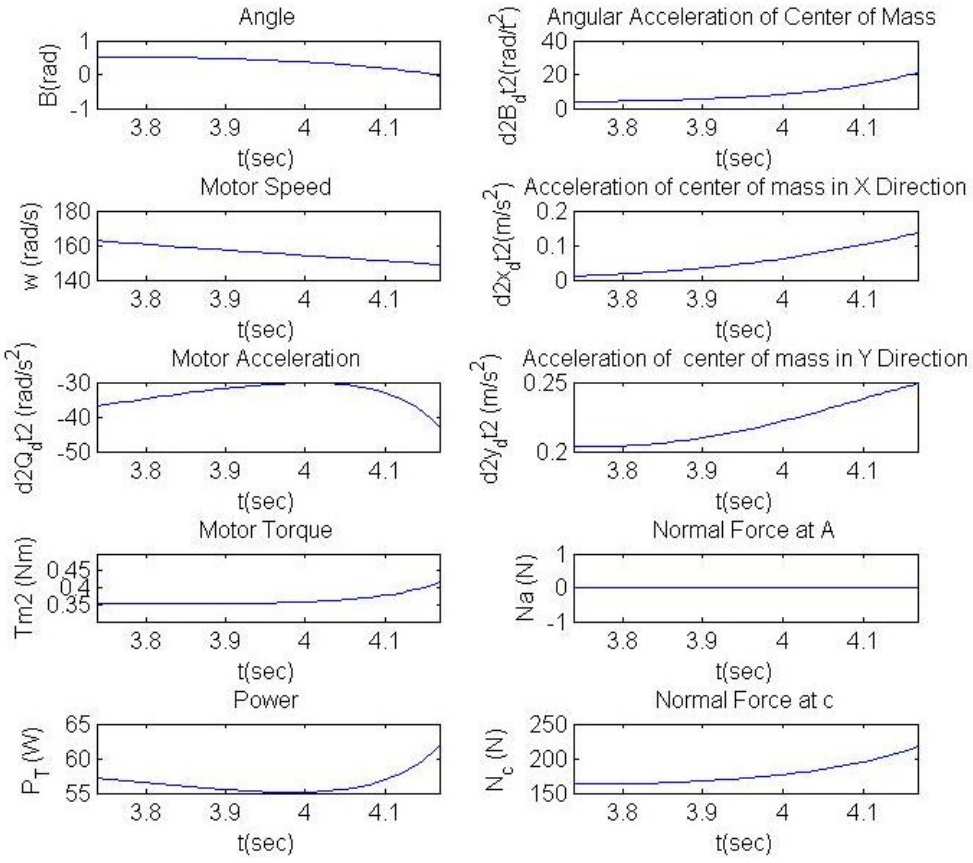


Figure 5.1.5 : Simulation of the Tracked Robot during *Phase* 6.

The following behavior occurs:

The angle β starts to decline from 30 degrees to 0 degree.

The rear reaction force of the robot (N_A) reduces suddenly to 0 because the robot starts to tip over pivoting on point *C*. The front reaction force of the robot (N_C) suddenly increases. The clockwise moment due to the robot's weight causes the robot to tip forward till it lays flat on the top staircase landing. The robot's angular acceleration thus suddenly increases until it collides with the level surface. The *X* and *Y* accelerations of the tracked robot also increase.

There is a slight increase of applied torque because the robot is still moving up.

The increasing motor torque causes the motor's angular velocity and acceleration to decrease. The power of motor slightly increases.

When the front of the robot moves above the top of the stairs, its motion is no longer completely bound to the stair geometry. The robot's configuration and velocity also affects how quickly the robot tips over and the location where the front of the robot hits the level top of the stairs.

5.1.14 Simulation analysis of the Tracked Robot during *Phases* 7and 8

After the robot climbs the stairs and reaches the flat surface at the top, the weight of the tracked robot is uniformly distributed along the grousers which have contact with level ground.

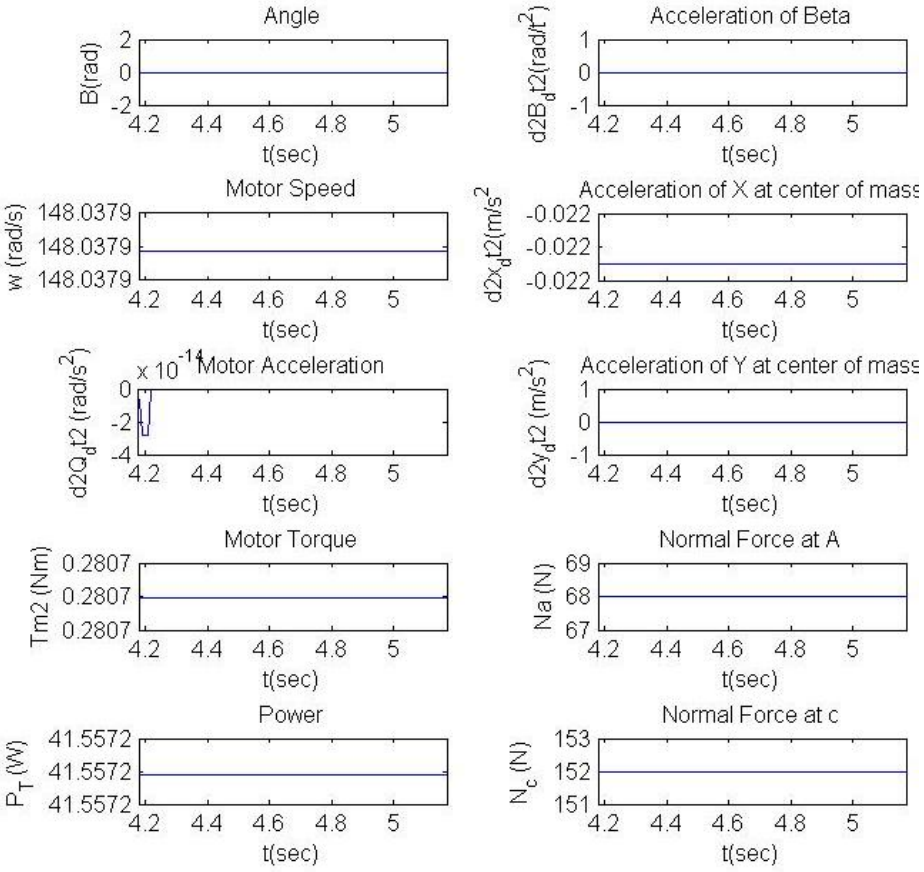


Figure 5.1.6 : Simulation of the Tracked Robot during *Phases* 7and 8.

The following behavior occurs:

There is no motion in the *Y* direction; therefore, the acceleration of the robot in the *Y* direction and its angular acceleration are both zero. The torque and speed of the motor are constant, and its angular acceleration is near zero. Consequently, the robot's acceleration in the *X* direction is slightly below zero. The normal forces remain constant. The power also remains constant.

5.1.15 Simulation of the Tracked Robot during *Phase* 9

Descending from the first step is one of the most critical situations for motion of robot that must be deliberately controlled by manipulating the robot's velocity and acceleration.

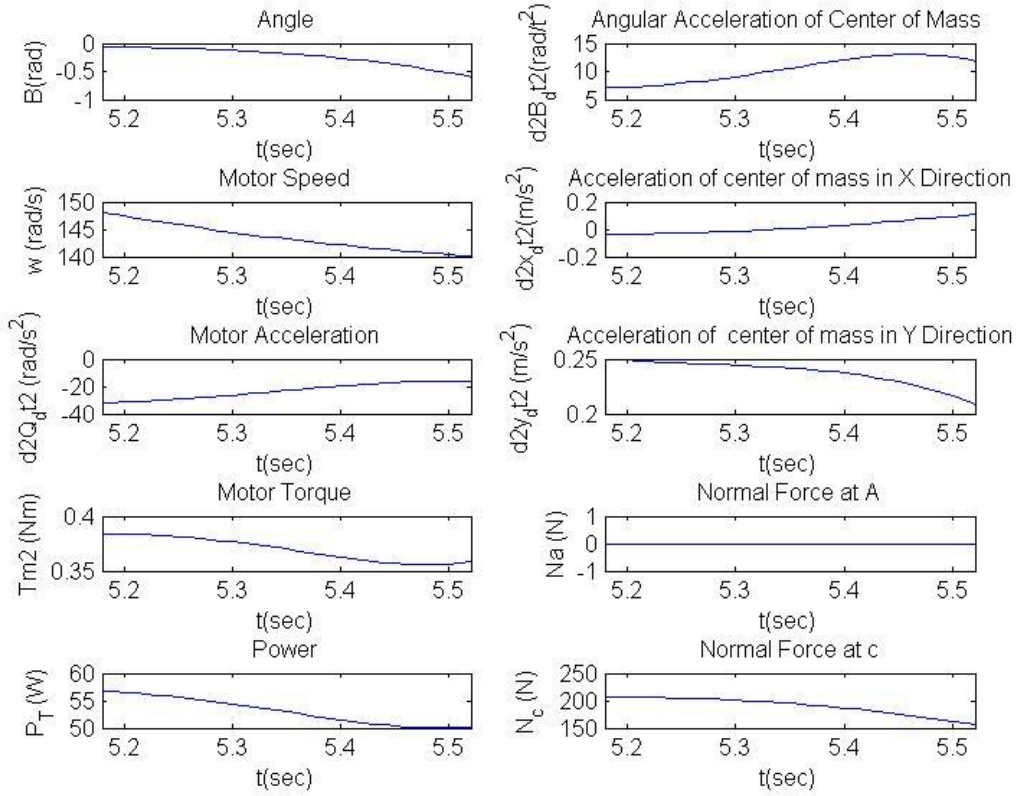


Figure 5.1.7 : Simulation of the Tracked Robot during *Phase* 9.

As the center of mass of the tracked robot crosses the rise of the first descending step, the distance between the center of mass and the rise of the step multiplied by the weight of the tracked robot produces a moment (Figure 4.2.8).

As illustrated in Figure 5.1.12, one of the highest angular accelerations occurs during the descending of the first step because of this moment. The robot tilts downwards towards the steps.

The angle β starts to decline from 0 degree to -34 degrees.

The rear reaction force of the tracked robot, N_A, goes to 0 because the robot starts to tip over around the pivot point *C*. Since all the mass of the robot is now applied at *C*, the front reaction force of the robot, N_C, suddenly increases.

Under the effect of the robot's weight and DC motor torque, the acceleration in the *X* direction also increases. There is an instantaneous increase in the acceleration in the *Y* direction, which then slightly drops as the robot's angular acceleration drops. There is a slight decrease of applied torque because the robot is forced downward by gravitational pull on the robot.

The angular acceleration of the motor increases during this phase, but always remains negative. Thus, the angular velocity continues to decrease. Furthermore, the motor power slightly decreases.

At the end of *Phase* 9, the tracked robot collides with the flat surface of the first step, behind its nose.

5.1.16 Simulation analysis of the tracked robot during *Phase*10

As shown in Figure 5.1.13, the time of this *Phase* is very short (t=0.017 sec).

The following behavior occurs:

The robot collides with the flat part of step in such a way that the bottom of the robot makes a -34 degree angle with the flat part of the step. The angle β starts to increase from -34 degrees to -30 degrees, until the front bottom part of the tracked robot touches the nose of the step (Figure 4.2.10).

The motor torque slightly decreases because the front of the tracked robot moves along a flat surface. The weight of the tracked robot pushes it downward (Figure 5.1.13).

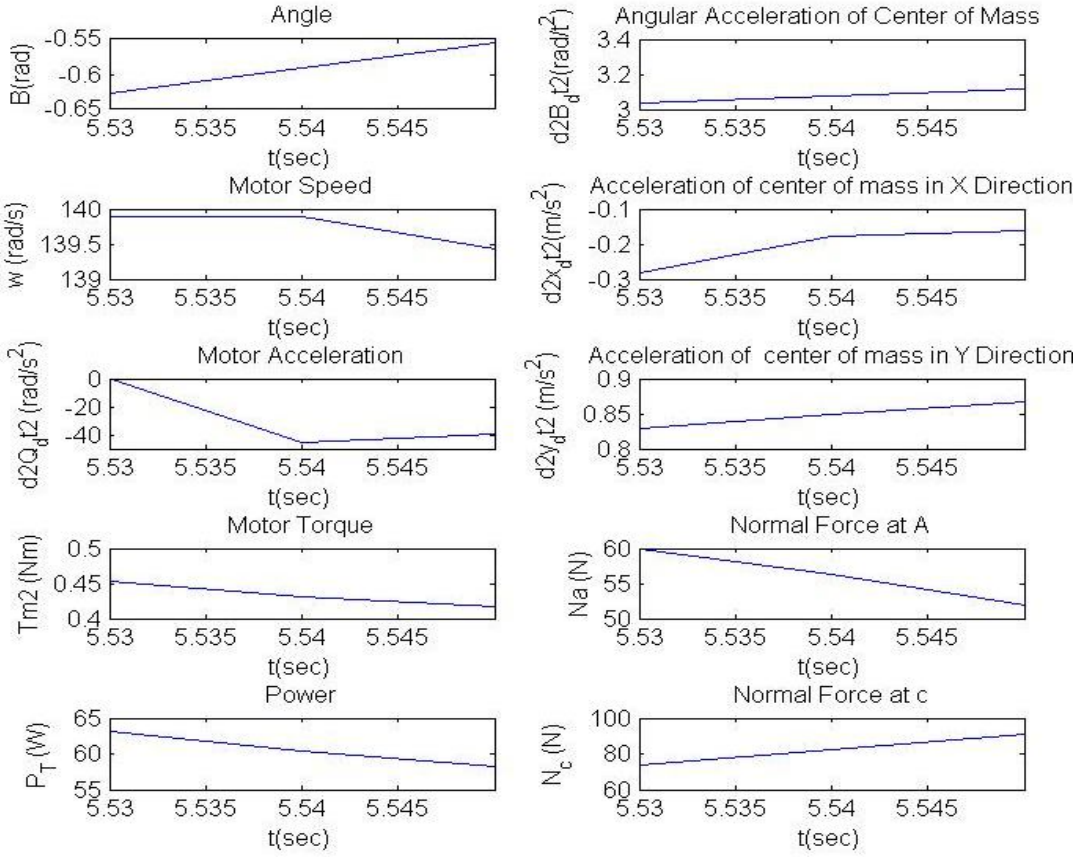


Figure 5.1.8 : Simulation of the Tracked Robot during *Phases* 10.

Under effect of the robots weight and the motors torque the accelerations of center of the mass in the X and Y directions, and the robot's angular acceleration, all slightly increase. The power of motor decreases as required torque decreases, and the change of the motor's angular velocity is negligible. The time period of this Phase is very short and the robot's motion slows down. As the tracked robot moves forward, its center of mass gradually gets closer to the front step's nose. Consequently, the front normal reaction force (N_C) increases and N_A decreases (Figure 4.2.10).

5.1.17 Simulation analysis of the tracked robot during *Phase*11

During this phase the following behavior occurs:

As the front bottom part of the tracked robot touches the second nose and starts to move forward, the angle β remains constant (equal to the stairs angle, Figure 4.2.11). The tracked robot's angular acceleration tends to zero because β is constant. The gravity component is in the direction of motion. Therefore, the motor torque remains constant and its angular acceleration

increase from negative value to zero and angular velocity decreases because of negative value of angular acceleration.

Because of constant torque, the accelerations of the tracked robot in the X and Y directions vary slightly above zero. The power consumption continuously decreases.

As the tracked robot moves forward, its center of mass gradually gets closer to the rear step's nose. Therefore the front normal force, N_C increases, and N_A decreases.

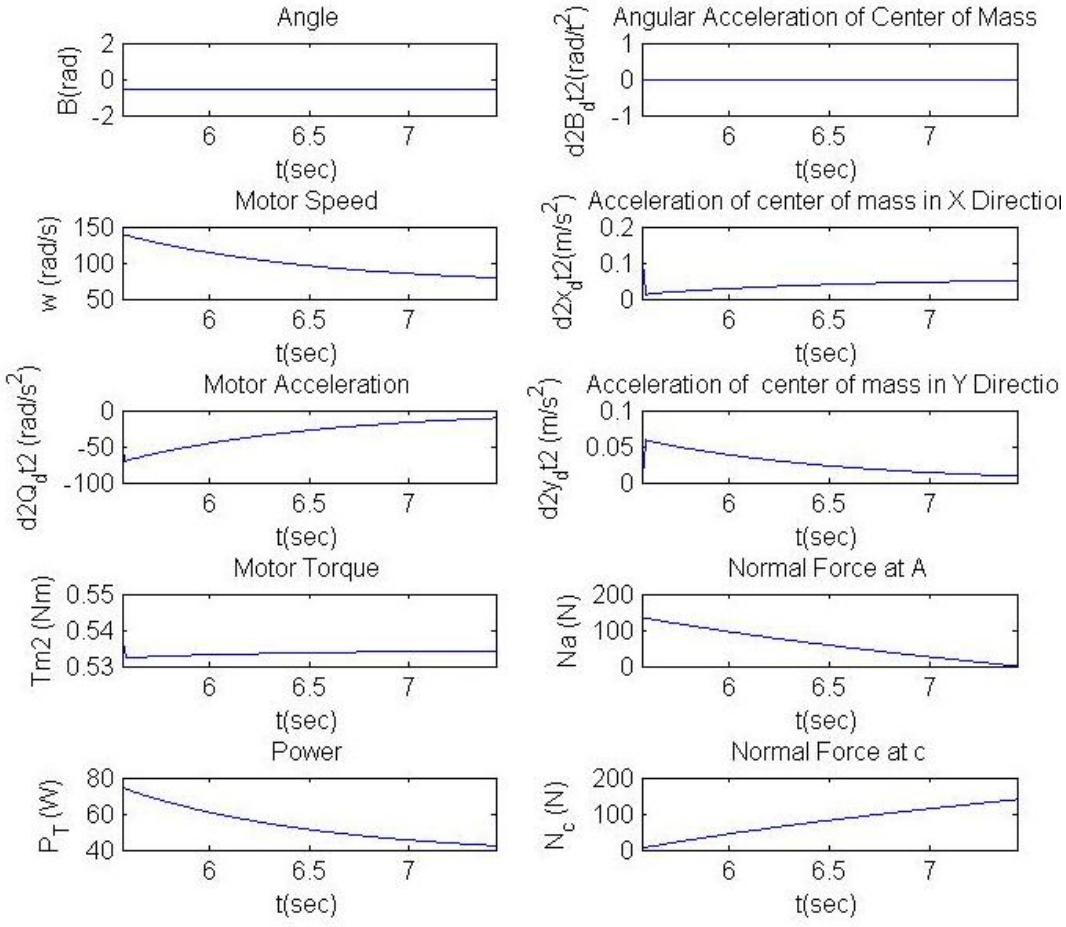


Figure 5.1.9 : Simulation of the Tracked Robot during *Phase* 11.

5.1.18 Simulation analysis of the tracked robot during *Phase* 12

The motion of the tracked robot during *Phase* 12 is similar to that during *Phase* 9, but there are two main differences. In *Phase* 9, the tracked robot starts from a flat surface (0 degree) and ends at -30 degrees, while in *Phase* 12, the angle changes are between -30.4 degrees and -34 degrees.

The total time of *Phase* 12 is much shorter than that of *Phase* 9.

Hence as it is shown in Figure 5.1.12 (*Phase* 9), except for the magnitude of the components, the behavior of these *phases* are similar.

5.1.19 Simulation analysis of the tracked robot during *Phase* 13

The motion of the tracked robot in *Phase* 13 is similar to that of *Phase* 10. The simulations are also similar.

5.1.20 Simulation analysis of the tracked robot during *Phases* 14 and 15

The time of *Phase* 14 is exactly equal to the time that was chosen for one unit time of simulation; therefore the simulation could be ignored or combined with *Phase* 15, which has similar motion. Also, *Phase* 15 is similar to *Phase* 11.

5.1.21 Simulation analysis of the tracked robot during *Phase* 16

The motion of the tracked robot and the results of *Phase* 16 are similar to those of *Phase* 12, but the magnitudes of components are slightly different.

5.1.22 Simulation analysis of tracked robot during *Phase* 17

The motion of the robot during *Phase* 17 is similar to that of *Phase* 13. As the robot reaches the level ground, there will be not any more steps to descend, and the tracked robot continues moving downward until its rear touches the nose of the last step (Figure 4.2.14).

The following behavior occurs:

The angle, β, starts at -34 degrees and ends at -14.17 degrees

The direction of gravity is in the same general downward direction as the tracked robot's motion. The motor torque slightly decreases. Thus, the angular velocity of the tracked robot slightly increases and the motor's angular acceleration varies around zero, but is never negative.

Under the effect of the increasing angular velocity of the motor, the power slightly increases.

Under the effect of the motor's angular acceleration, the robot's angular acceleration slightly increases during most of this phase and then decreases for the remainder. The robot's *X* and *Y* accelerations are almost constant, due to the robots small variations in acceleration.

As the tracked robot moves forward, its center of mass gradually gets closer to the front step's nose. Therefore the front normal force, N_C increases, and N_A decreases.

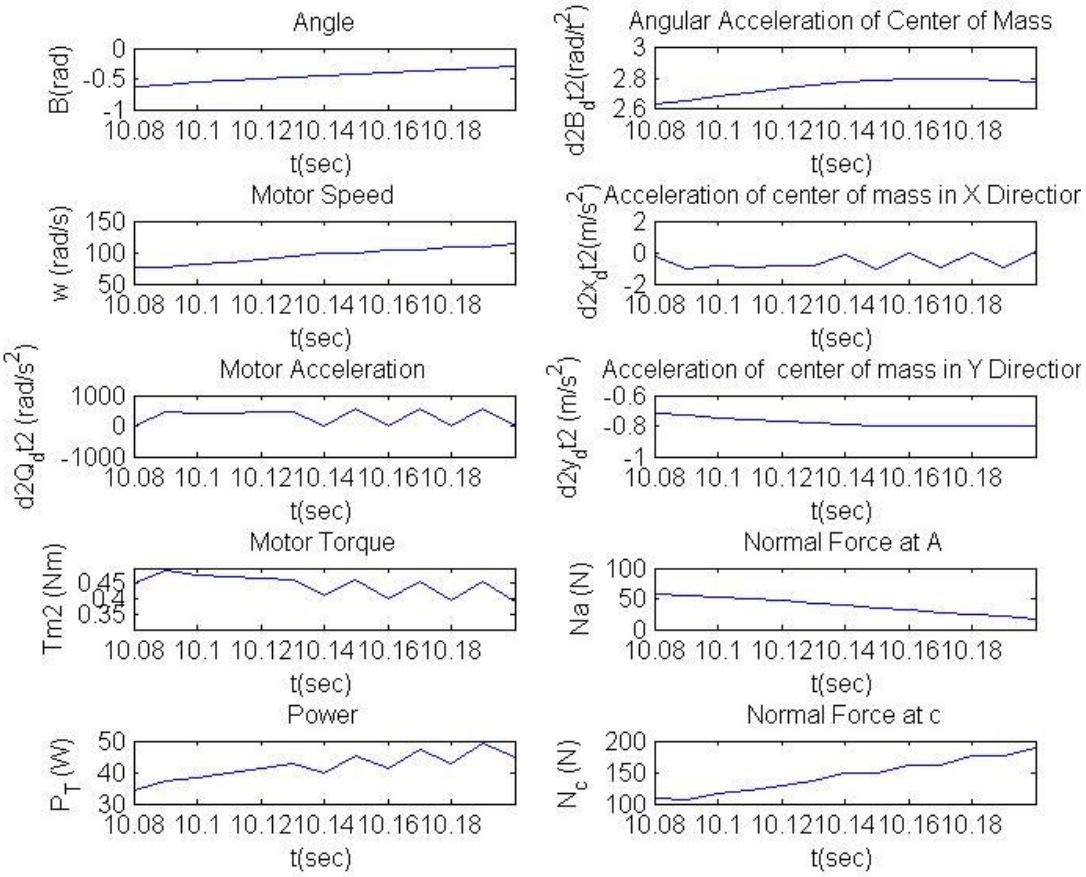


Figure 5.1.10 : Simulation of the Tracked Robot during *Phase* 17.

5.1.23 Simulation analysis of tracked robot during *Phase* 18

When the rear of the flat bottom of the robot clears the nose of the bottom step, the back of the robot falls down onto the bottom landing. At this point, the robot bangs on the bottom landing and lies flat on it.

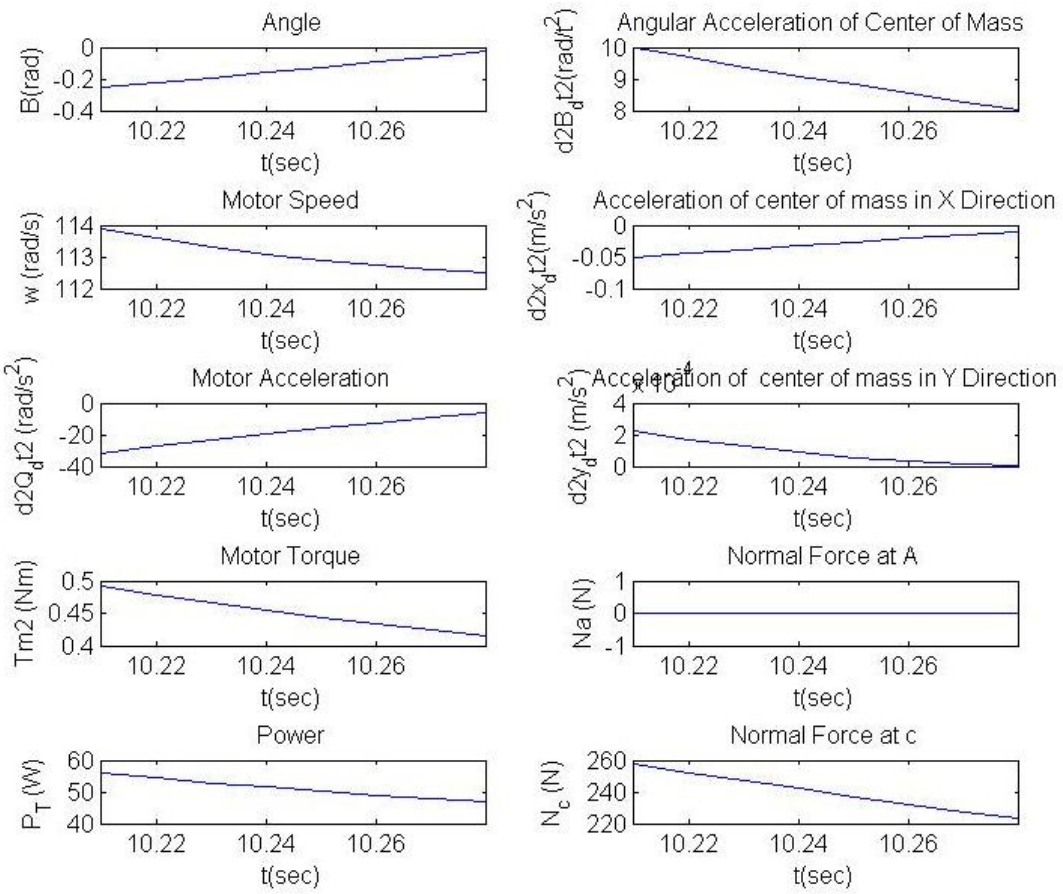


Figure 5.1.11 : Simulation of the Tracked Robot during *Phases* 18.

The following behavior occurs:

The angle β starts from -14.17 degrees and ends at 0 degree (Figure 4.2.15).

There is nothing to climb, therefore, the motor torque decreases and its angular acceleration and velocity both slightly increase. The power consumption also decreases.

The tracked robot starts with its front in contact with the ground and the rest of the robot falling down to the level ground surface. This causes a high angular acceleration of the robot, as shown in Figure 5.1.16. The load is carried by the robot's front contact point as the rear falls toward the level ground. As the robot starts to fall down, the angular acceleration of the robot suddenly increases. It then declines as the rear of the robot falls towards the ground, and suddenly drops to zero when the tracks contacts the level ground at the staircase landing. The acceleration of the robot's center of mass in the *Y* direction also first increases then starts

decreasing until the tracks contact level ground, where the acceleration drops to zero. The acceleration in the *X* direction starts with a small negative value and slowly tends towards zero.

The front normal reaction force, N_C, suddenly increases at the beginning of the phase, but then continuously decreases as the tracks contact level ground. The normal force at the rear, N_A, does not exist until the tracked robot collides with the ground.

This phase is one where the robot's accelerations depend on both the geometry and the robot's configuration and velocity.

5.1.24 Comparing the Motion of the Tracked Robot with Grousers VS Flat Belt

The behavior of the tracked robot with flat belts is generally similar to the one with grousers, but there are some differences:

When the Tracked Robot with Flat Belts (TRFB) is climbing stairs, it requires a higher torque than the robot with grousers (TRG) in the same situation.

The grousers are acting as hooks and change the direction of the reaction forces in a way that reduces the required motor torque for climbing. The TRFB requires a higher torque and thus has a lower angular velocity than the TRG. Thus, the TRFB requires a longer amount of time to travel the same distance.

The magnitudes of the normal and tangential forces at the rear of the TRFB are much higher than those of the TRG when climbing stairs. There are no grouser hooks to take a part of the load, and most of the robot's load is transferred to the back.

Because it was assumed that there is no slippage, the motion of the tracked robot with grousers and without grousers are similar during descending.

There are several critical situations for both TRFB and TRG where the robot could tip over or be damaged. These situations occur when the tracked robot starts to climb the last step (*Phase* 6) and when it begins to descend (*Phases* 9, 12, 16, and 18).

Figures 5.1.17 and 5.1.18 show the simulations of the tracked robot with grousers and with flat belts for all 18 Phases. Details of these simulations are provided in Appendix D.

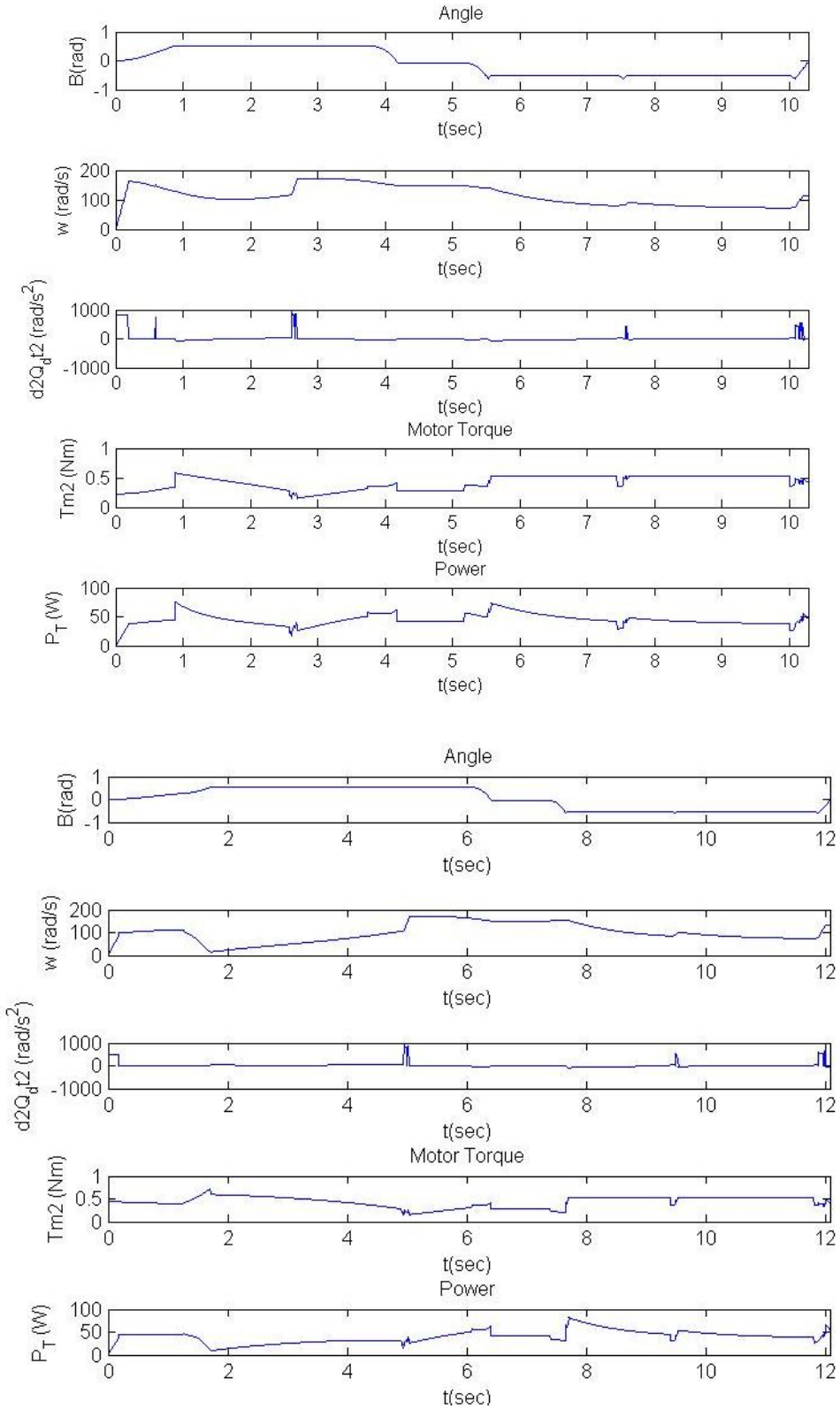


Figure 5.1.12 : Top _Simulation of the 18 *Phases* of the Tracked Robot with grousers (Part1)

Bottom_ Simulation of the 18 *Phases* of the Tracked Robot with flat belt (Part1)

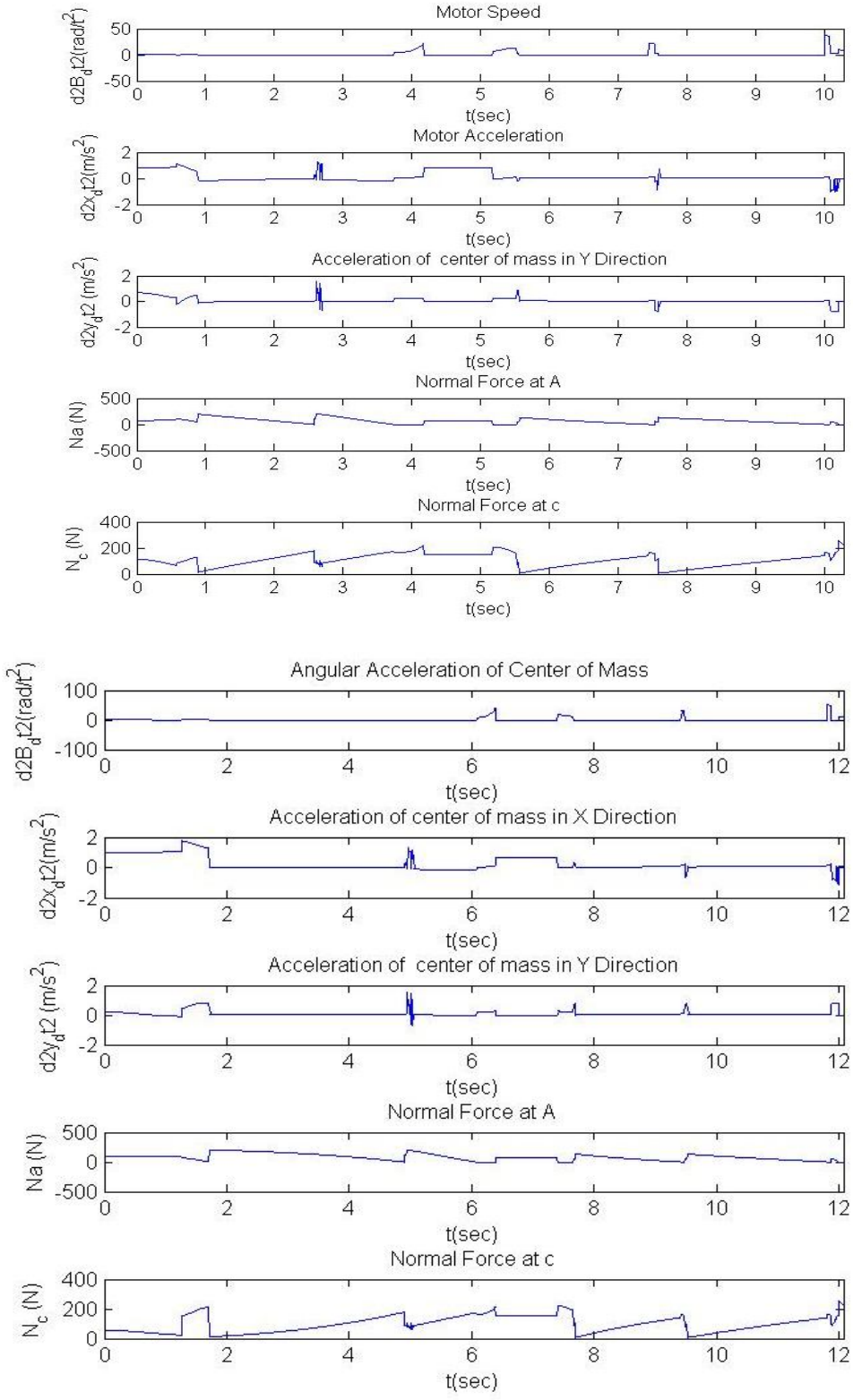


Figure 5.1.13 : Top _Simulation of the 18 *Phases* of the Tracked Robot with grousers (Part2)

Bottom_ Simulation of the 18 *Phases* of the flat belt Tracked Robot with (Part2)

Chapter 6

6.1 Conclusion

6.1.1 Conclusion and Recommendation for Future Work

The main purpose of this research was to define the best geometry and mechanical design of a stair climbing robot to optimize its efficiency and effectiveness. The thesis starts with a review of the North American stair standards and the existing robot designs that are described in the available technical and scientific literature. These findings are summarized in Chapter 2 of this thesis.

Researching the existing literature narrowed the suitable robot configurations to three possible candidates: the Star-wheel, the variable configuration tracked robot, and the non variable configuration tracked robot. These candidate robots were then examined in greater detail to determine their suitability. The Starr-wheel configuration was eliminated due to its tighter requirements on the design for a specific stair type. This robot has a higher possibility of slippage than tracked robots do when climbing or descending stairs with a geometry other than the optimum one that the robot was designed for.

The variable configuration tracked robot has many advantages in stair climbing duty. Their front track configurability and ability to change their center of mass gives them great flexibility when encountering different stair geometries. This is also useful for crossing other

obstacles and is a major help when descending stairs. These advantages ensure that many currently existing stair climbing robots are variable configuration ones.

Variable configuration robots do come with many disadvantages that can offset their advantages. The variable tracks require that the robot carries at least two additional motors, as well as the mechanical linkages to operate and drive the variable tracks. These additional parts increase the power consumption and create a heavier and more expensive robot.

The third, and final, candidate is the non variable configuration tracked robot. This type of robot has many of the same advantages as the variable configuration type, but trades some flexibility for a potentially lighter and smaller robot. The main thrust of this thesis was to determine the best design for a non variable tracked robot for climbing and descending stairs. In return for spending more time and effort up front to design an efficient machine for its specific purpose, there is a great potential for creating a simpler, more robust, smaller, and more efficient robot.

A literature search was carried out to determine if any literature has been published on the detailed dynamic analysis of a tracked robot climbing stairs with simple non-variable configuration robot throughout the staircase climb and descend. To the best of available knowledge so far such an analysis was not done. This set the main task of this thesis to be the design of a non variable configuration tracked robot and analyze the behaviors of the robot during the different stair ascending and descending phases. This research is not only useful for non variable tracked robots, but can also be a base for analyzing any other types of robots that have stair climbing tasks.

After selecting the tracked robot as the most reliable type for climbing stairs, the factors that must be considered for the robot design must be determined. The first of these constraints is to know the geometries of the stairs that are to be climbed. In the general case, a robot will be required to climb and descend several sets of stairs, each of which may have a different geometry. Thus, the stair dimensions might vary within a set range of values provided the staircases adhere to the requirements of the building code. For the purposes of this design, the North American stair standards were selected.

The stair specifications were then used to constrain the robot design. The minimum length of the robot was determined, based on the distance between two adjacent stair noses. This

analysis revealed that the bottom length of the tracked robot must contact a minimum of two stairs at all times during the stair climb.

The time when the robot first contacts the stairs and starts climbing them is very critical and is one of the most difficult situations in the stair climbing situation. This makes it important to devote a significant amount of time to designing the robot in order to minimize the power consumption and maximize the stability during this phase. The shape of the front of the robot has a significant effect on the ease of the initial climbing operation. For a robot with a round front shape, as defined by the track's front wheel, static analysis revealed that the minimum wheel radius must be more than twice the step's rise, for practical coefficient of friction values. An alternate configuration to the round front shape is a robot with a front track section that forms an angle of attack to the stairs. The analysis of a robot with smooth belts and a front angle of attack showed that this type of robot can be smaller and use less energy than an equivalent robot with a round front.

The robot with a front angle of attack and smooth belts was then compared to a similar robot where the smooth belts were replaced by tracks containing grousers. The grousers allow the robot to get a better grip on the stair noses during climbing. This analysis showed that a tracked robot with grousers can have a larger angle of attack and the tracks can have a lower coefficient of friction than the smooth belt robot. The robot with grousered tracks consumes less energy when climbing, due to the better stair grip. The larger maximum allowed angle of attack allows the robot to be smaller and thus weigh less than the smooth belt robot. These comparisons lead to the selection of a robot with a front angle of attack and tracks with grousers as the best configuration for the stair climbing functionality.

The next task of this study was to determine the location of the robot's center of mass. Static analysis showed that the center of mass must be above the nose of the first step when the tip of the flat bottom contacts the second step. The dynamic analysis imposed additional constraints; namely, that the location of the center of mass cannot be too far to the front otherwise the robot may tip over during descent. Additionally, the center of mass cannot be too far to the back otherwise it could create a strong impact when the rear of the robot falls to the ground after leaving the final step during descent. This set of constraints limits the possible location of the center of mass.

The shape and dimensions of the track were also studied to determine the best grouser pitch. The range of allowable pitch values is constrained by possible slippage or falling when the robot is climbing or descending stairs. The pitch must also be selected to minimize the possible yaw angle on the stairs.

A detailed dynamic analysis was then performed for the robot as it climbs and then descends a set of stairs. This analysis verified the robot design and helps in an understanding of the power consumption of the robot. It also showed where the critical conditions are that would cause the robot to tip over. For this analysis, it was assumed that there is no track slippage, yaw angle, or belt deflection.

As the robot ascends and descends the staircase, it transitions through several distinct phases, each of which must be separately analyzed. There are a total of eighteen phases to be studied. These phases were analyzed in Chapter 4 and simulated in Chapter 5.

This analysis was performed for the cases of the robot with grousers and one with smooth belts. This comparison showed that the robot with grousers consumes less energy and takes less time to climb the stairs than the equivalent smooth belt robot does. Both robots consume the same power and take the same time during the descent stage, under the assumptions of no track slippage in both cases.

The results of the studies reported in this thesis give a good foundation for the design of a stair climbing robot. The robot design is much simpler than most existing stair climbing robots, due to the elimination of the variable configuration arms. This robot is smaller and weighs less than existing designs, and can thus run for a longer period of time with the same set of batteries. The static and dynamic analysis techniques presented here put the design on a sound theoretical basis. The theory was confirmed by simulation studies and the construction of a prototype robot.

Other issues not tackled in the current study include; the effect of the impact force on the tracked robot when it bangs on the next step during descending, the effect of track slip, track deflection when contacting the nose of a step, the effect of the YAW angle when the robot is climbing and descending stairs, the effect of track/ground friction during skid steering.

6.1.2 Recommendations for Future work

The time that a tracked robot spends climbing and descending stairs in malls and buildings is much less than the time that it travels on smooth flat surfaces or flat surfaces covered with rugs or mats. Chapter 2 showed that the robot's energy consumption while turning is greater than that when it climbs stairs. The effect of these situations is that, in nearly all cases, the robot consumes much more energy on flat surfaces than when climbing stairs. (This is true for both variable and non variable configuration tracked robots.)

Wheeled robots have been shown in Chapter 2 to be the best choice for flat surfaces; being more maneuverable and consuming less energy. If the robot did not need to climb stairs, a wheeled robot would suffice, and would require fewer batteries for the same run time and would be smaller and lighter than a tracked robot.

A combination of the tracked robot and the wheeled robot could be a better choice for building surveillance, if such a robot could be designed. One such concept is a design inspired from Rail Cars. Rail Cars are ordinary vehicles that have both a set of road wheels and a set of railway wheels. A set of levers and clutches selects which set of wheels are engaged. In one position, the road wheels are pushed down to connect them to the ground and lift the rail wheels off of the surface. The road wheels are then driven by the motors. When the vehicle is driven onto a railroad track, the levers are set to lower and disengage the road wheels, connecting the railway wheels to the train tracks. These vehicles can travel on both roads and train tracks.

This design could be modified for tracked robots so that when the robot is on level ground, wheels are lowered to contact the level ground and lift the tracks off of the ground. The robot would then act as a wheeled robot (with the additional payload weight of the set of tracks and its supporting mechanical structures). This would result in a huge decrease in energy consumption on a flat surface compared to a pure tracked robot, even with the additional weight of the track mechanisms. This would allow either the run time to be increased or the battery weight to be decreased, or a combination of the two.

The tracked/wheel robot combination could also be implemented by using a different configuration. The front wheels could be attached to an arm mechanism that can rotate 135 degrees about the Z axis (perpendicular to the X axis of forward motion and the vertical Y axis). In the 0 degree configuration, the front wheels contact the ground and the robot's tracks are off of

the ground. The robot acts as a wheeled robot in this configuration. When a set of stairs is being approached, the robot would rotate the arm mechanism by 135 degrees, which raises the wheels off of the ground and gives them a 45 degree angle with the level ground. The tracks then make contact with the ground. The robot then acts as a tracked robot with the rotated front wheel part acting as an angle of attack for the robot when it begins to climb the stairs.

Both of these wheeled/tracked robot proposals are possible configurations that merit additional feasibility studies. They offer potential power savings over a tracked robot. The additional size and weight of the additional wheels and related switching mechanisms might be offset by the reduced battery requirements. This could potentially result in a smaller and lighter robot than the dedicated tracked robot.

Finally, studies could be performed to replace much of the robot's steel structures with composite materials. This could reduce the robot's weight and potentially strengthen its structural integrity, if the composites are deployed correctly. This could be a fruitful approach to consider.

References

[1] History of Robotics, Robotics – Intelligent Connection of the Perception to Action Codrin Pasca, University of Ottawa May 5, 2003

[2] The industrial revolution, Lee T Wyatt III, 2009

[3] Robotics and Automation Handbook, Edited By Thomas R. Kurfess, Ph.D. PE, 2005, CRC Press.

[4] Robôs Móveis Autônomos. [Online]. http://davidbuckley.net/DB/HistoryMakers/HM-Elektro.htm

[5] Organisms, Simulated and Real, Larry Yeager; Professor of Informatics, Indiana University. [Online]. http://www.ias.uwe.ac.uk/Robots/gwonline/gwonline.html 2 June 2010.

[6] Manish Parashar and Omer Rana, Autonomic (Grid) Company. [Online]. http://www.extremenxt.com/walter.htm5 Feb 2011.

[7] Samsung Techwin SGR- A1 Sentry Guard Robot.[Online]. http://www.globalsecurity.org/military/world/rok/sgr-a1.htm

[8] iRobot Corporation government and industrial robots.[Online] http://www.irobot.com/gi/ground 2010.

[9] Dirauf, F., Gohlke, B, Fischer, E., “Innovative Robotics and Ultrasonic Technology at the Examination of Reactor Pressure Vessels in BWR and PWR Nuclear Power Stations.”, http://www.ndt.net/article/ecndt98/nuclear/222/222.htm, 1998.

[10] Sutter, John D. “Deep-sea mysteries: Why drilling in 'inner space' tests human limits”, http://edition.cnn.com/2010/TECH/innovation/07/06/oil.ocean.mystery, 2010.

[11] Field Experiment on Multiple Mobile Robots conducted in an Underground Mall, Tomoaki Yoshida, Keiji Nagatani, Eiji Koyanagi, Yasushi Hada, Kazunori Ohno, Shoichi Maeyama, Hidehisa Akiyama, Kazuya Yoshida, and Satoshi Tadokoro, J uly16, 2009.

[12] “Underwater Robot Sent to Study Deepwater Horizon Spill”, [Online].http://www.sciencedaily.com/releases/2010/05/100528211158.htm, 2010.

[13] Marc H. Raibert, H. Benjamin Brown, Jr., Michael Chepponis, Eugene Hastings, Jeff Koechling, Karl N. Murphy, Seshashayee S. Murthy, Anthony J. Stentz, Progress Report, **“**Dynamically Stable Legged Locomotion”: Leg Laboratory, the Robotics Institute and Department of Computer Science Carnegie-Mellon University Pittsburgh, PA. 1521313 October 1982 - October 1983.

[14] Filipe M. Silva J.A. Tenreiro Machado “Energy Analysis During Biped Walking” in *IEEE International Conference on Robotics & Automation Detroit, Michigan* l May 1999.

[15] Martin Buehler, “Dynamic Locomotion with One, Four and Six-Legged Robots” in Journal of the Robotics Society of Japan , 20(3):15-20, April 2002.

[16] E. Z. Moore and M. Buehler, “Stable Stair Climbing in a Simple Hexapod Robot” in *IEEE International Conference on Robotics & Automation*(ICRA), Taipei, Taiwan, Sept 2003.

[17] A. Seyfarth, F. Iida, R. Tausch, M. Stelzer, O. von Stryk and A. Karguth “Towards Bipedal Jogging as a Natural Result of Optimizing Walking Speed for Passively Compliant.

Three-Segmented Legs*”* In *The International Journal of Robotics Research* vol. 28 no.2 257-265 February 2009.

[18] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter and the BigDog Team “BigDog, the Rough-Terrain Quaduped Robot”, in the 17th World Congress. The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.

[19] Murthy, S. S., Raibert, M. H. “3D balance in legged locomotion: modeling and simulation for the one-legged case. In Inter-Disciplinary Workshop on Motion”: Representation and Perception, ACM, 1983-194.[Online]. www.ai.mit.edu/projects/leglab/robots/3D.../3D_hopper.html.

[20] S. Cordes, K. Berns, “A Flexible Hardware Architecture for the Adaptive Control of Mobile Robots”, in *3rd Symposium on Intelligent Robotic Systems '95,* Pisa, Italy, 1995.

[21] Imran Waheed, “Trajectory / Temporal Planning of a Wheeled Mobile Robot” A Thesis Submitted to the College of Graduate Studies and Research in Partial Fulfillment of the Requirements for the Degree of Master of Science in the Department of Mechanical Engineering Page 23 University of Saskatchewan Canada, December 2006.

[22] Chun-Kyu Woo, Hyun Do Choi, Mun Sang Kim*, Soo Hyun Kim and Yoon Keun Kwak **“**Optimal Design of a New Wheeled Mobile Robot by Kinetic Analysis for the Stair-Climbing States” *Korea Advanced Institute of Science and Technology* Korea Institute of Science and Technology South Korea*, Publisher: InTech, October 2007.

[23] Shigeo Hirose, “Super Mechano-System: New Perspective for versatile Robotic System”.Tokyo Institute of Mechanical and Aerospace Engineering 2-12-1 Ookayama Meguro-Ku, Tokyo 152-8552, Japan, 2001.

[24] Mohsen M.Dalvand and Majid M. Moghadam “Stair Climbing Smart Mobile Robot (MSRox)” Tarbiat Modares University of Iran, in: · Journal Autonomous Robots archive, Volume 20 Issue 1, January 2006.

[25] Amendment to the Human Factors Design “Standards of stairs and Doors” Chapter 10: Exhibit 10.4.8.3.1 (B) Tread depth and (G) Handrail height June 2006.

[26] Majid M. Moghadam and Mojtaba Ahmadi "Climbing Robots" Publisher: InTech, September 2007.

[27] ROBHAZ-DT3: Woosub Lee, Sungchul Kang, Munsang Kim "Teleoperated Mobile Platform with Passively Adaptive Double-Track for Hazardous E environment Applications". In IEEE ASJ International Conference on Intelligent Robots and Systems, vol.1, pp 33 - 38, Sendal, Japan, September 28 -October 2, 2004.

[28] Christopher Luke Cawley "The Enhancement of a Multi-Terrain Mechatron for Autonomous Outdoor Applications" A thesis submitted in partial fulfillment of the requirements for the Degree of Master of Science Physics and Electronic Engineering in at the University of Waikato, 2006.

[29] Taro Iwamoto Hiroshi Yamamoto "Mechanical Design of Variable Configuration Tracked Vehicle" [Journal of Mechanical Design 112, 289 (1990)].

[30] S. Ali Moosavian, Hesam Semsarilar, Arash Kalantari "Design and manufacturing of mobile rescue robot", in International Conference on Intelligent RObots and Systems - IROS - IROS , pp. 3982-3987, 2006.

[31] Jinguo Liu, Yuechao Wang, Shugen Ma,and Bin LiJinguo Liu, Yuechao Wang, Shugen Ma,and Bin Li "Analysis of Stairs-Climbing Ability for a Tracked Reconfigurable Modular Robot". In *IEEE*, Safety, Security and Rescue Robotics workshop pp 36 – 41, Japan, 6-9 June 2005.

[32] Park Nam-Eun, Park Dong-II and Kwak Yoon-Keeum. "Track shape for stair adaptability improvement of the rescue robot". KSME Spring season conference (2005)

[33] Haijun Mo Ping Hung Shaowei Wu. "Study on Dynamic Stability of a Tracked Robot Climbing over an Obstacle or Descending Stairs" in Third *IEEE* International workshop, 4 pp. – 213,17-19Jan.2006.

[34] Anastasios I. Mourikis1, Nikolas Trawny1, Stergios I. Roumeliotis1, Daniel M. Helmick2, and Larry Matthies2, "Autonomous Stair Climbing for Tracked Vehicles". International Journal of

Robotics Research & International Journal of Computer Vision, vol. 26, no. 7, pp. 737-758, July 2007.

[35] Design Advisor: Prof. Mavroidis Jon Hastie, Ben Jacobs Design Team Chris Martell, Tim Parker Amir Talakoub “Development of a New Arm For The Foster-Miller TALON Robot” MIMEU701-702 Technical Design Report. The Capstone Design Course Report Format Project #F07 Mid-Term Report Sponsor: Foster-Miller Inc． November 05, 2007.

[36] Daniel M. Helmickl, Stergios I. Roumeliotis2, Michael C. McHenry3, Larry Matthies4 JPL, Pasadena, “Multi-Sensor, High Speed Autonomous Stair Climbing”, in IEEE *RSJ,* Conference on inelegant Robots and Systems, vol.1, pp 733 – 742, Dec10, 2002. lhm@telerobotics.jpl.nasa.gov

[37] Jean-Luc Paillat “Variable Geometry Tracked Vehicle, description, model and behavior” Laboratoire d’Ingenierie des Systèmes Automatisés Angers, France Philippe Lucidarme

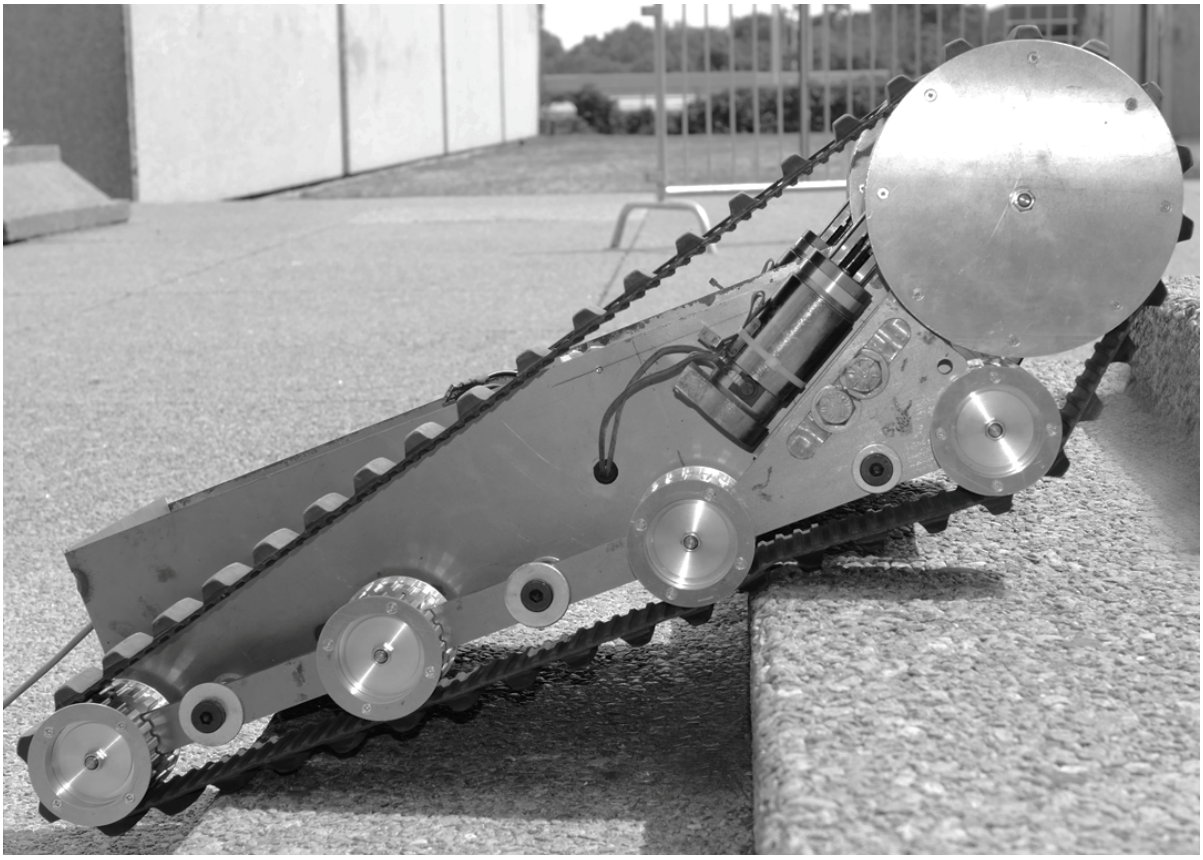
Laboratoire d’Ingenierie. Email: jlpaillat@gmail.com des Systèmes Automatisés. Mecatronics, 21-23 May 2008. laurent.hardouin@istia.univ-angers.fr

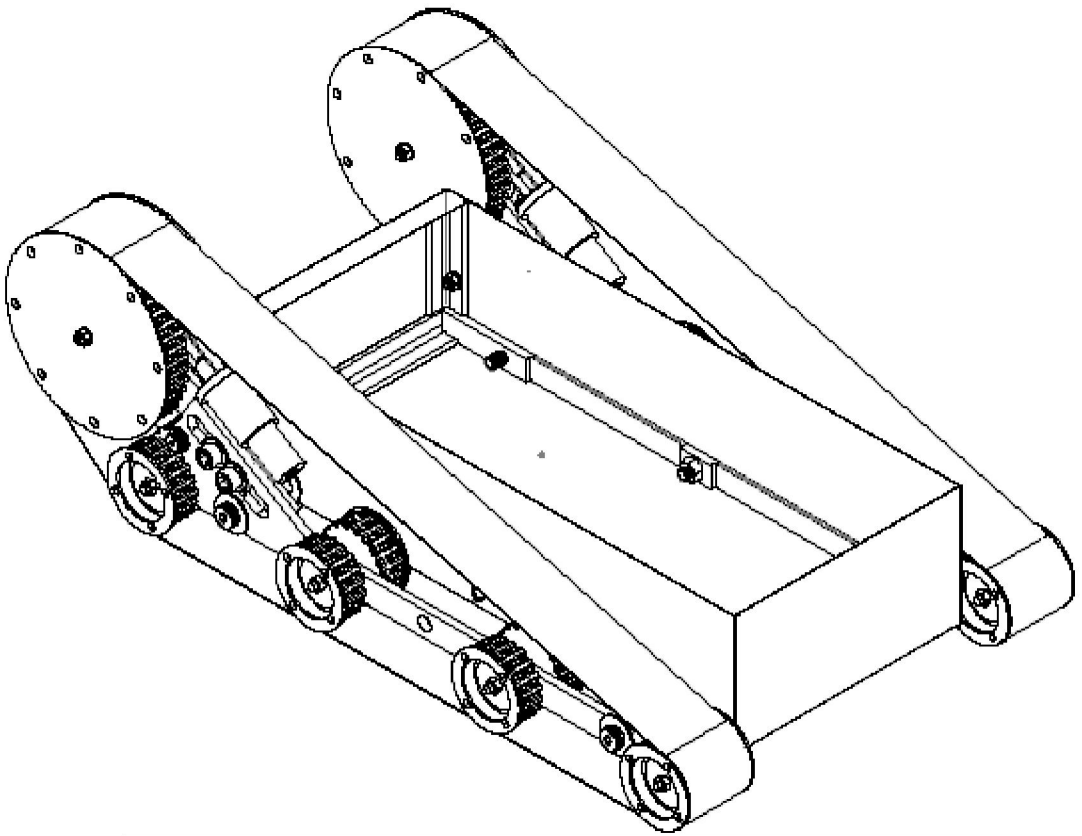
[38] A Comprehensible Guide to Servo Motor Sizing, By Wilfried Voss, Published by Copperhill Technologies Corporation, 158 Log Plain Road, Greenfield, MA 01301 Copyright © 2007 by Copperhill Technologies Corporation, Greenfield, Massachusetts

[39] maxon DC motor and maxon EC motor, Key information ,May 2009 edition

Appendix A

A.1 Engineering Drawing of the Prototype Tracked Robot

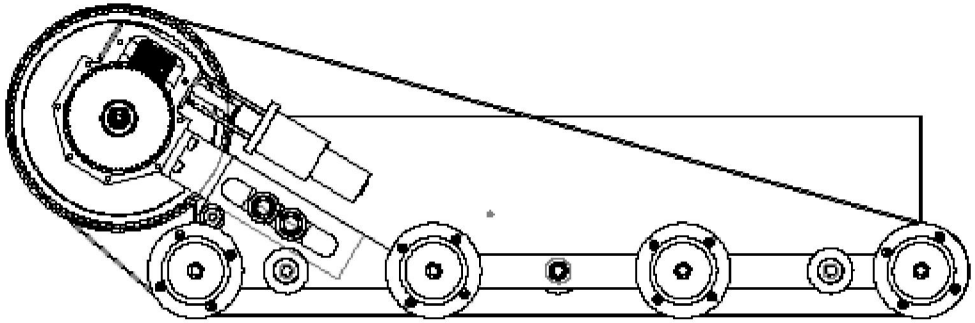




		UNLESS OTHERWISE SPECIFIED:		NAME	DATE
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: ±1° TWO PLACE DECIMAL ±0.01 THREE PLACE DECIMAL ±0.001	DRAWN		
			CHECKED		
			ENG APPR.		
			MFG APPR.		
			Q.A.		
		INTERPRET GEOMETRIC TOLERANCING PER:	COMMENTS:		
		MATERIAL	Made By Homayuon Rastan Edited b Dr. Fahim This part is dangerous		
NEXT ASSY	USED ON	FINISH			
APPLICATION		DO NOT SCALE DRAWING			

TITLE: Tracked Robot Assembely		
SIZE	DWG. NO.	REV
SCALE: 1:10	WEIGHT:	SHEET 1 OF 1

5 4 3 2 1



		UNLESS OTHERWISE SPECIFIED:		NAME	DATE
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN		
			CHECKED		
			ENG APPR.		
			MFG APPR.		
		INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
		MATERIAL	COMMENTS:		
NEXT ASSY	USED ON	FINISH	Made By Homayuon Rastan Edited b Dr. Fahim This part is dangerous		
APPLICATION		DO NOT SCALE DRAWING			

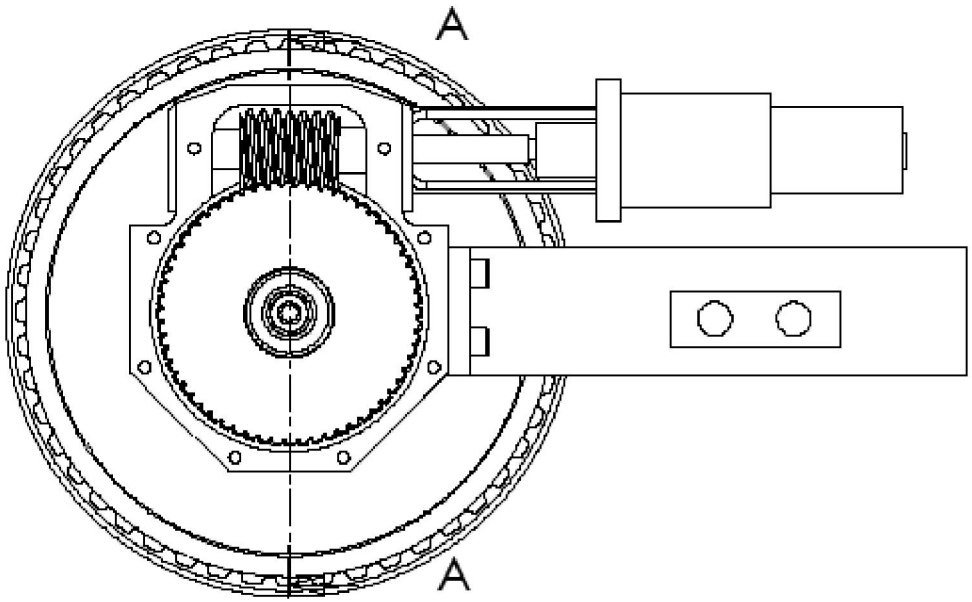
TITLE:

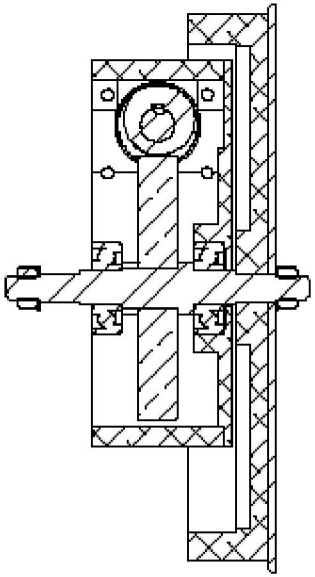
Tracked Robot assembly

SIZE	DWG. NO.	REV
SCALE: 1:10	WEIGHT:	SHEET 1 OF 1

5 4 3 2 1

SECTION A-A
SCALE 1 : 2

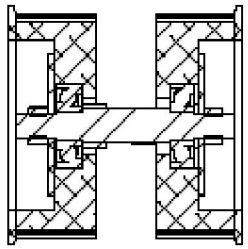


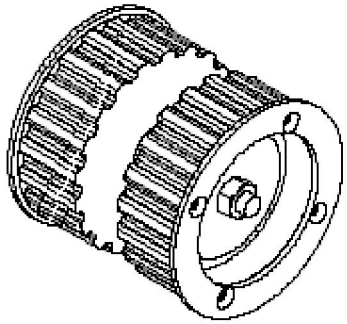


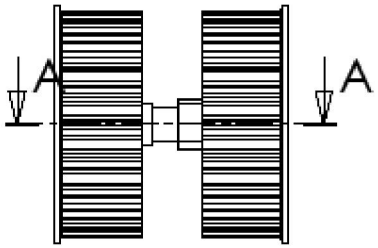
		UNLESS OTHERWISE SPECIFIED:		NAME	DATE
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN		
			CHECKED		
			ENG APPR.		
			MFG APPR.		
		INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
		MATERIAL	COMMENTS:		
NEXT ASSY	USED ON	FINISH	Made By Homayuon Rastan Edited b Dr. Fahim This part is dangerous		
APPLICATION		DO NOT SCALE DRAWING			

TITLE: Gearbox &Pulley Assembely		
SIZE	DWG. NO.	REV
SCALE: 1:5	WEIGHT:	SHEET 1 OF 1

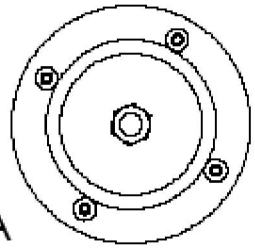
5 4 3 2 1







SECTION A-A



		UNLESS OTHERWISE SPECIFIED:		NAME	DATE
		DIMENSIONS ARE IN millimeters TOLERANCES: FRACTIONAL ± ANGULAR: ±1° TWO PLACE DECIMAL ±0.01 THREE PLACE DECIMAL ±0.001	DRAWN		
			CHECKED		
			ENG APPR.		
			MFG APPR.		
			Q.A.		
		INTERPRET GEOMETRIC TOLERANCING PER:	COMMENTS:		
		MATERIAL	Made By Homayuon Rastan Edited b Dr. Fahim This part is dangerous		
NEXT ASSY	USED ON	FINISH			
APPLICATION		DO NOT SCALE DRAWING			

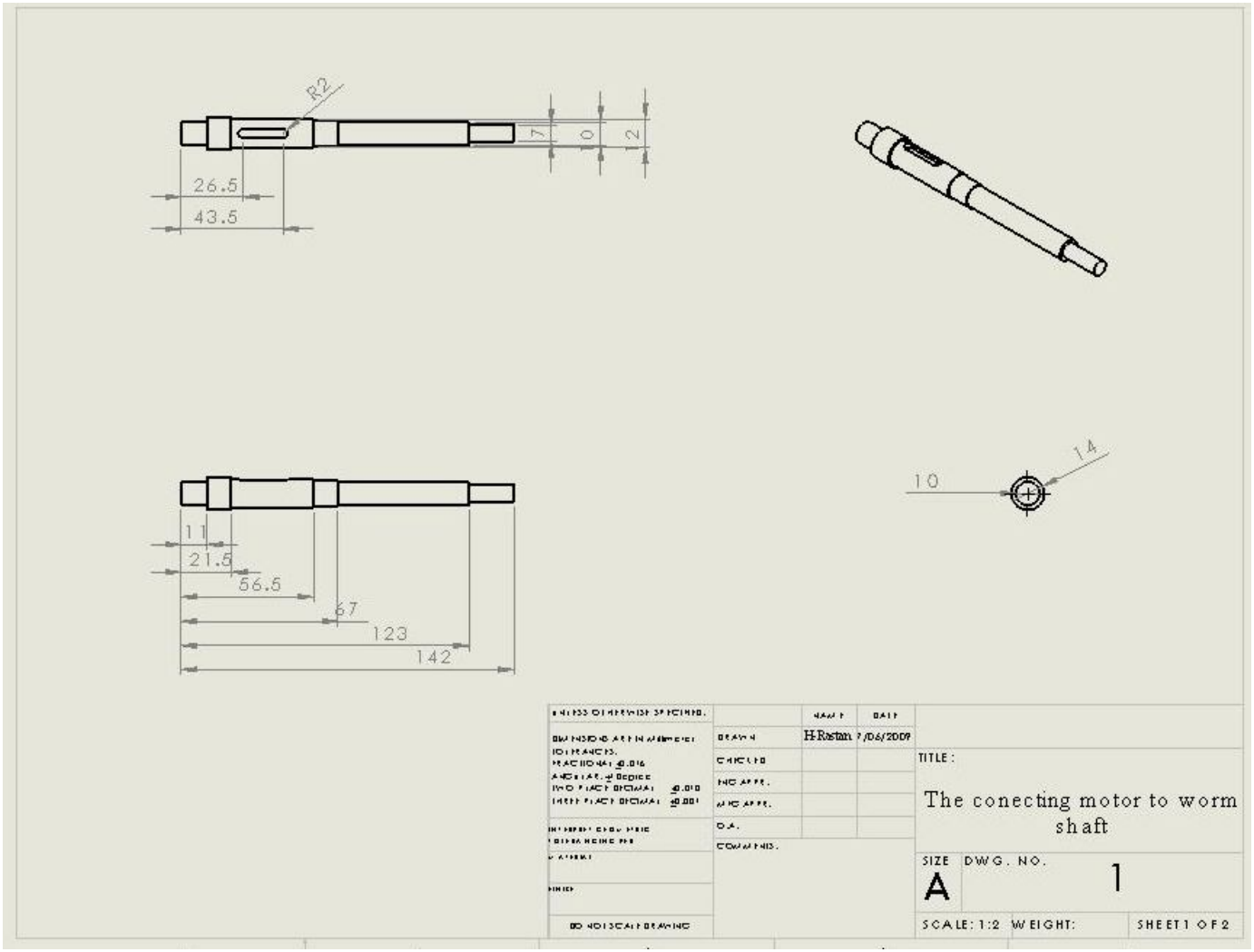
TITLE:

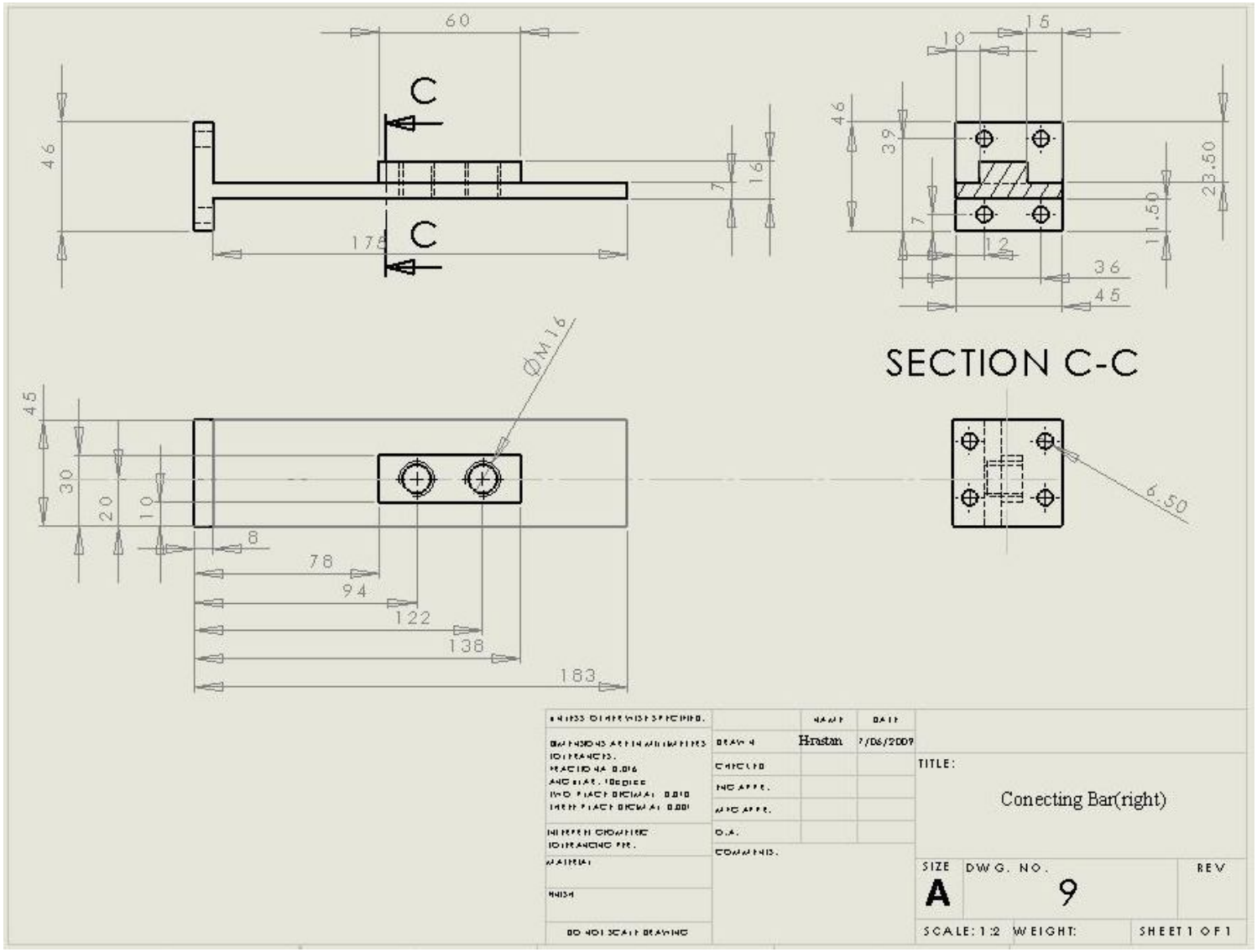
Ideler Assembly

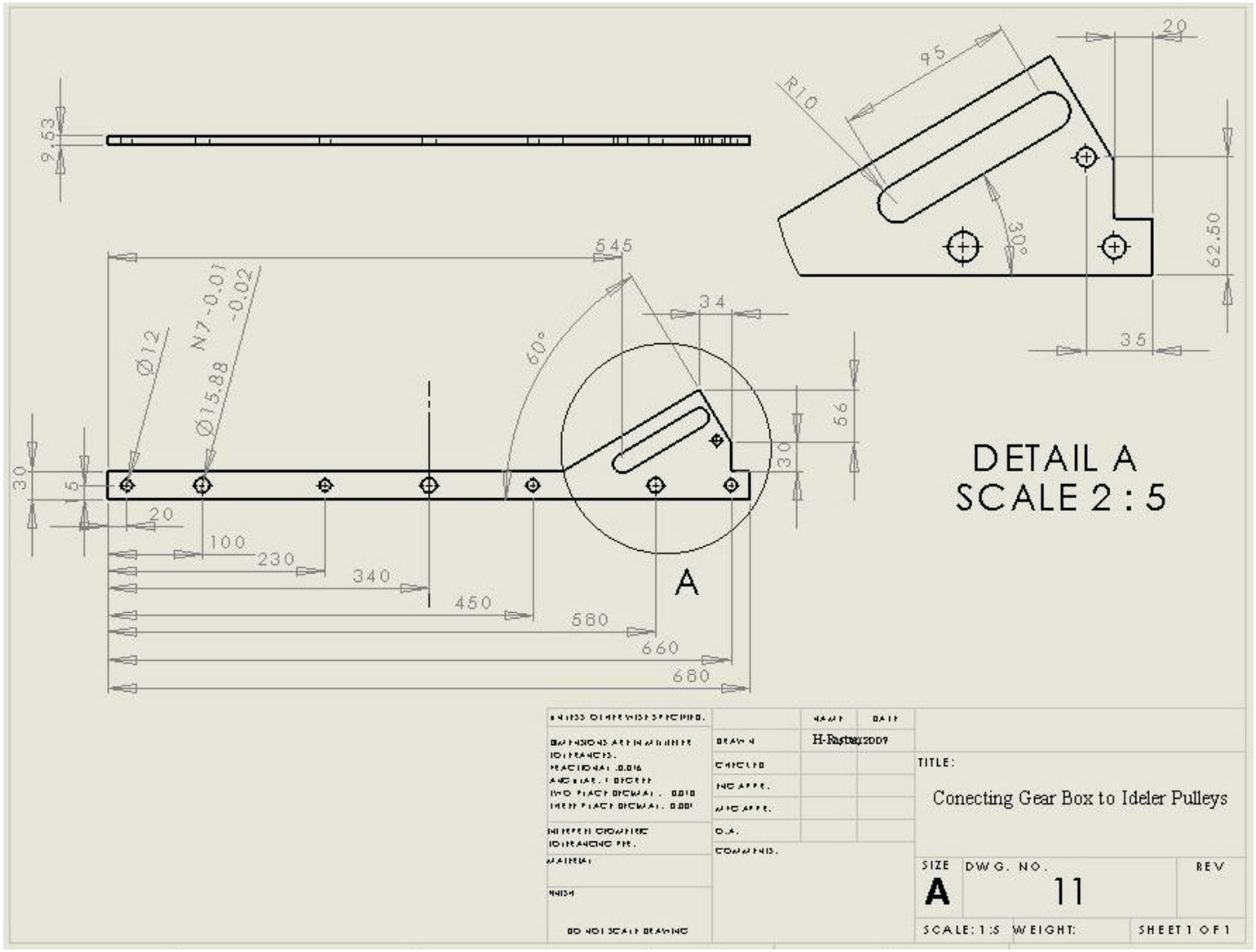
SIZE	DWG. NO.	REV

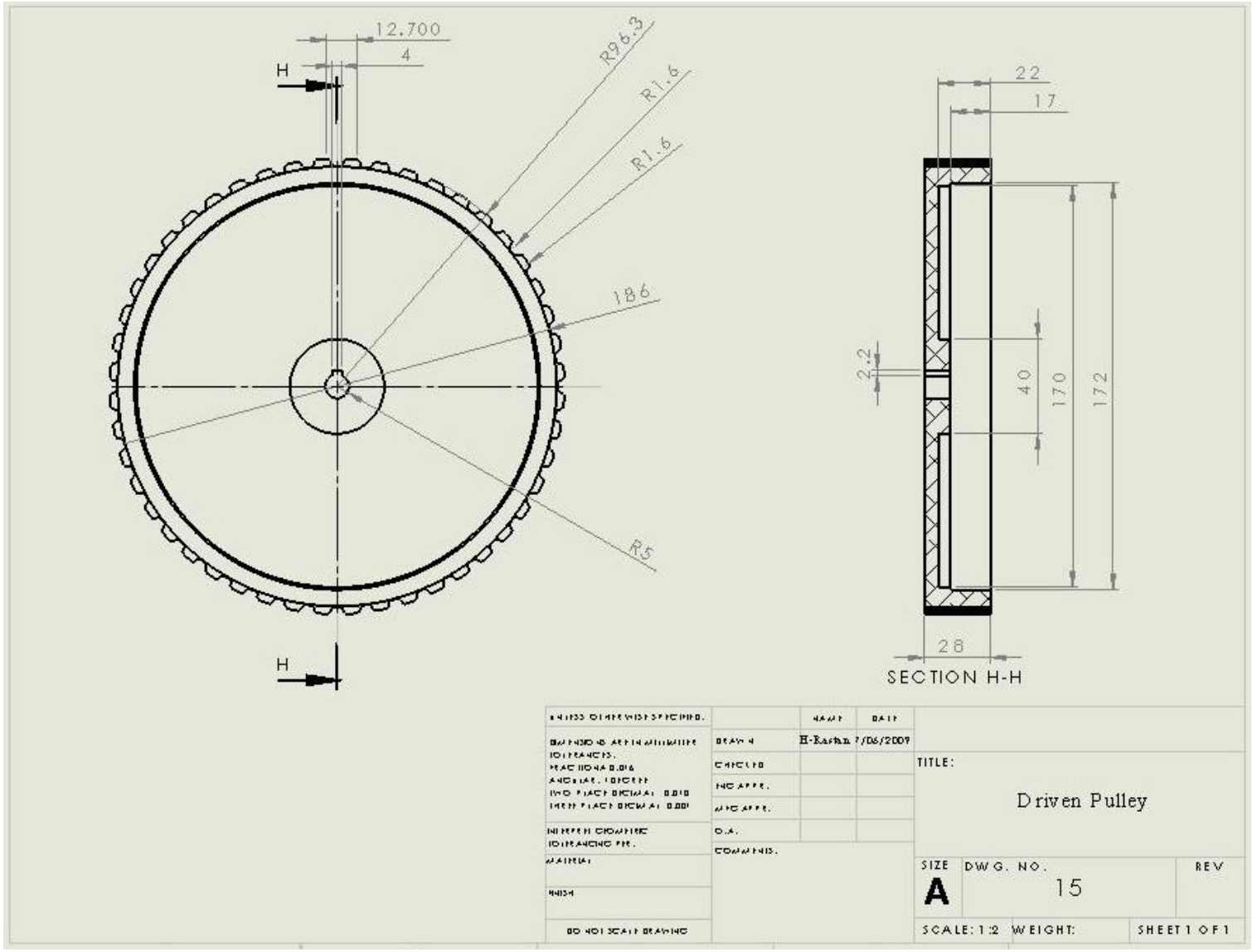
SCALE: 1:2	WEIGHT:	SHEET 1 OF 1

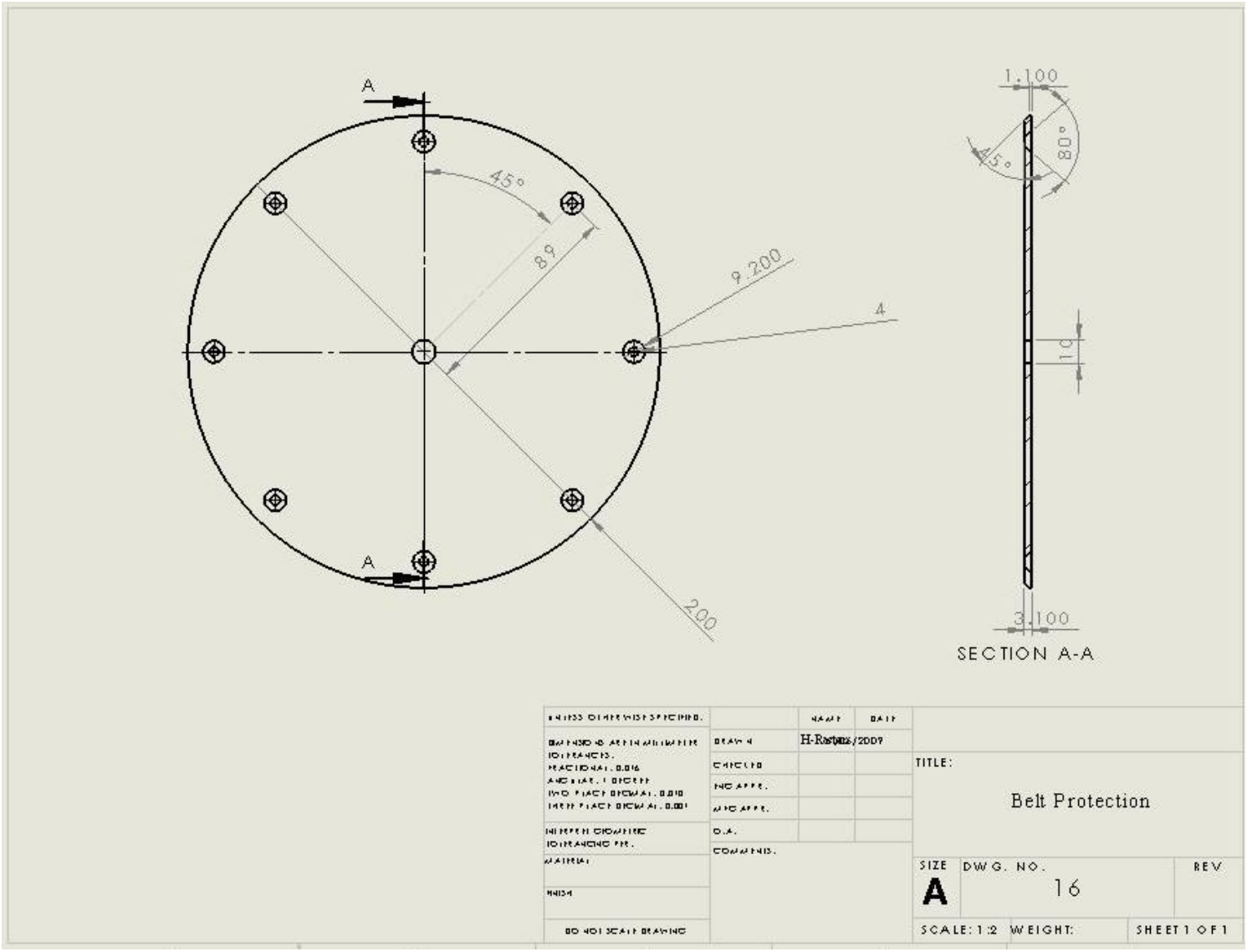
5 4 3 2 1

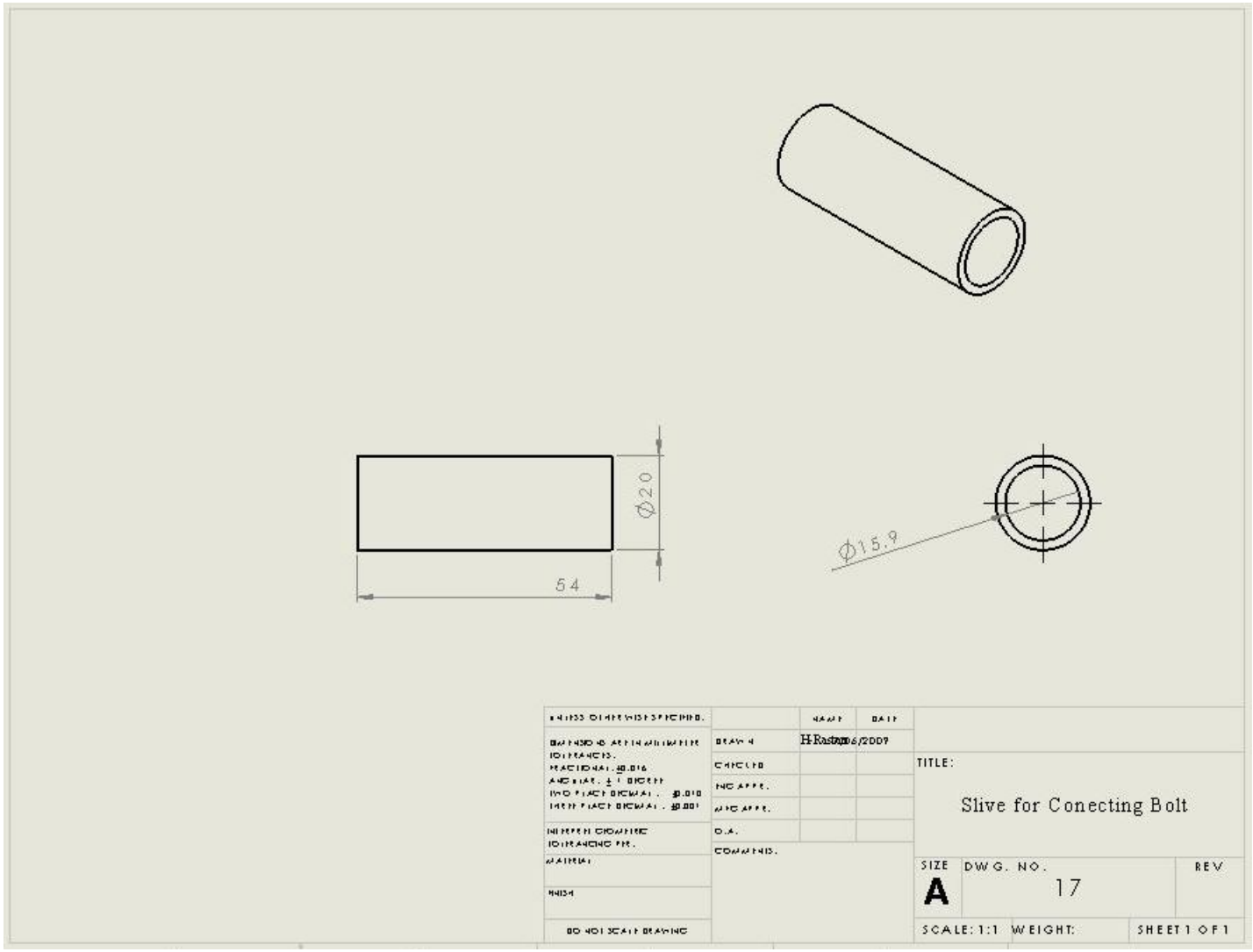

R2
7
10
12
26.5
43.5
14
10
11
21.5
56.5
67
123
142
H-Rastam 7/06/2009
TITLE :
The conecting motor to worm shaft
SIZE
A
DWG. NO.
1
SCALE: 1:2
WEIGHT:
SHEET 1 OF 2

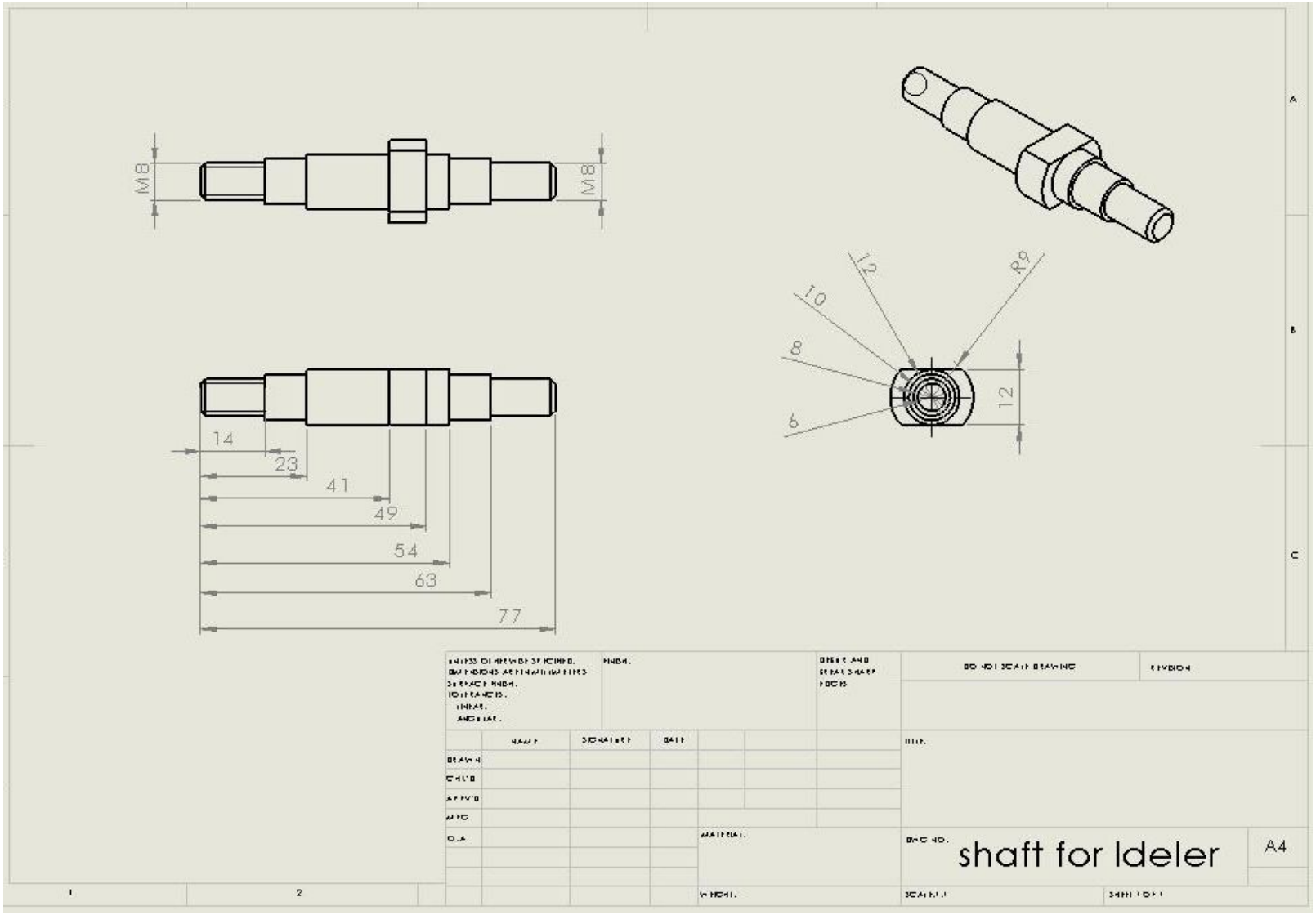

60
C
C
46
16
7
178
15
10
46
39
23.50
11.50
7
12
36
45
SECTION C-C
6.50
ØM16
45
30
20
10
8
78
94
122
138
183
NAME
DATE
Hrastan
7/06/2009
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS.
TITLE:
Conecting Bar(right)
SIZE
A
DWG. NO.
9
REV
SCALE: 1:2
WEIGHT:
SHEET 1 OF 1

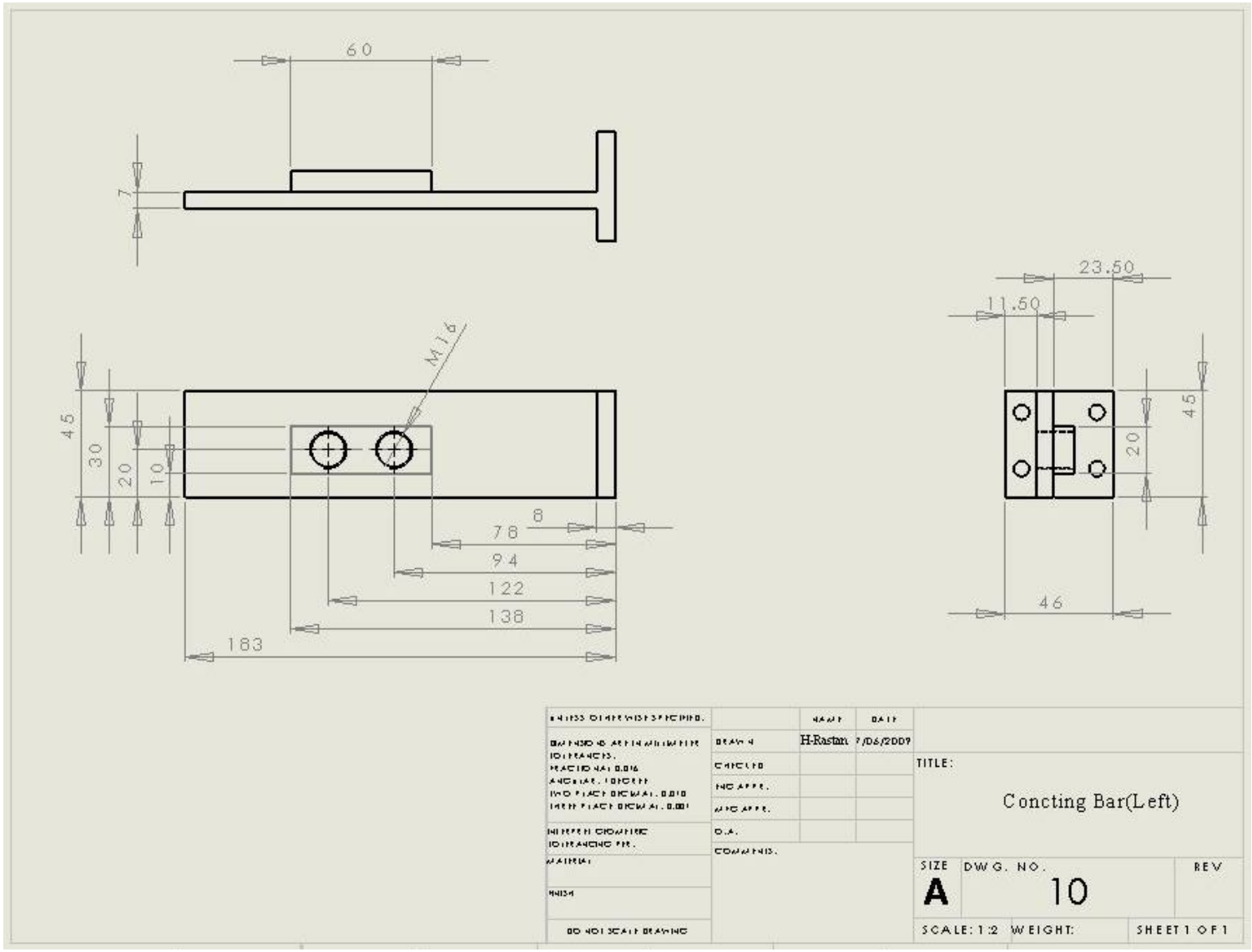

9.53
545
Ø12
N7 -0.01 -0.02
Ø15.88
60°
34
56
30
15
20
100
230
340
450
580
660
680
A
R10
95
20
30°
62.50
35
DETAIL A
SCALE 2 : 5
DRAWN
H-Foster
2009
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS.
MATERIAL
DO NOT SCALE DRAWING
TITLE:
Conecting Gear Box to Ideler Pulleys
SIZE
A
DWG. NO.
11
REV
SCALE: 1:5
WEIGHT:
SHEET 1 OF 1

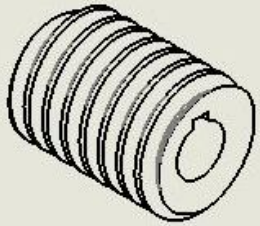

12.700
4
H
R96.3
R1.6
R1.6
186
R5
H
22
17
2.2
40
170
172
28
SECTION H-H
NAME
DATE
DRAWN
H-Rafhan
7/06/2009
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Driven Pulley
SIZE
A
DWG. NO.
15
REV
SCALE: 1:2
WEIGHT:
SHEET 1 OF 1

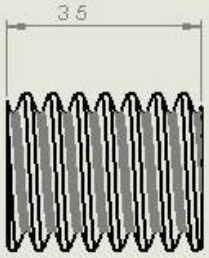

A
45°
8.9
9.200
4
2.00
A
1.100
80°
45°
1.0
3.100
SECTION A-A
DIMENSIONS ARE IN MILLIMETERS
TOLERANCES:
FRACTIONAL ±0.016
ANGULAR: ± 0.5 DEGREE
TWO PLACE DECIMAL ±0.010
THREE PLACE DECIMAL ±0.001
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
H-Rastgoo
2009
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Belt Protection
SIZE
A
DWG. NO.
16
REV
SCALE: 1:2
WEIGHT:
SHEET 1 OF 1

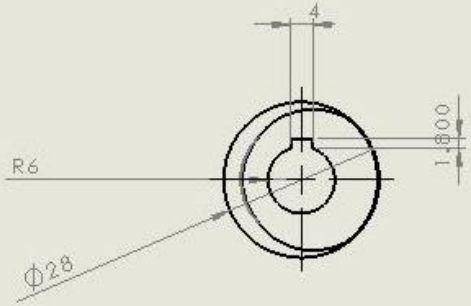

Ø20
54
Ø15.9
Slive for Conecting Bolt
A
17
SCALE: 1:1
SHEET 1 OF 1


M8
M8
14
23
41
49
54
63
77
12
10
8
6
R9
12
shaft for Ideler
A4


60
7
M16
45
30
20
10
8
78
94
122
138
183
23.50
11.50
45
20
46
TITLE:
Concting Bar(Left)
SIZE
A
DWG. NO.
10
REV
SCALE: 1:2
WEIGHT:
SHEET 1 OF 1
DRAWN
H-Rastan
7/06/2009
DO NOT SCALE DRAWING







Lead Anglae	3 °26'
Module	1.5
Pitch Diamiter	25 mm
Material	Steel

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
TOLERANCES:
FRACTIONAL ±0.016
ANGULAR: ± °
TWO PLACE DECIMAL ±0.010
THREE PLACE DECIMAL ±0.001
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
NEXT
DO NOT SCALE DRAWING

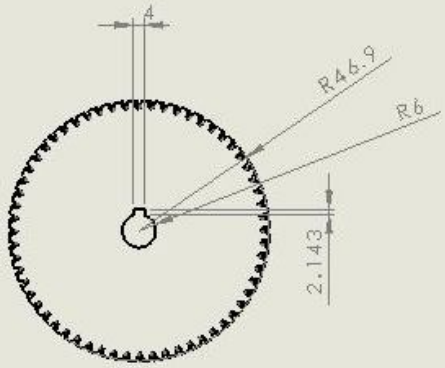
	NAME	DATE
DRAWN	H-Rostom	7/06/2009
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		
COMMENTS:		

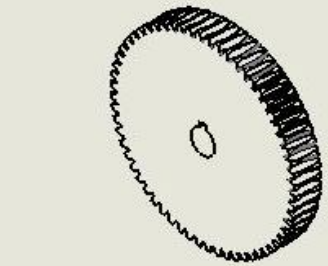
TITLE:

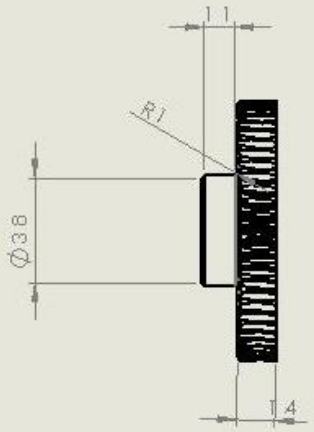
Worm

SIZE	DWG. NO.	REV
A	19	

SCALE: 1:1 WEIGHT: SHEET 1 OF 1







Pressure Angle is 20°

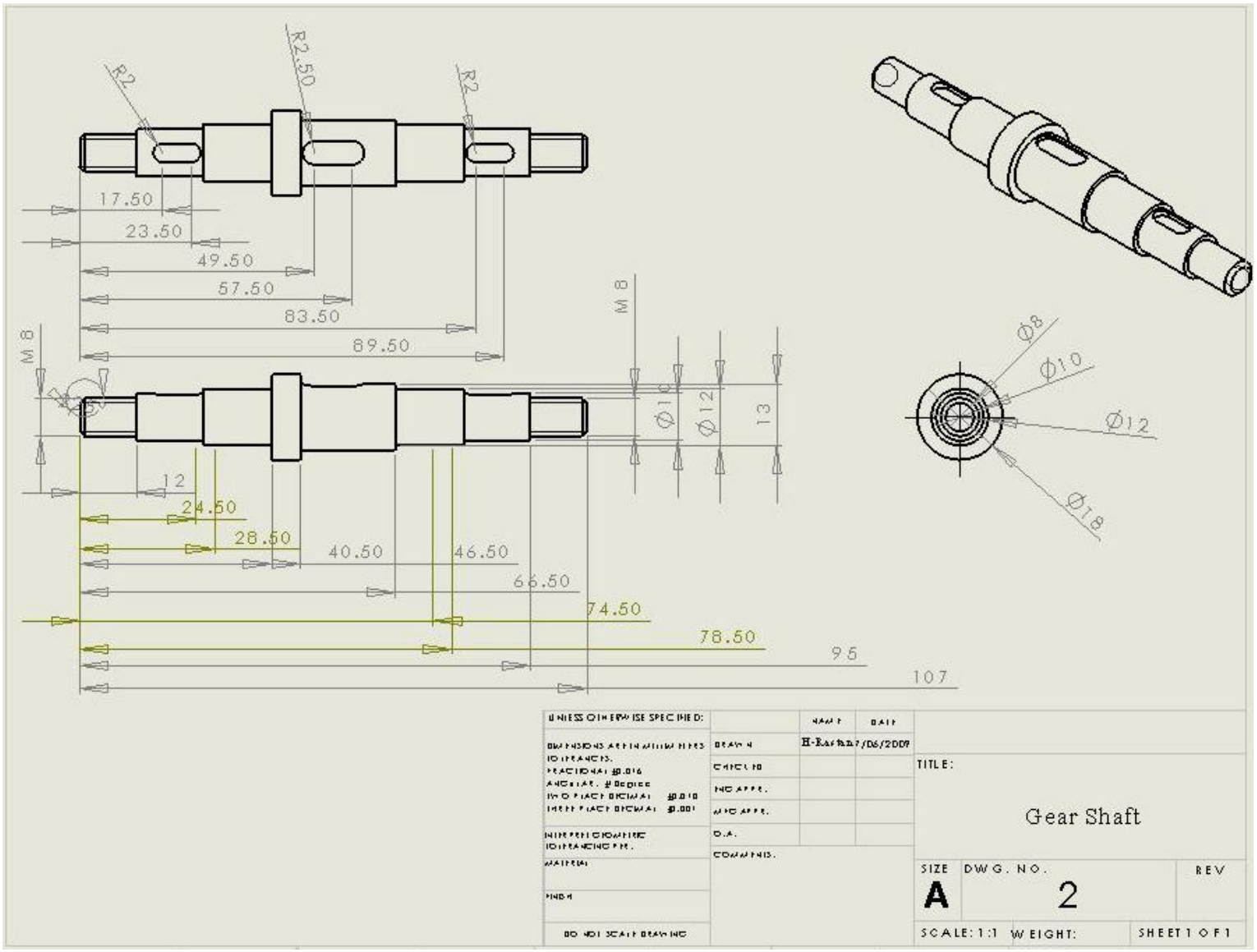
Madule	1.5
Lead Angle	3° 26'
Number of Teeth	60
Pitch Diamiter	90

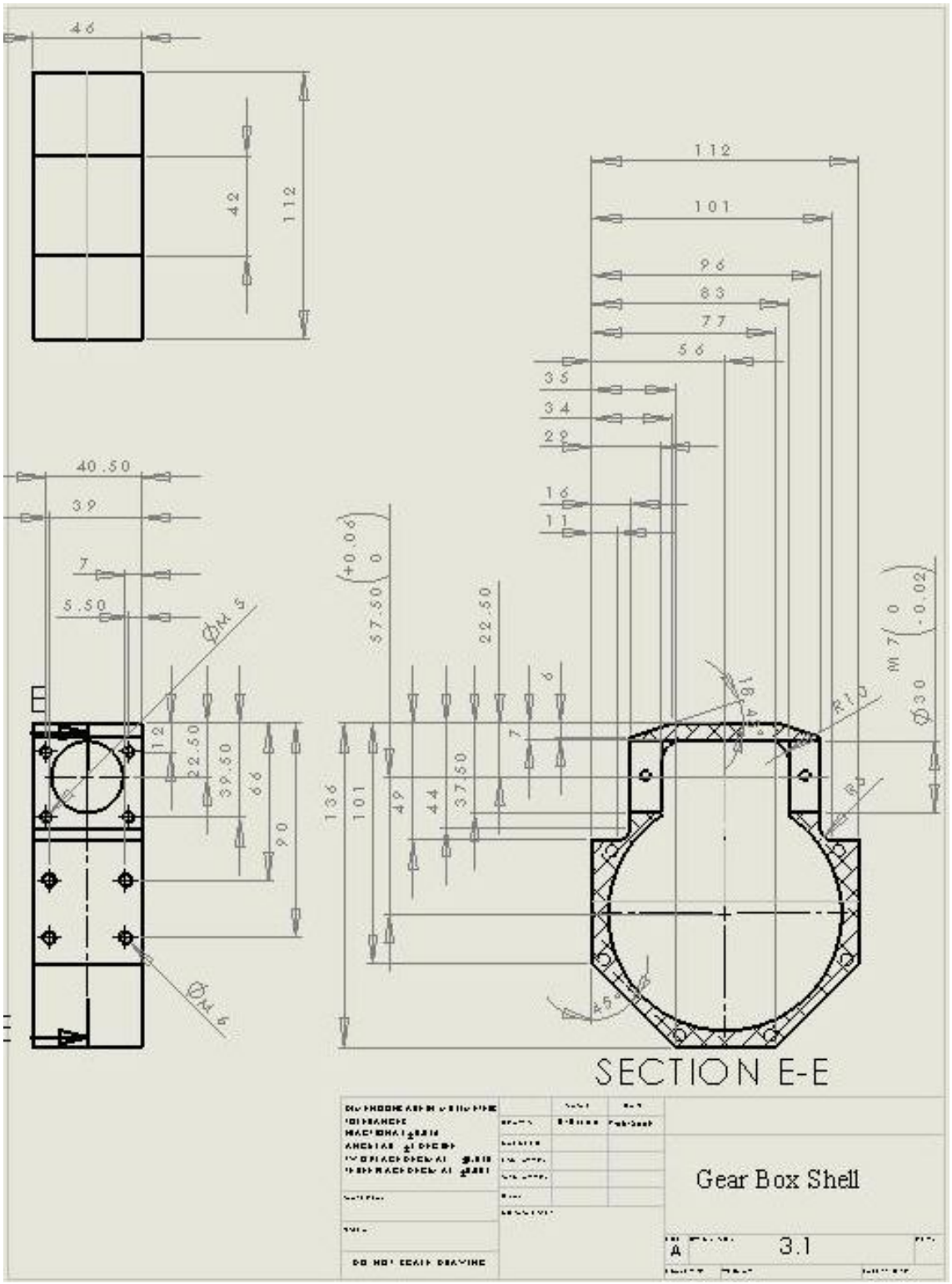
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
TOLERANCES:
FRACTIONAL ±0.016
ANGULAR: ± °
TWO PLACE DECIMAL ±0.010
THREE PLACE DECIMAL ±0.001
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL:
Cast Iron

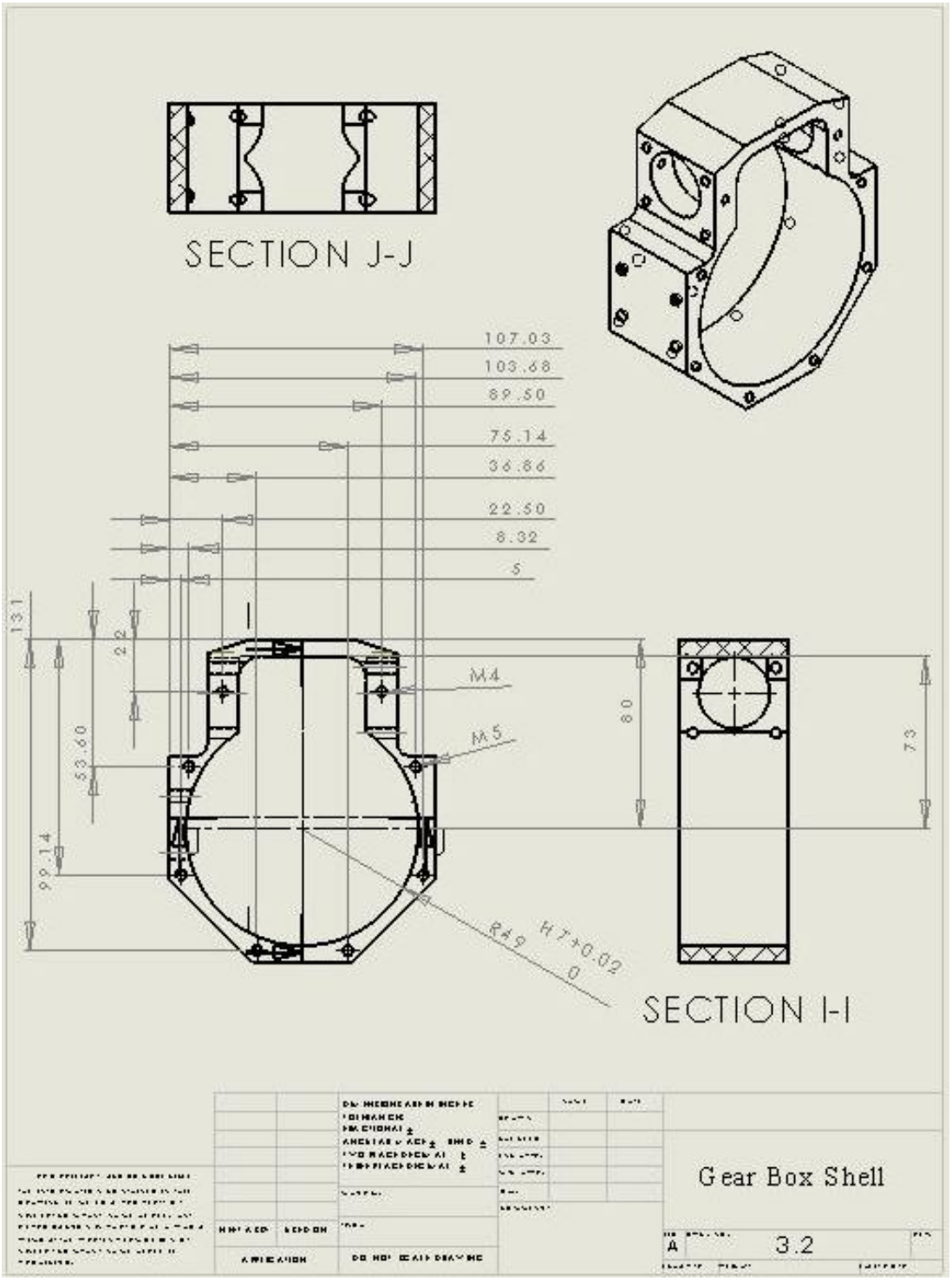
	NAME	DATE
DRAWN	H-Rostom B	05/2009
CHECKED		
ENG APPR.		
MFG APPR.		
Q.A.		
COMMENTS:		

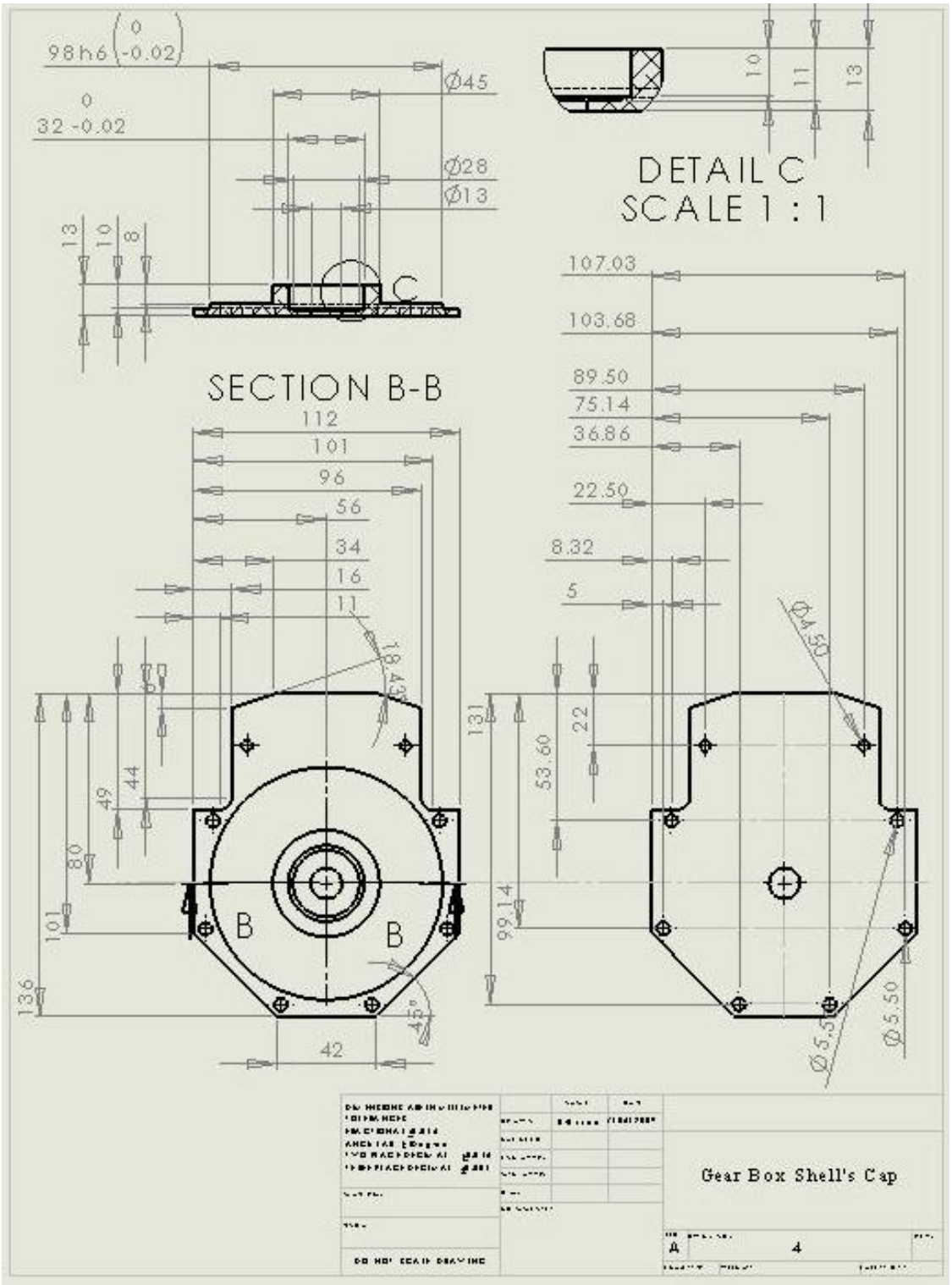
TITLE:
Worm Gear

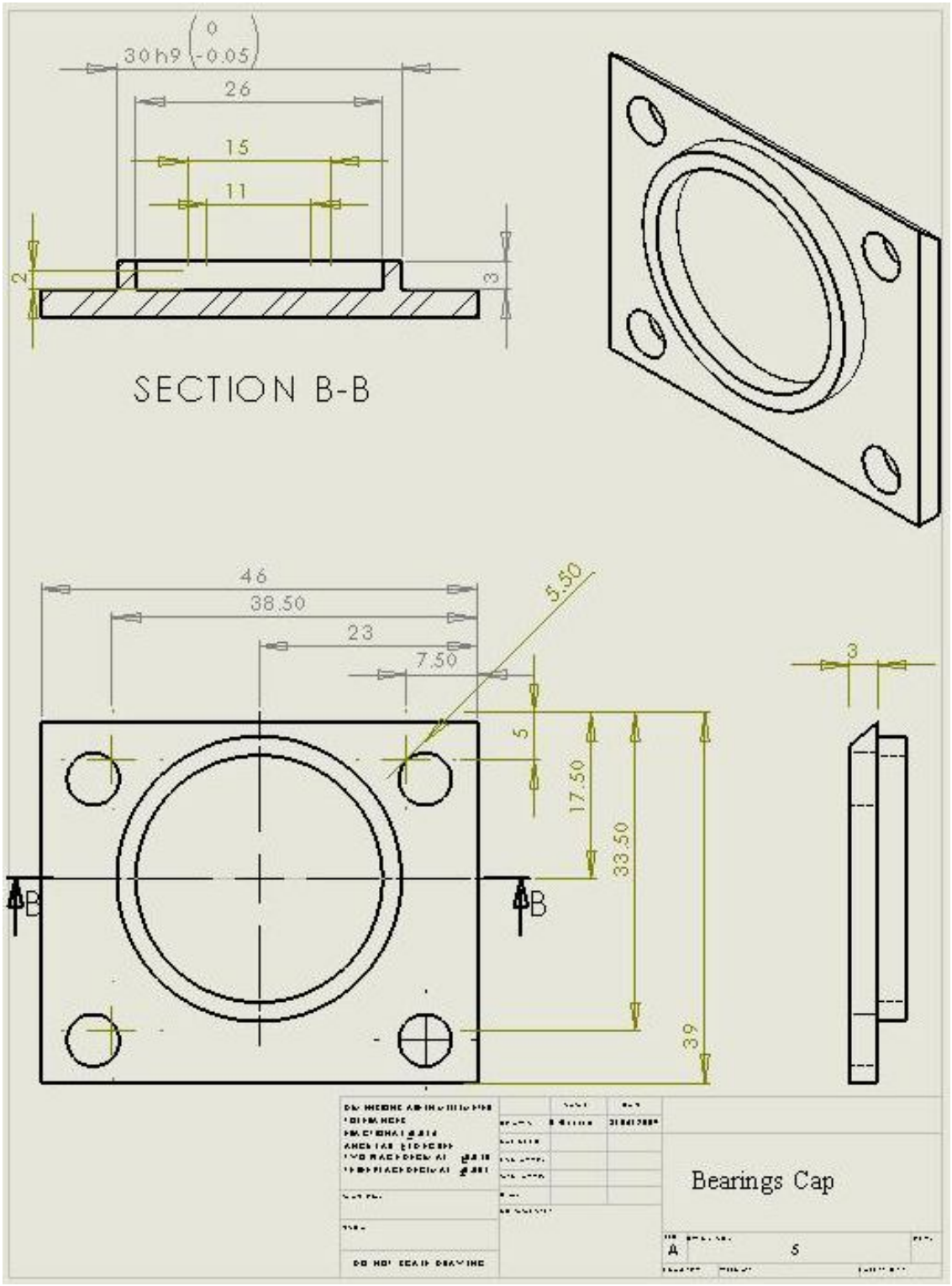
SIZE A | DWG. NO. 20 | REV
SCALE: 1:2 | WEIGHT: | SHEET 1 OF 1

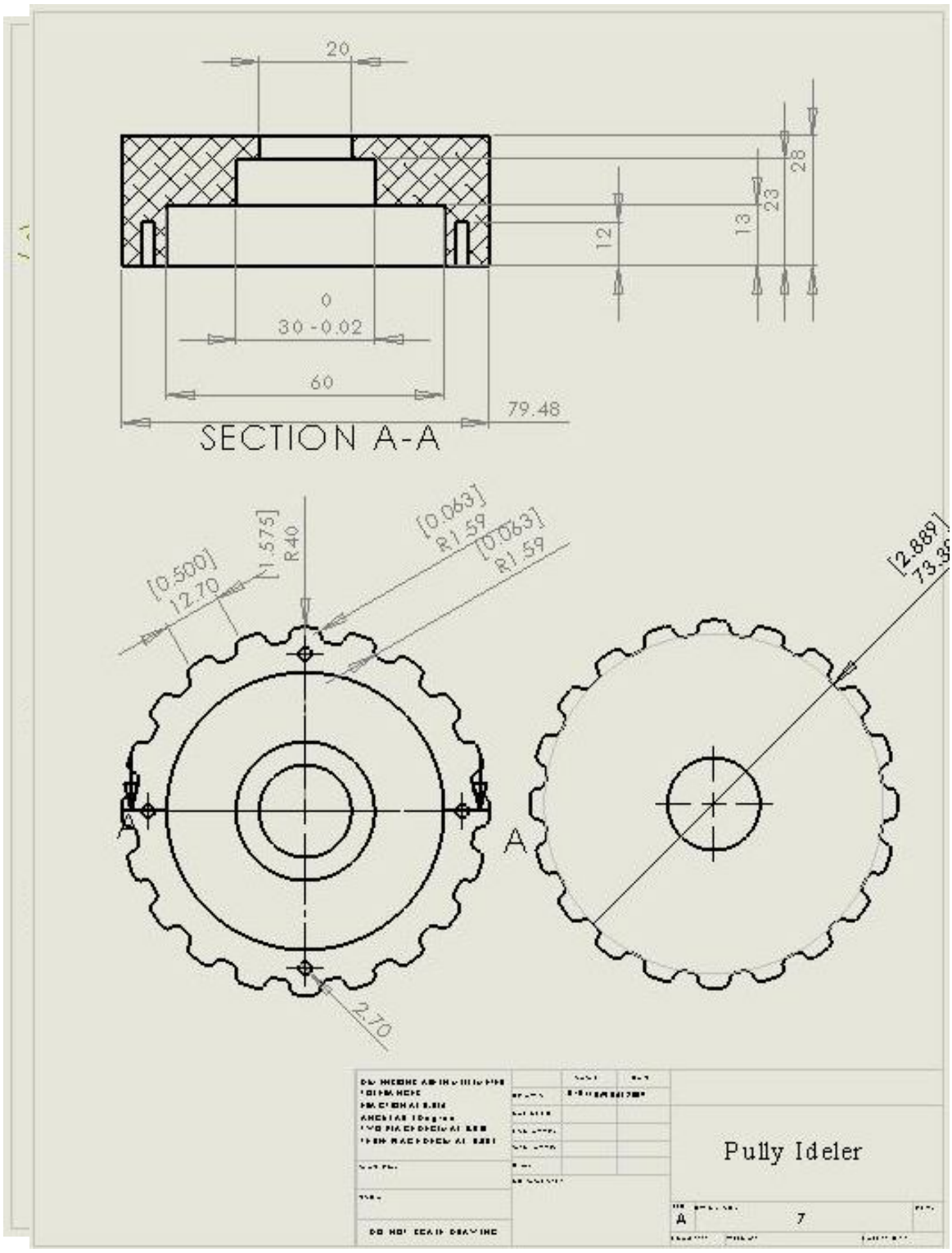

R2
R2.50
R2
17.50
23.50
49.50
57.50
83.50
89.50
M 8
M 8
Ø1
Ø12
13
12
24.50
28.50
40.50
46.50
66.50
74.50
78.50
95
107
Ø8
Ø10
Ø12
Ø18
UNLESS OTHERWISE SPECIFIED:
DRAWN
H-Rafhan 7/06/2009
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
NAME
DATE
TITLE:
Gear Shaft
SIZE
A
DWG. NO.
2
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1

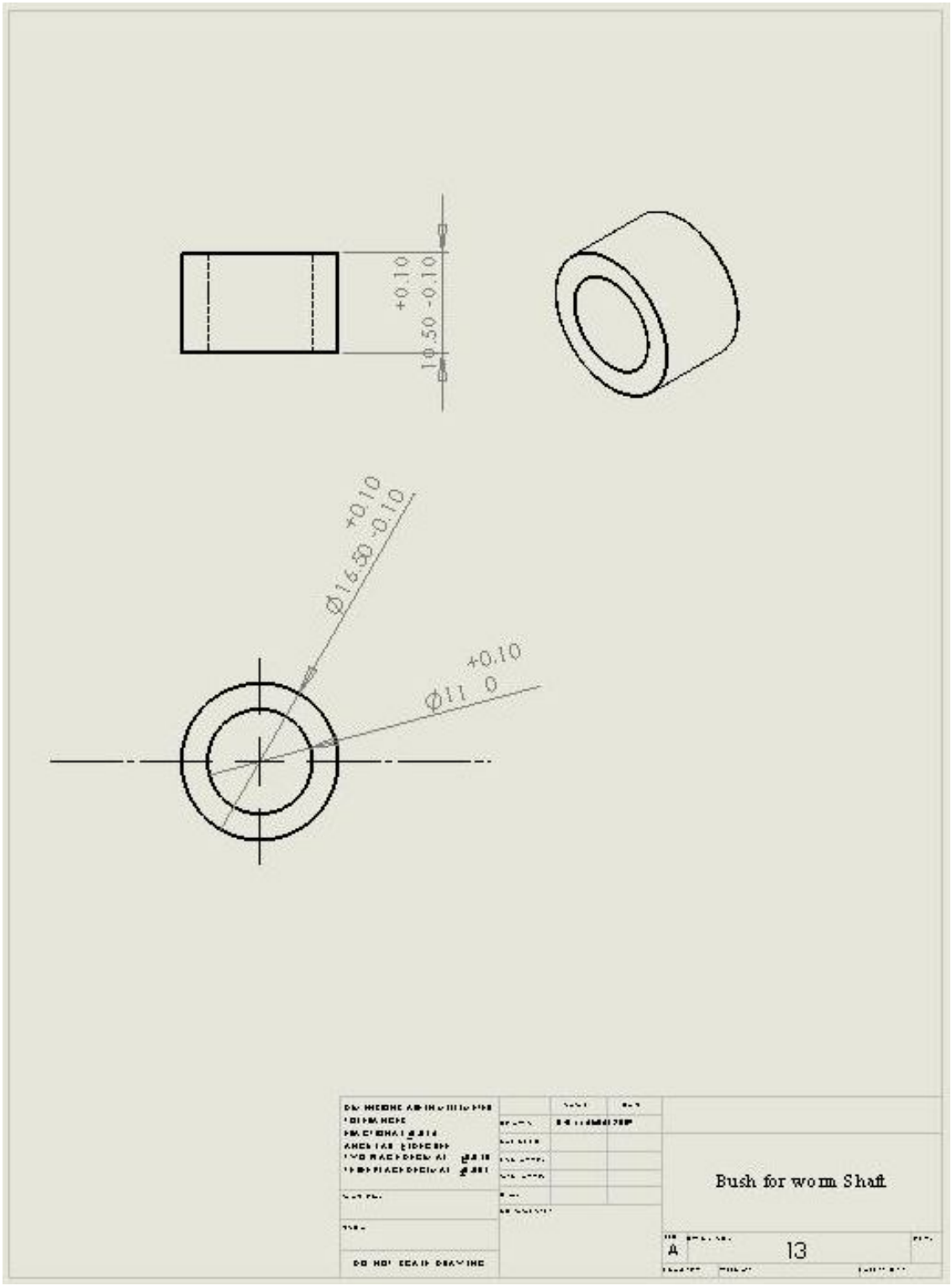

SECTION E-E
Gear Box Shell
3.1

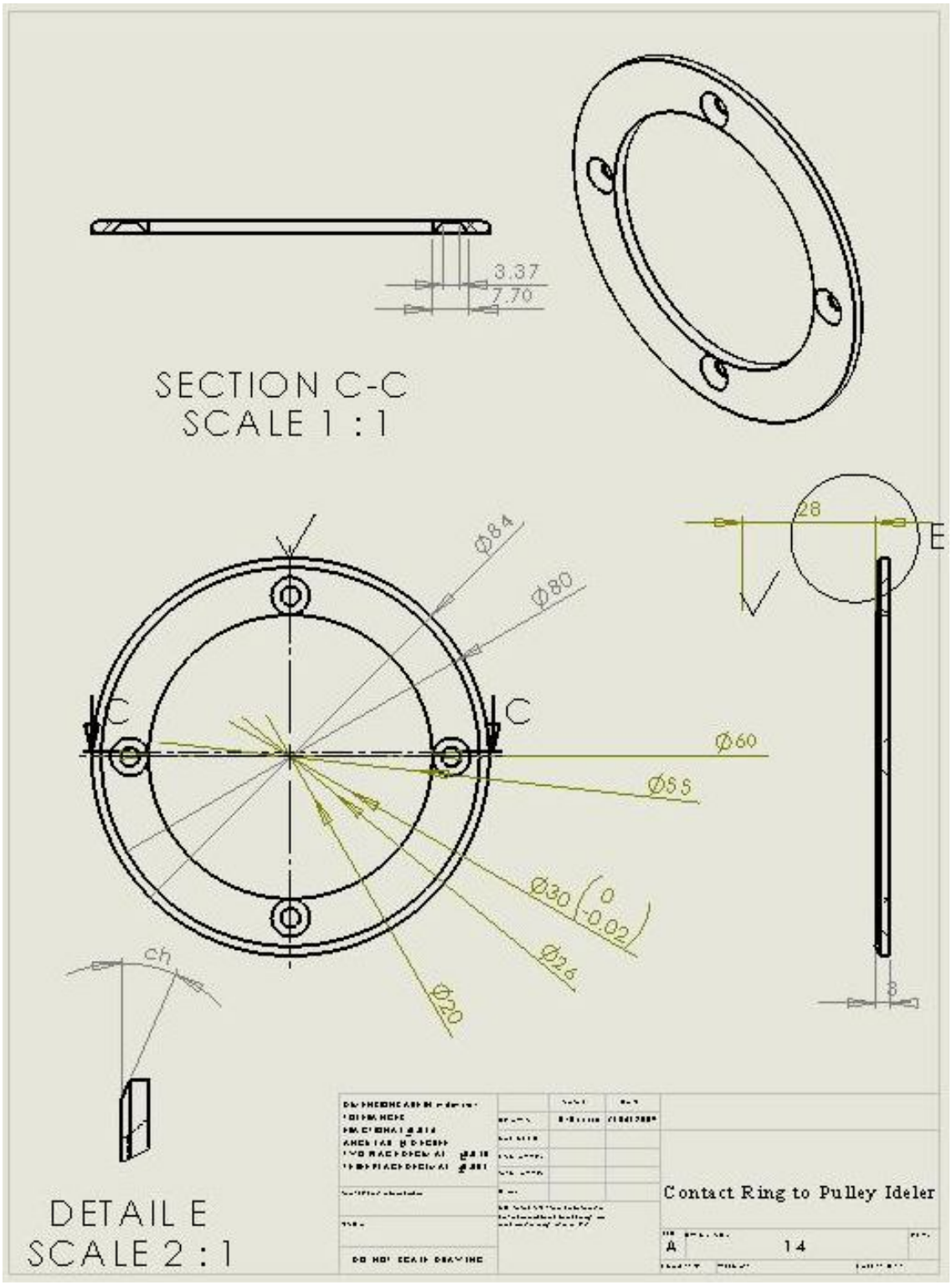

SECTION J-J
107.03
103.68
89.50
75.14
36.86
22.50
8.32
5
131
22
53.60
99.14
M4
M5
80
73
R49
H7+0.02
0
SECTION I-I
Gear Box Shell
A
3.2

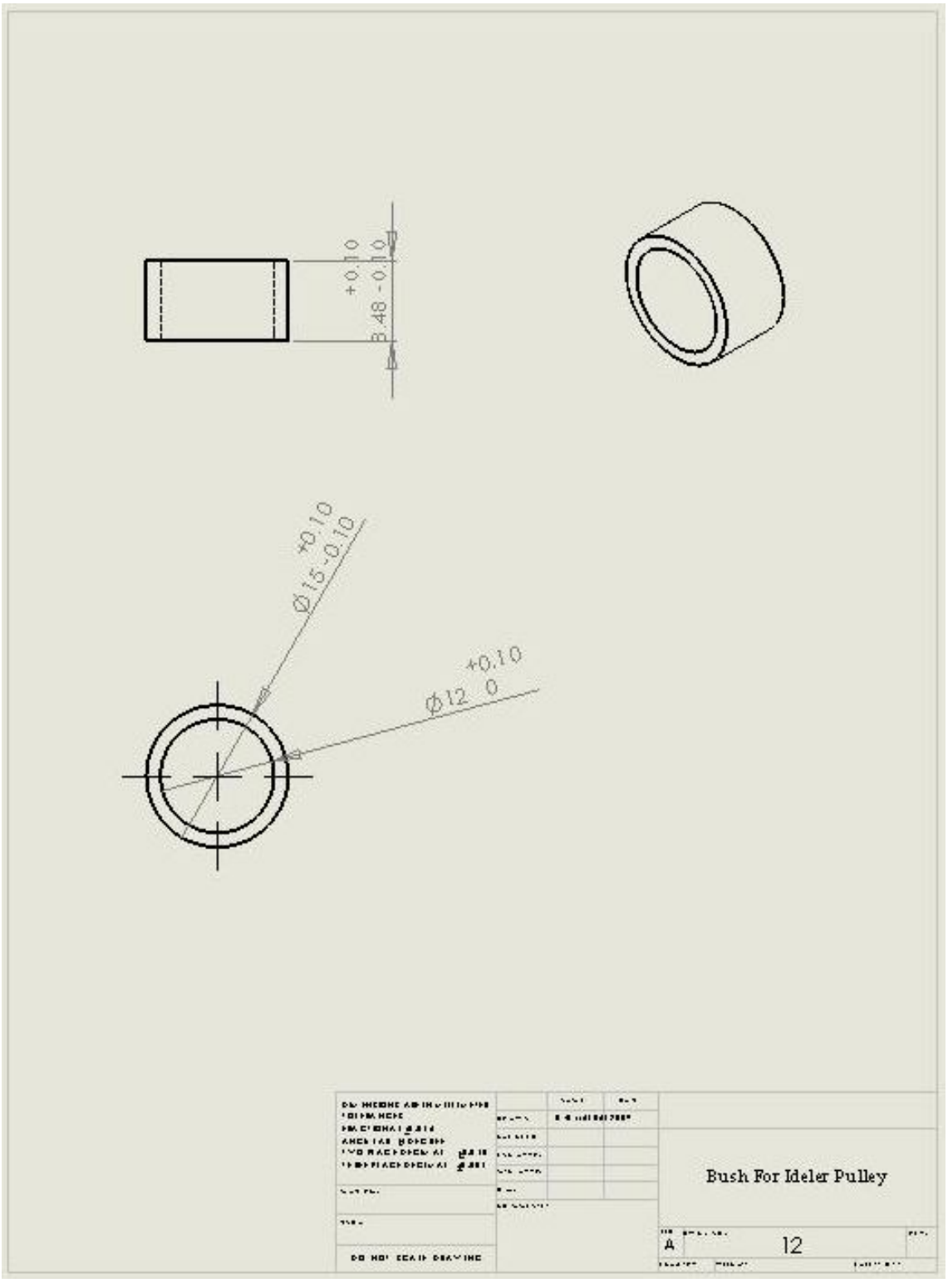

98h6 (0 / -0.02)
32 -0.02
Ø45
Ø28
Ø13
13
10
8
C
SECTION B-B
112
101
96
56
34
16
11
18.43°
65
49
44
80
101
136
B
B
45°
42
DETAIL C
SCALE 1 : 1
10
11
13
107.03
103.68
89.50
75.14
36.86
22.50
8.32
5
Ø4.50
131
22
53.60
99.14
Ø5.50
Ø5.50
Gear Box Shell's Cap
A
4

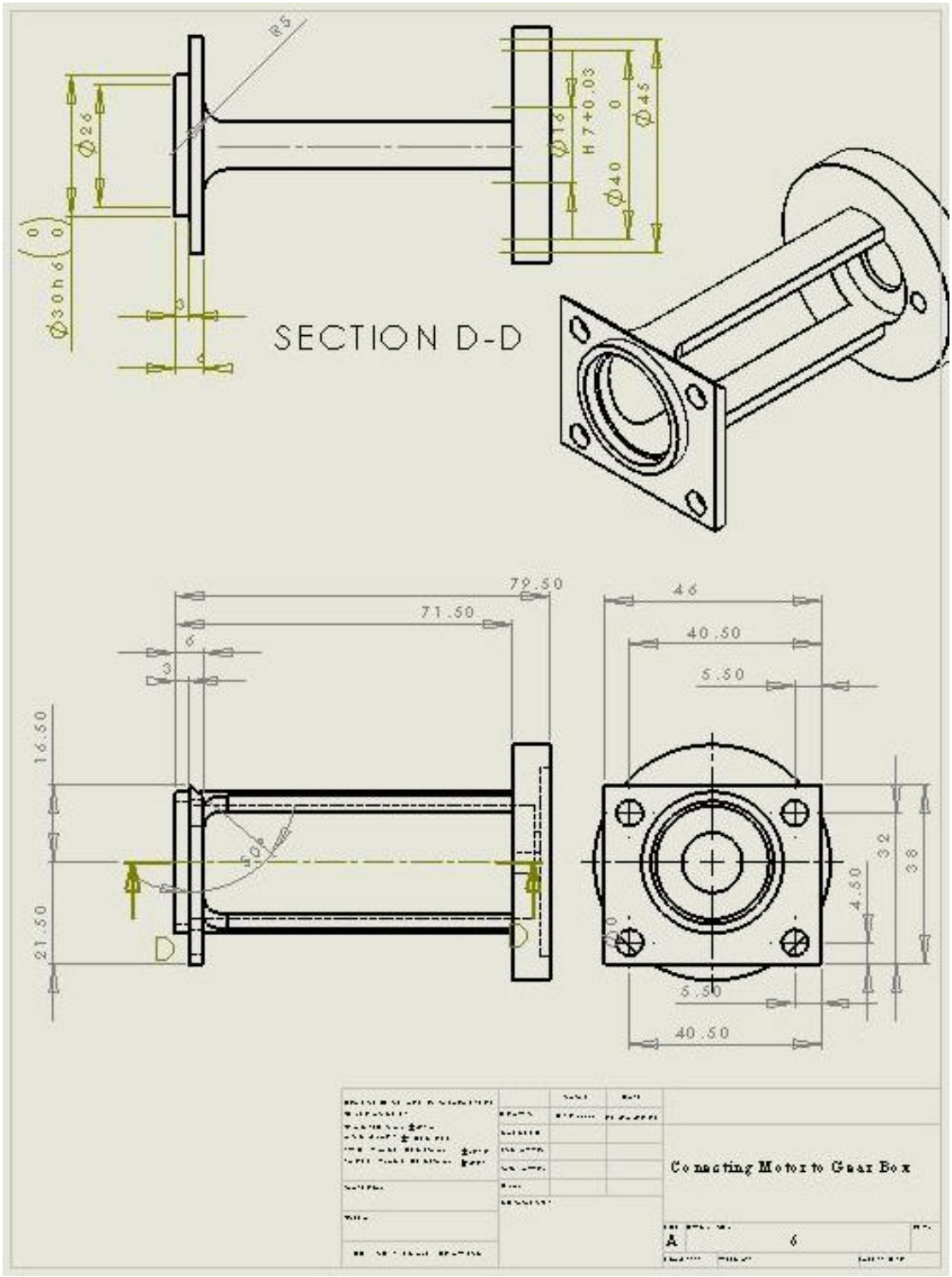

30h9 (0 -0.05)
26
15
11
2
3
SECTION B-B
46
38.50
23
7.50
5.50
5
17.50
33.50
39
3
B
B
Bearings Cap
A
5


20
28
23
13
12
0
30 -0.02
60
79.48
SECTION A-A
[0.500]
12.70
[1.575]
R40
[0.063]
R1.59
[0.063]
R1.59
[2.889]
73.38
A
A
2.70
Pully Ideler
A
7


+0.10
-0.10
16.50
+0.10
-0.10
Ø16.50
+0.10
0
Ø11
Bush for worm Shaft
A
13


3.37
7.70
SECTION C-C
SCALE 1 : 1
Ø84
Ø80
C
C
Ø60
Ø55
Ø30 (0 -0.02)
Ø26
Ø20
ch
28
E
8
DETAIL E
SCALE 2 : 1
Contact Ring to Pulley Ideler
A
14


+0.10
8.48 - 0.10
+0.10
Ø15 -0.10
+0.10
Ø12 0
Bush For Ideler Pulley
A
12


SECTION D-D
Connecting Motor to Gear Box
A
6

Appendix B

B.1 Engineering Design of the Tracked Robot

For designing the mechanical part of a tracked robot, it is important to define some design specifications and find out some salient design parameters. These include: the purpose of the design, the required power, how to minimize the energy consumption, what design and machining methods can be used to reduce the cost, how to minimize the size, and what are the critical stresses. Another important goal is to create the simplest possible design.

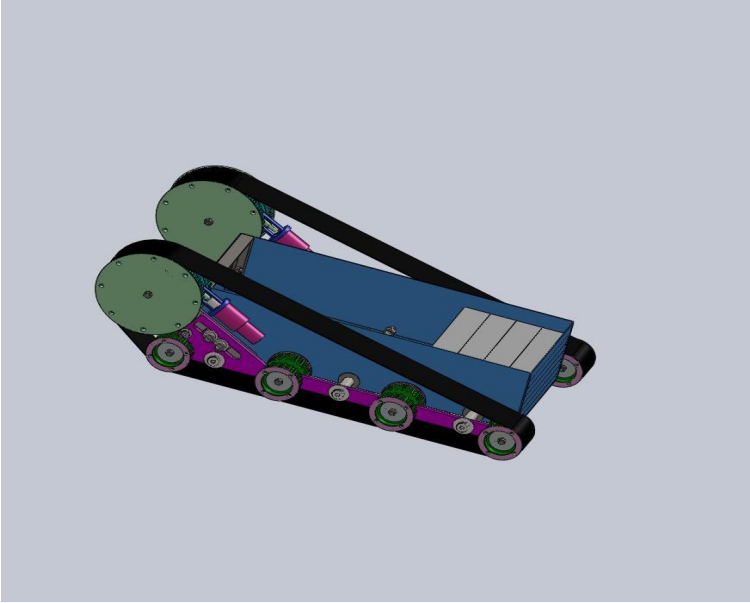


Figure B.1 : illustrates the manufactured prototype of tracked robot

The tracked robot's tasks are: moving on flat ground, climbing and descending stairs, and turning on flat ground. The maximum torque the robot's motors must provide occurs when the robot is climbing stairs that have the maximum specified angle. The SolidWorks *3D CAD* software was used to create the mechanical design. Mechanical engineering design is an iterative process; Solid Works speeds up these tasks, and allows trial designs to be made much quicker than would otherwise be possible.

For a first analysis, a set of stairs with a 45 degree slope was chosen. This is on the upper end of the slopes allowed by the North American standards, and should provide a good first case analysis.

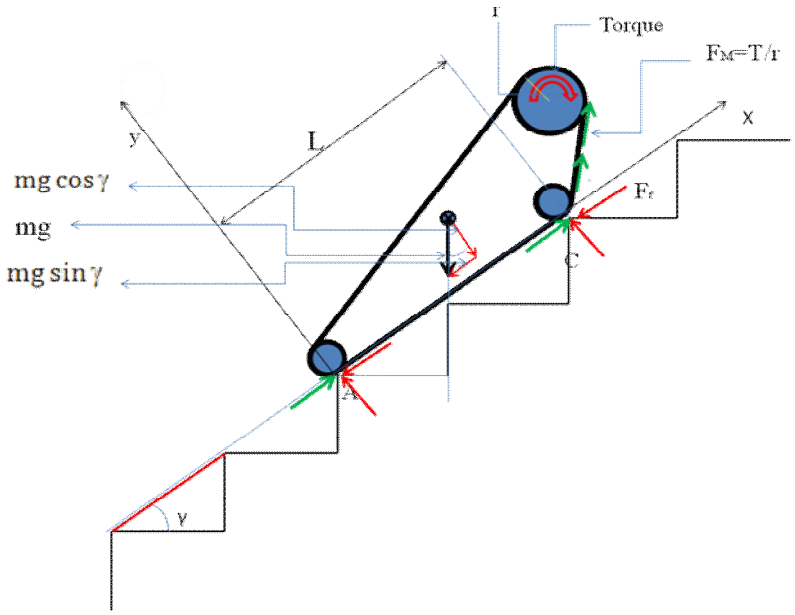


Figure B.2 : Illustrates the minimum force and torque for overcoming the opposed forces while the tracked robot starts to climb stairs.

The force required to move the track robot up the stairs is F_M (in the direction of the track length). F_r is the tangential resistance force, and F_V is the component of the gravitational force in the track direction. The movement force must equal the combined resistance and gravitational forces:

$$F_M = F_r + F_V \qquad B.\ 1$$

And:

$$F_r = \mu mg\ cos\ \gamma \qquad B.\ 2$$

$$F_V = mg\ sin\ \gamma \qquad B.\ 3$$

$$F_M = mg(sin\ \gamma + \mu\ cos\ \gamma) \qquad B.\ 4$$

The tracked robot moves at a constant speed, v. The required power, P, is:

$$p = T_G \omega \qquad B.\ 5$$

Typically *DC* motors require a gear box to reduce the speed and increase the output torque. The force applied by the motor/gear box combination is:

$$F_G = \frac{T_G}{r_G} \qquad B.\ 6$$

Where T_G is the output torque from the gearbox, r is the radius of the pulley, and r_G is the radius of the last gear.

B.2 The Type of Gearbox

The type of gearbox chosen for the robot is the worm gear. There are three main reasons for choosing this type of gear box.

First, the worm gear has a greater resistance to backlash than the other gear designs. If the lead angle is small, the circumference force to the worm could not easely overcome friction. This helps prevent the robot from slipping down the stairs when the gear is applying a force to the worm gear, instead of the usual case where the worm gear applies a force to the gear.

Second, designing this type of gear box is a much easier task than designing several gears for reducing the speed of the DC motor and increasing its torque.

Third, the worm gear is convenient for sepecial design which the gear box can be mounted in the center of a driven pulley. This reduces the required robot size and also transfers the torque directly from the gear box to the track, as shown in Figures B.3 and B.4.

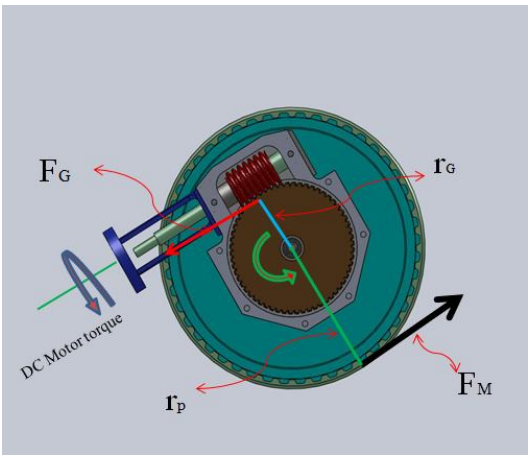
The main disadvantage of a worm gear is its low efficiency. Worm gear power efficiency is among the lowest of any common gear box design. Another disadvantage is that the gears are not very strong against sudden shocks because of the small gear addendum.This must be considered when specifying the worm gear mechanism, to reduce the possibility of the gear teeth breaking.

Despite these problems, the worm gear is overall the most appropriate gear box for this purpose.

B.3 Gear Box Forces Analysis

As illustrated in Figure B.3, the motor torque is applied directly to the pulley, and then to the track. The forces applied to the gear must be analyzed. The gear size and material can then be chosen

.



.

Figure B.3 Illustrates the torque applied by the DC motor, and the output force, F_M

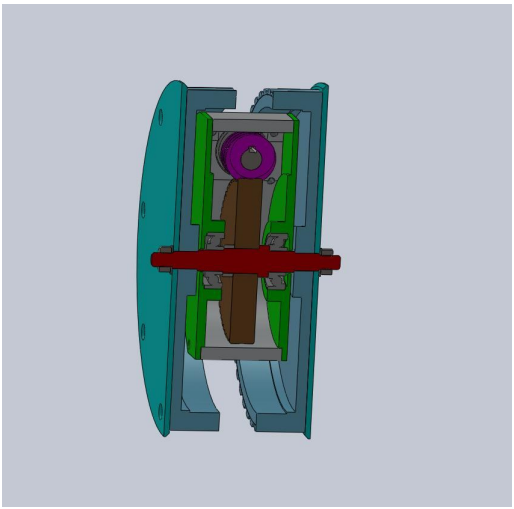


Figure B.4 : illustrates another view of the gear box.

By calculating the forces F_M and F_G from Equations *B.4* and *B.6*, it is possible to calculate the tangential, axial and radial forces on the gear from a free body diagram. From Equations B.7, B.8, and B.9, the forces on the worm gear in the *X*, *Y*, and *Z* directions, and the reaction forces on the bearing can be calculated:

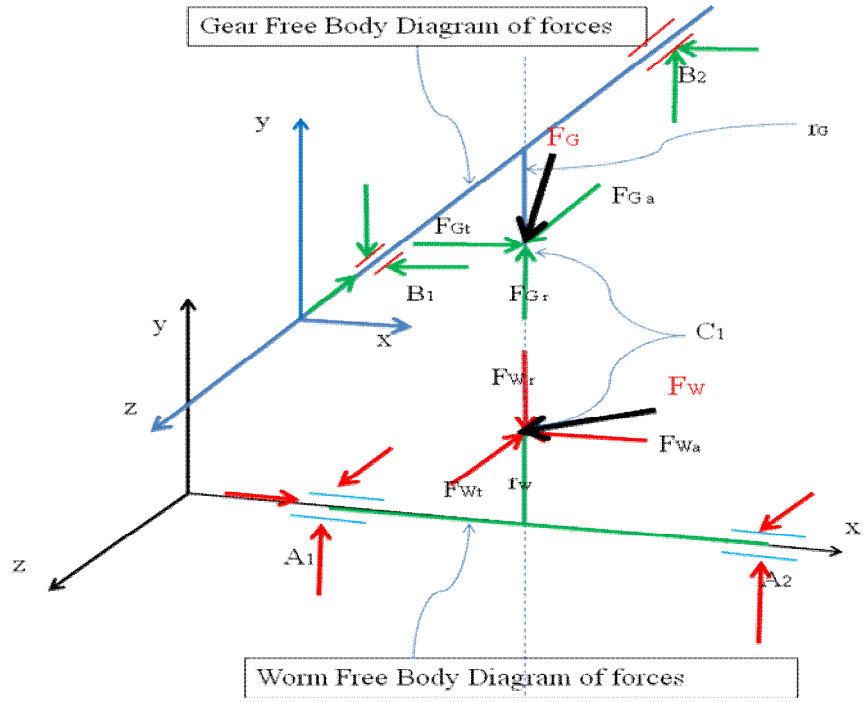


Figure B.5 : Free bodies of applied forces to the worm and the gear.

$$F_{Wt} = -F_{Ga} \qquad B.\ 7$$

$$F_{Wa} = -F_{Gt} \qquad B.\ 8$$

$$F_{Wr} = -F_{Gr} \qquad B.\ 9$$

The allowable gear bending stress is:

$$\sigma = \frac{K_v W^t}{F_m Y} \qquad B.\ 10$$

The allowable tangential force is:

$$(W^t)_{all} = C_s D_m{}^{0.8} F_e C_v \qquad B.\ 11$$

The efficiency of the worm gear is obtained from the equation:

$$\xi = \frac{\cos\phi_n - f\tan\lambda}{\cos\phi_n + f\tan\lambda} \qquad B.\ 12$$

The type of ball bearings for the worm gear shaft and the gear shaft can be determined after their axial loads have been calculated. These can be estimated:

$$\frac{F_a}{C_o} \approx e \qquad B.\ 13$$

$$F_e = X_2 V F_r + Y_2 F_a \qquad B.\ 14$$

$$L_{10} = \frac{60 L_R n_R}{60 n_D}\left(\frac{C_{10}}{F_e}\right)^a \qquad B.\ 15$$

The formula for designing the Tracked Robot is from Shgley's Mechanical Engineering Design, Richard G.Budynas and J. Keith Nisbett Eight Edition 2008.

Table B.1 : Worm and Gear data

Worm Gear-Right Hand		
1	Module	1.5
2	Number of teeth	60
3	Pressure Angle	20 Degree
4	Lead Angle	$3^0 26'$
5	Gear Face Width	14 mm
6	Pitch Diameter	90 mm
7	Outside Diameter	96 mm
8	Bore Diameter (H8)	12mm
	Material	Cast Iron
Worm		
1	Module	1.5
2	Single Thread	-
3	Right Hand	-
4	Length	55mm
5	Lead Angle	$3^0 26'$
6	Outside Diameter	28mm
7	Pitch Diameter	25
8	Material	Steel

Center distance of worm and Gear is 57.5 mm

Table B.2 : North American standards of stairs and steps [25]

North American standards of stairs and steps			
Degree(γ)	Γ: (In)	b: (In)	$A_{\Gamma,b} = \sqrt{\Gamma^2 + b^2}$(In)
30	6.5	11	12.78
32	6.75	10.75	12.69
33	7	10.5	12.61
35	7.25	10.25	12.55
36	7.5	10	12.5
38	7.75	9.75	12.45
40	8	9.5	12.41
41	8.25	9.25	12.39
43	8.5	9	12.37
45	8.75	8.75	12.374
46	9	8.5	12.38
48	9.25	8.25	12.4
50	9.5	8	12.42

Appendix C

C.1 Dynamic Analysis of aTracked Robot with Flat Belts during Climbing Stairs from Phase 1 through Phase 18

C.1.1 Dynamic of the Flat Belt Robot during *Phase* 1

With reference to Figure 3.5 the behavior of the tracked robot with a flat belt is similar to the one with grousers, but the direction of the front tangential force is the same as the belt's direction. The normal reaction force is perpendicular to the belt.

With reference to Figure 3.5, the equations of motions of the robot are given by:

$$\sum F_x = m(a_{CM})_x = F_M \cos(\theta + \beta) + F_A - F_C \cos(\theta + \beta) - N_C \sin(\theta + \beta) \qquad C.\ 1$$

$$\sum F_y = m(a_{CM})_y = F_M \sin(\theta + \beta) + N_A - F_C \sin(\theta + \beta) + N_C \cos(\theta + \beta) - mg \qquad C.\ 2$$

Taking the moment at point C yields:

$$\sum M_C = \left(I + mR_{r_1}{}^2\right)\Omega = N_A(\mu\Gamma - L\cos\beta - q\ ) + (L_C \cos\beta + q)mg \qquad C.\ 3$$

These equations are valid from $\beta = 0\ to \beta = \cos^{-1}\frac{\Gamma}{L}$

The acceleration of the center of mass with respect to the acceleration of the rear of the tracked robot is:

$$a_{CM} = a_A + \alpha \times r_{Cm/A} - \omega^2 r_{Cm/A} \qquad C.\ 4$$

$$(a_{CM})_x i + (a_{CM})_y j =$$

$$= a_{Ai} + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_A cos\,\beta - h\,sin\,\beta) & (L_A sin\,\beta + h\,cos\,\beta) & 0 \end{vmatrix} \qquad C.\ 5$$

$$(a_{CM})_x = (a_A)_x - [\Omega(L_A sin\,\beta + h\,cos\,\beta)] \qquad C.\ 6$$

$$(a_{CM})_y = [\Omega(L_A cos\,\beta - h\,sin\,\beta)] \qquad C.\ 7$$

C.1.2 Dynamic of the Tracked Robot during Phase 2

With reference to Figure C.2, the behavior of the tracked robot with a flat belt is similar to the one with grousers, but the direction of the front tangential force is the same as the belt's direction. The normal reaction force is perpendicular to the belt.

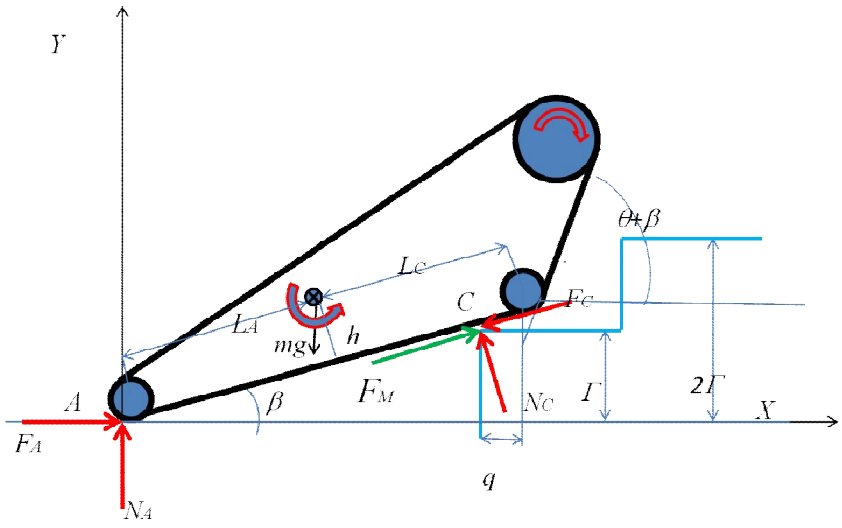


Figure C.1 : Robots posees, as it pass the first step

The equations of motions of the robot are:

$$\sum F_x = m(a_{cm})_x = F_M\,cos\,\beta + F_A\ - F_C\,cos\,\beta - N_C\,sin\,\beta \qquad C.\ 8$$

$$\sum F_y = m(a_{cm})_y = F_M\,sin\,\beta + N_A - F_C\,sin\,\beta\ + N_C\,cos\,\beta - mg \qquad C.\ 9$$

$$\sum M_C = [I + m(R_{r2})^2]\Omega = N_A\,(\mu\Gamma - L\,cos\,\beta - q) - mg(L_C cos\,\beta - q + h\,sin\,\beta) \qquad C.\ 10$$

The acceleration of the center of mass with respect to the acceleration of the rear of the tracked robot is:

$$(a_{CM})_x i + (a_{CM})_y j = a_A i + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_A \cos\beta - h\sin\beta) & (L_A \sin\beta + h\cos\beta) & 0 \end{vmatrix} \qquad C.\ 11$$

$$(a_{CM})_x = (a_A)_x - \Omega\,(L_A \sin\beta + h\cos\beta) \qquad C.\ 12$$

$$(a_{CM})_y = \Omega(L_A \cos\beta - h\sin\beta) \qquad C.\ 13$$

C.1.3 Dynamic Analysis of the Tracked Robot during *Phases*3 and 4

With reference to Figure C.3, the behavior of the tracked robot with a flat belt is similar to the one with grousers during *Phase* 3, but the direction of the front tangential force is the same as the belt's direction. The normal reaction force is perpendicular to the belt.

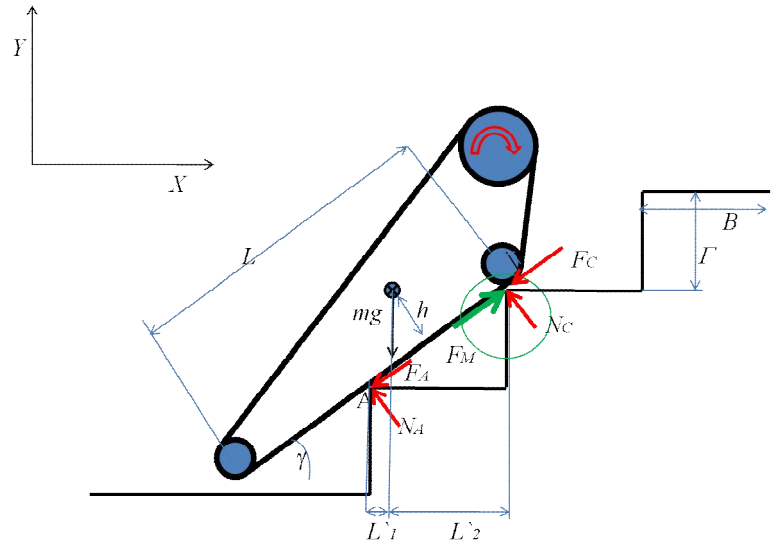


Figure C.2 : Robot poses, while its bottom part is contacting two step noses.

The equations of motions of the robot are given by:

$$\sum F_x = m(a_{CM})_x = F_M \cos\gamma + F_A \cos\gamma - N_A \sin\gamma - N_c \sin\gamma - F_C \cos\gamma \qquad C.\ 14$$

$$\sum F_y = m(a_{CM})_y = F_M \sin\gamma + F_A \sin\gamma + N_A \cos\gamma + N_C \cos\gamma - F_C \sin\gamma - mg \qquad C.\ 15$$

$$\sum M_A = -mgL_1^{`} + F_M[(\Gamma\cos\gamma) - (B)\sin\gamma] + N_C\{[\cos\gamma\,(\Gamma\mu + (B))] + [\mu\sin\gamma\,((B) - \Gamma)]\} \qquad C.\ 16$$

With reference to Figure C.4, the behavior of the tracked robot with a flat belt is similar to the one with grousers during *Phase* 4, but the direction of the front tangential force is the same as the belt's direction. The normal reaction force is perpendicular to the belt.

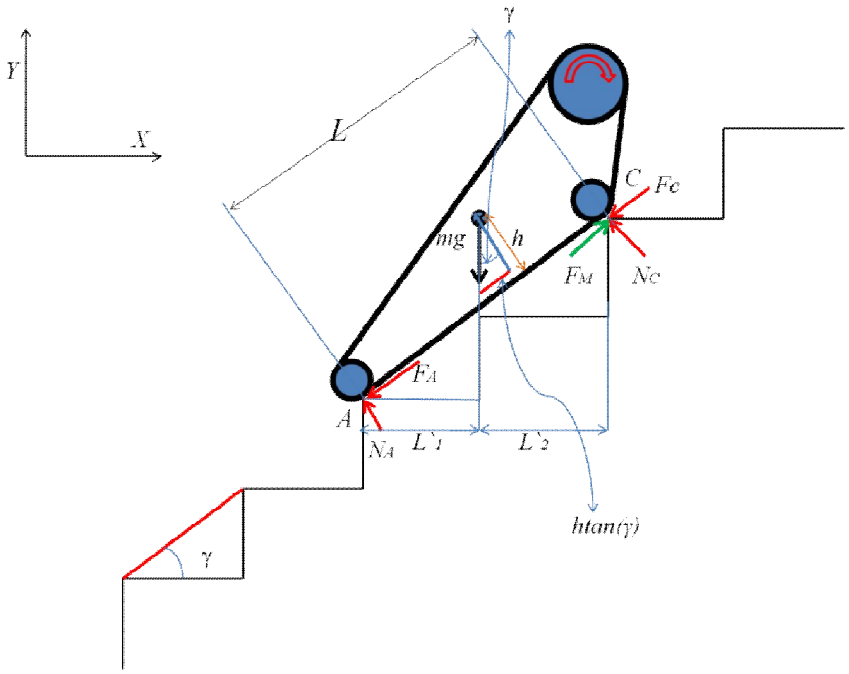


Figure C.3 : Robot poses, while its bottom part is contacted with three steps noses.

The equations of motions of the robot are given by:

$$\sum F_x = m(a_{CM})_x = F_M \cos\gamma - F_A \cos\gamma - N_A \sin\gamma - N_c \sin\gamma - F_C \cos\gamma \qquad C.\ 17$$

$$\sum F_y = m(a_{CM})_y = F_M \sin\gamma - F_A \sin\gamma + N_A \cos\gamma + N_c \cos\gamma - F_C \sin\gamma \qquad C.\ 18$$

$$\sum M_A = -mgL_1^{`} + F_M[(2\Gamma\cos\gamma) - (2B)\sin\gamma] + N_C\{[\cos\gamma\,(2\Gamma\mu + (2B))] + [\mu\sin\gamma\,((2B) - 2\Gamma)]\} \qquad C.\ 19$$

C.1.4 Dynamic Analysis of the Tracked Robot during *Phase* 5

With reference to FigureC.3 the motion of Tracked Robot in *Phase* 5 is similar to that of *Phase* 3.

C.1.5 Dynamic Analysis of the Tracked Robot in *Phase* 6

With reference to Figure C.5, the behavior of the tracked robot with a flat belt is similar to the one with grousers, but the direction of the front tangential force is the same as the belt's direction. The normal reaction force is perpendicular to the belt.

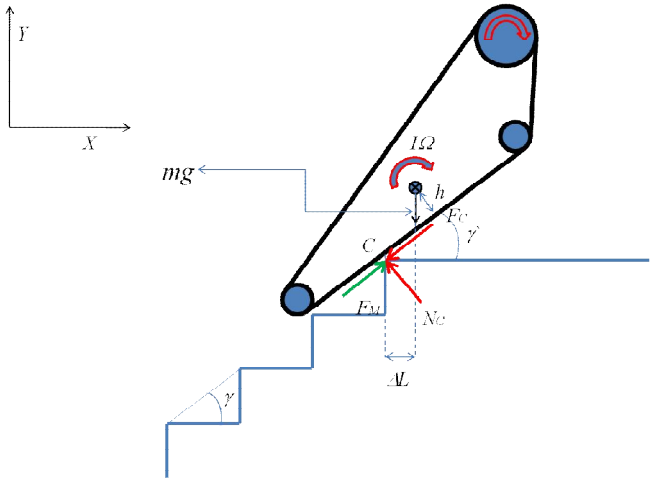


Figure C.4 : Robot poses, while its center of mass crosses pivot point.

The equations of motions of the robot are:

$$F_X = F_M \cos\beta - F_C \cos\beta - N_C \sin\beta \qquad C.\ 20$$

$$F_y = F_M \sin\beta - F_C \sin\beta + N_C \cos\beta - mg \qquad C.\ 21$$

$$\sum M_{cm} = I\Omega = -F_M \cos\beta\, \Delta L + N_C \sin\beta \Delta L + F_M \sin\beta \left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) + N_C \cos\beta \left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) \qquad C.\ 22$$

The acceleration of the center of mass with respect to the acceleration of the rear of the tracked robot is given by:

$$(a_{AM})_x i + (a_{CM})_y j = (a_c)_x i + (a_c)_y + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ \Delta L & \left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) & 0 \end{vmatrix} \qquad C.\ 23$$

$$(a_{CM})_x = (a_c)_x - \Omega\left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) \qquad C.\ 24$$

$$(a_{CM})_y = (a_c)_y + \Omega\Delta L \qquad C.\ 25$$

C.1.6 Dynamic Analysis of the Tracked Robot during *Phase*7and 8

With reference to Figure C.6 the behavior of the tracked robot with a flat belt is similar to the one with grousers for *Phase*7and 8. The direction and magnitudes of the external forces are the same as in the grouser case.

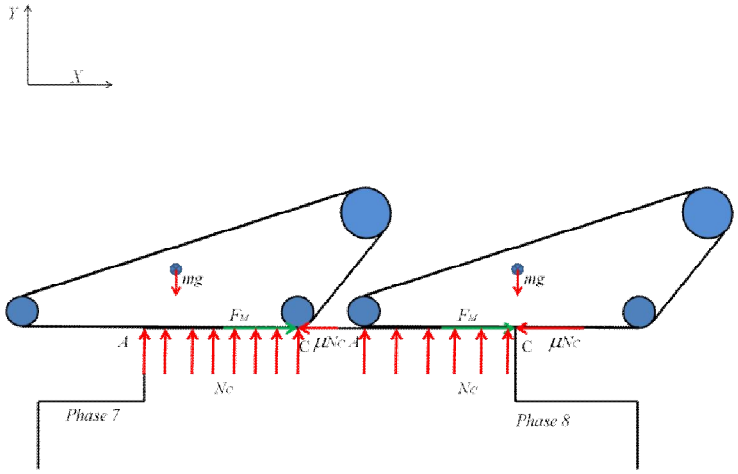


Figure C.5 : Robot poses, while its center of mass is on top of the level surface (Phases 7and 8)

$$F_X = m\frac{d^2x}{dt^2} = F_M - mg \times \mu \qquad C.\ 26$$

C.1.7 Dynamic Analysis of the Tracked Robot during *Phase* 9 (Descending)

As the center of mass cross the pivot point, the increasing distance between the center of gravity and the edge of the step generates an increasing torque. While the vehicle moves forward with velocity *v*, the torque caused by gravity increases the robot's acceleration, *a*. It is very important to control this acceleration; otherwise the possibility of overturning is very high.

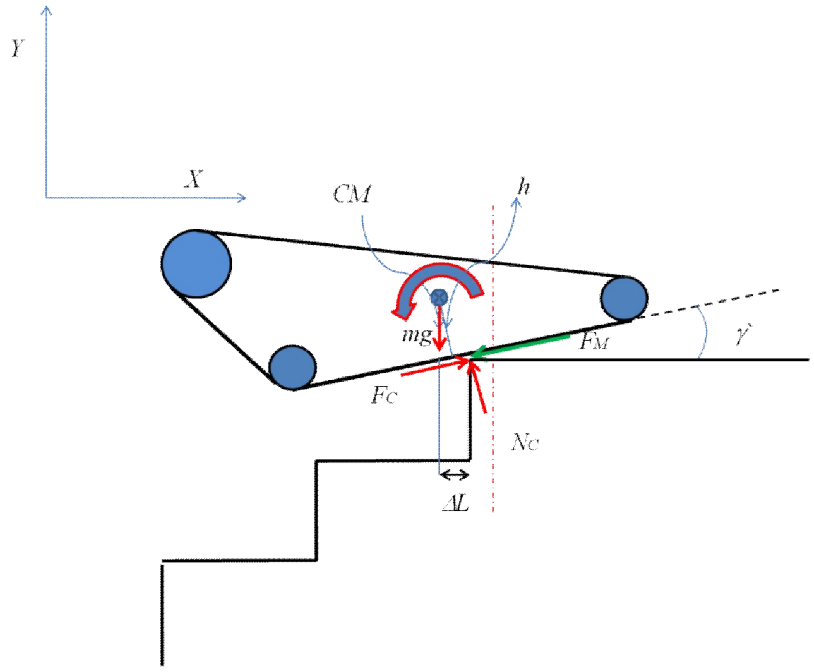


Figure C.6 : Robot poses, during the descending in (*Phase*9).

The equations of motions of the robot are:

$$\sum F_x = m(a_{cm})_x = -F_M \cos\beta + F_C \cos\beta - N_C \sin\beta \qquad C.\ 27$$

$$\sum F_y = m(a_{cm})_y = -F_M \sin\beta + F_C \sin\beta + N_C \cos\beta - mg \qquad C.\ 28$$

$$\sum M_{cm} = I\Omega = -F_M\left[\left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right)\cos\beta + \sin\beta\Delta L\right] +$$

$$N_C\left[\cos\beta\Delta L - \sin\beta\left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right)\right] + F_C\left[\cos\beta\left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) + \sin\beta\,\Delta L\right]$$

C. 29

The acceleration of the center of mass with respect to the acceleration of the rear of the tracked robot is given by:

$$(a_{CM})_x i + (a_{CM})_y j = (a_c)_x i + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ \Delta L & \left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) & 0 \end{vmatrix} \qquad C.\ 30$$

$$(a_{CM})_x = (a_c)_x - \Omega\left({}^{h}/_{\cos\beta} + \Delta L \tan\beta\right) \qquad C.\ 31$$

$$(a_{CM})_y = (a_c)_y + \Omega\Delta L \qquad C.\ 32$$

C.1.8 Dynamic Analysis of the Tracked Robot during *Phase*10

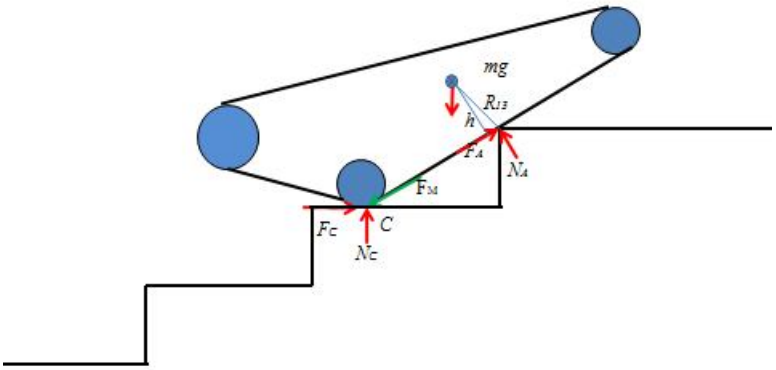


Figure C.7 : Robot poses, while it tips over the first step before reaching the second nose (during *Phase*10).

With reference to Figure C.8, the equations of motions of the robot are given by:

$$R_{r3} = \sqrt{h^2 + \left({\Gamma}/{\tan\beta} - L_C\right)^2} \qquad C.\ 33$$

$$\sum F_X = m\ddot{x} = F_C + F_A\cos\beta - N_A\sin\beta - F_M\cos\beta \qquad C.\ 34$$

$$\sum F_y = m\ddot{y} = N_c + N_A\cos\beta - F_A\sin\beta - F_M\sin\beta \qquad C.\ 35$$

$$\sum M_C = (I + m(R_{r3})^2)\,\Omega = N_A\Gamma\left[{1}/{\tan\beta}(\mu\sin\beta - \cos\beta) - (\sin\beta + \mu\cos\beta)\right] +$$

$$mg\left[{\Gamma}/{\tan\beta} - \cos\beta\,(\Gamma\sin\beta - L_C) - h\sin\beta\right] \qquad C.\ 36$$

The acceleration of the center of mass with respect to the acceleration of the front of the tracked robot is given by:

$$(a_{CM})_x i + (a_{CM})_y j =$$

$$a_A i + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_C \cos\beta - h \sin\beta) & (L_C \sin\beta + h \cos\beta) & 0 \end{vmatrix} - \varpi^2\big((L_C \cos\beta - h \sin\beta)i + (L_C \sin\beta + h \cos\beta)j\big) \quad C.\ 37$$

$$(a_{CM})_x = (a_A)_x - \ddot{\beta}(L_C \sin\beta + h \cos\beta) \quad C.\ 38$$

$$(a_{CM})_y = \ddot{\beta}\,(L_C \cos\beta - h \sin\beta) \quad C.\ 39$$

C.1.9 Dynamic Analysis of the Tracked Robot during *Phase* 11

The motions of the tracked robot without grousers are similar to those of the robot with grousers during the *Phase* 11 descent.

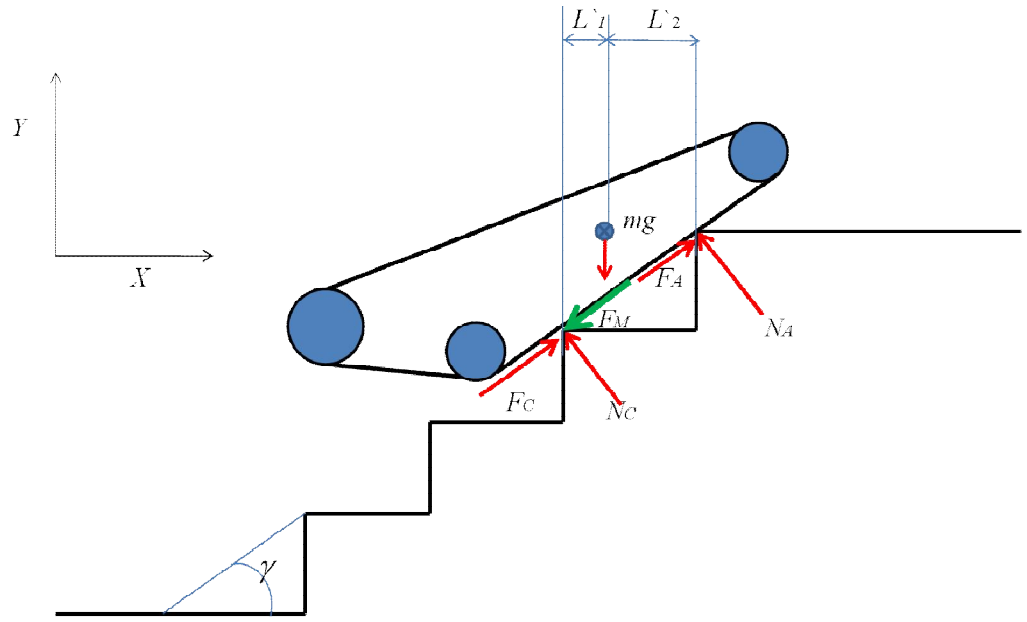


Figure C.8 : Robot poses, during the *Phase* 11 descent. (The Tracked Robot has two contacts with stairs noses).

With reference to Figure C.9, the equations of motions of the robot are given by:

$$\sum F_x = m(a_{cm})_x = -F_M \cos\gamma + \cos\gamma\,(-F_A + F_C) - \sin\gamma\,(N_A + N_C) \qquad C.\ 40$$

$$\sum F_y = -F_M \sin\gamma + \sin\gamma(-F_A + F_C) + \cos\gamma\,(N_A + N_C) - mg \qquad C.\ 41$$

Applying the equations of motion yields:

$$\sum M_A = F_M[(\Gamma\cos(\gamma)) - (B)\sin(-\gamma)] + mgL_1^{`} - N_C[\Gamma(\sin(\gamma) + \mu\cos(\gamma)) + (B)(\cos(-\gamma) + \mu\sin(-\gamma))] \qquad C.\ 42$$

C.1.10 Dynamic Analysis of the Tracked Robot during *Phase* 12

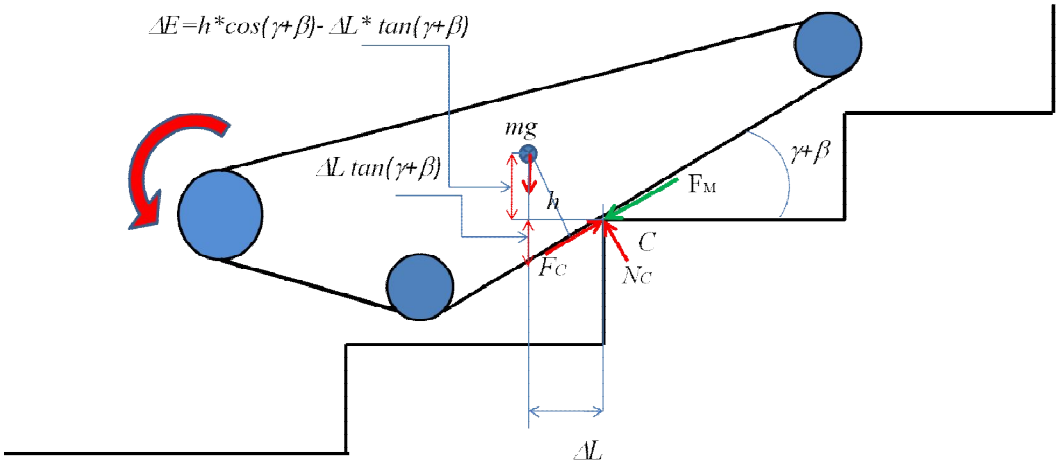


Figure C.9 : Robot poses, while it tips over the step and before reaching the following nose (during *Phase*12).

With reference to Figure C.10, the equations of motions of the robot are given by:

$$\Delta E = {}^{h}/_{\cos(\gamma + \beta)} - \Delta L\tan(\beta) \qquad C.\ 43$$

$$\sum F_x = m(a_{cm})_y = -F_M\cos(\beta) + F_C\cos(\beta) - N_C\sin(\beta) \qquad C.\ 44$$

$$\sum F_y = m(a_{cm})_y = -F_M\sin(\beta) + F_C\sin(\beta) + -mg \qquad C.\ 45$$

$$\sum M_{cm} = I\Omega = -F_M[\sin(\beta)\,\Delta L + \cos(\beta)\,\Delta E] + N_C\{[-\sin(\beta)\,\Delta L + \mu\cos(\gamma + \beta)]\Delta L + \Delta E[\mu\sin(\beta)\,\Delta L + \cos(\beta)]\} \qquad C.\ 46$$

The acceleration of the center of mass with respect to the pivot point is given by:

$$(a_{CM})_x i + (a_{CM})_y j = (a_c)_x i + (a_c)_y j + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ -\Delta L & \Delta E & 0 \end{vmatrix} \quad C.\ 47$$

$$(a_{CM})_x = (a_c)_x - \Omega \Delta E \quad C.\ 48$$

$$(a_{CM})_y = (a_c)_y + \Omega \Delta L \quad C.\ 49$$

C.1.11 Dynamic Analysis of the Tracked Robot during *Phase*13

The motion of the tracked robot during *Phase* 13 is similar to that of the *Phase*10 motion.

C.1.12 Dynamic Analysis of the Tracked Robot during *Phase* 14

The motions of the tracked robot with grousers and without grousers are similar to the *Phase* 14 descent.

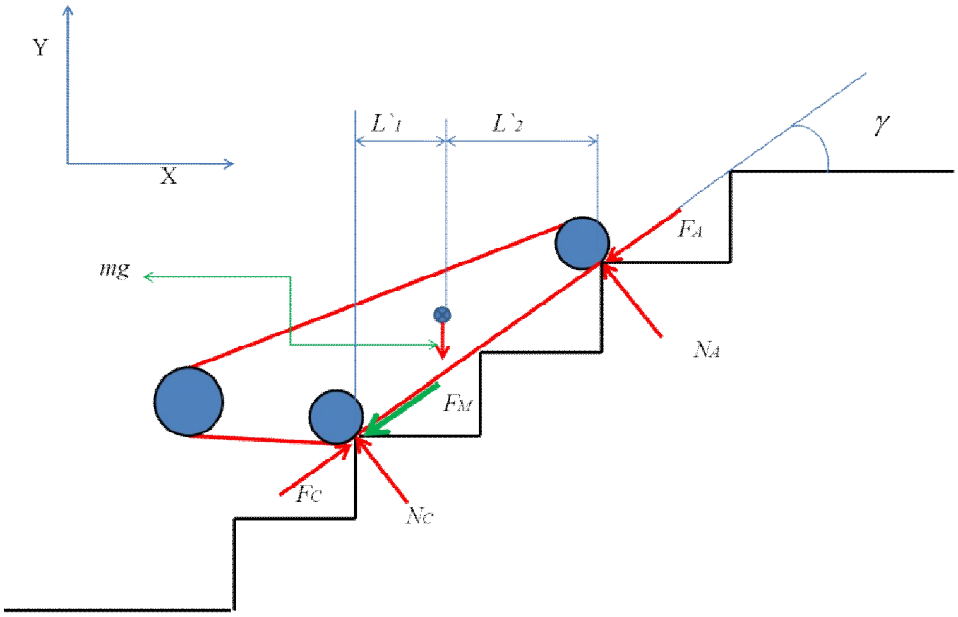


Figure C.10 : Robot poses, during the descending in (Tracked Robot has two contact with stairs noses).

With reference to Figure C.11, the equations of motions of the robot are given by:

$$\sum F_x = m(a_{cm})_x = -F_M \cos\gamma + \cos\gamma\,(-F_A + F_C) - \sin\gamma\,(N_A + N_C) \quad C.\ 50$$

$$\sum F_y = -F_M \sin\gamma + \sin\gamma(-F_A + F_C) + \cos\gamma\,(N_A + N_C) - mg \qquad C.\ 51$$

$$\sum M_A = F_M[(2\Gamma\cos(\gamma)) - (2B)\sin(\gamma)] + mgL_1' - N_C[2\Gamma(\sin(\gamma) + \mu\cos(\gamma)) + (2B)(\cos(\gamma) + \mu\sin(\gamma))] \qquad C.\ 52$$

C.1.13 Dynamic Analysis of the Tracked Robot during *Phase* 15

The motion of theTracked Robot during *Phase* 15 is similar to the *Phase*11.

C.1.14 Dynamic Analysis of the Tracked Robot during *Phase* 16

The motion of theTracked Robot during *Phase* 16 is similar to the *Phase*12.

C.1.15 Dynamic Analysis of the Tracked Robot during *Phase* 17

The motions of the tracked robot with grousers and without grousers are similar to each other during the *Phase* 17 descent. The directions of the external forces are the same in both cases.

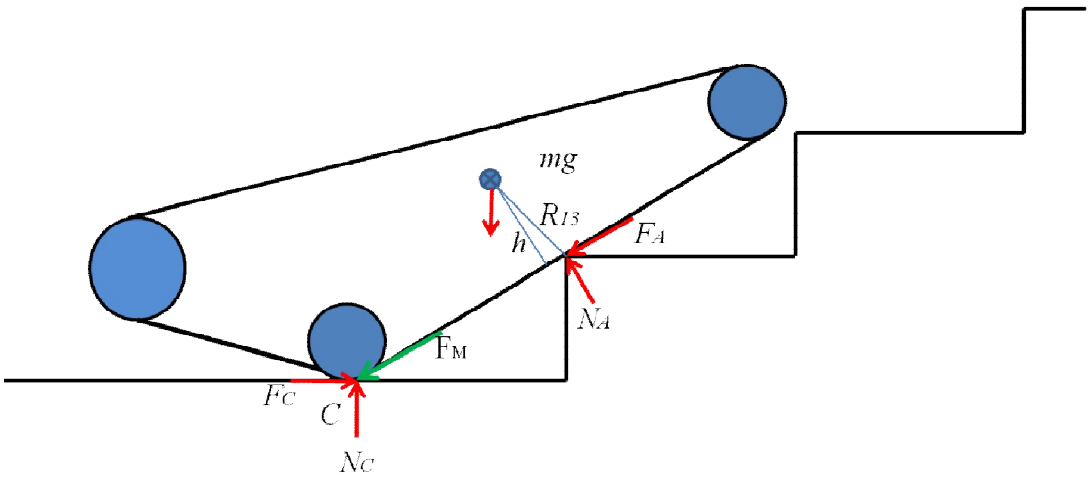


Figure C.11 : Robot poses, as it heat the level surface until the rear of Robot reaches the last steps nose.

With reference to Figure C.12, the equations of motions of the Robot are given by:

$$R_{r3} = \sqrt{h^2 + \left(\frac{\Gamma}{\cos\beta} - L_C\right)^2} \qquad C.\ 53$$

$$\sum F_X = m\ddot{x} = -F_M\cos\beta + F_C - F_A\cos\beta - N_A\sin\beta \qquad C.\ 54$$

$$\sum F_y = m\ddot{y} = -F_M\sin\beta + N_c - F_A\sin\beta + N_A\cos\beta \qquad C.\ 55$$

$$\sum M_C = (I + m(R_{r3})^2)\,\Omega = N_A \Gamma \left[{}^{1}/_{\tan\beta}(\mu \sin\beta - \cos\beta) - (\sin\beta + \mu\cos\beta) \right] +$$

$$mg \left[{}^{\Gamma}/_{\tan\beta} - \cos\beta\,(\Gamma \sin\beta - L_C) - h \sin\beta \right] \qquad C.\ 56$$

The acceleration of the center of mass with respect to the acceleration of the front of the tracked robot is given by:

$$(a_{CM})_x i + (a_{CM})_y j = a_A i + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_c \cos\beta - h \sin\beta) & (L_c \sin\beta + h \cos\beta) & 0 \end{vmatrix} \qquad C.\ 57$$

$$(a_{CM})_x = (a_A)_x - \ddot{\beta}(L_c \sin\beta + h \cos\beta) \qquad C.\ 58$$

$$(a_{CM})_y = \ddot{\beta}\,(L_c \cos\beta - h \sin\beta) \qquad C.\ 59$$

C.1.16 Dynamic Analysis of the Tracked Robot during Phase 18

The motions of the tracked robot with grousers and without grousers are similar during descending. Also the directions of external forces are same.

The motion of the tracked robot with grousers and without grousers is similar during the Phase 18 descent. The directions of the external forces are the same in both cases.

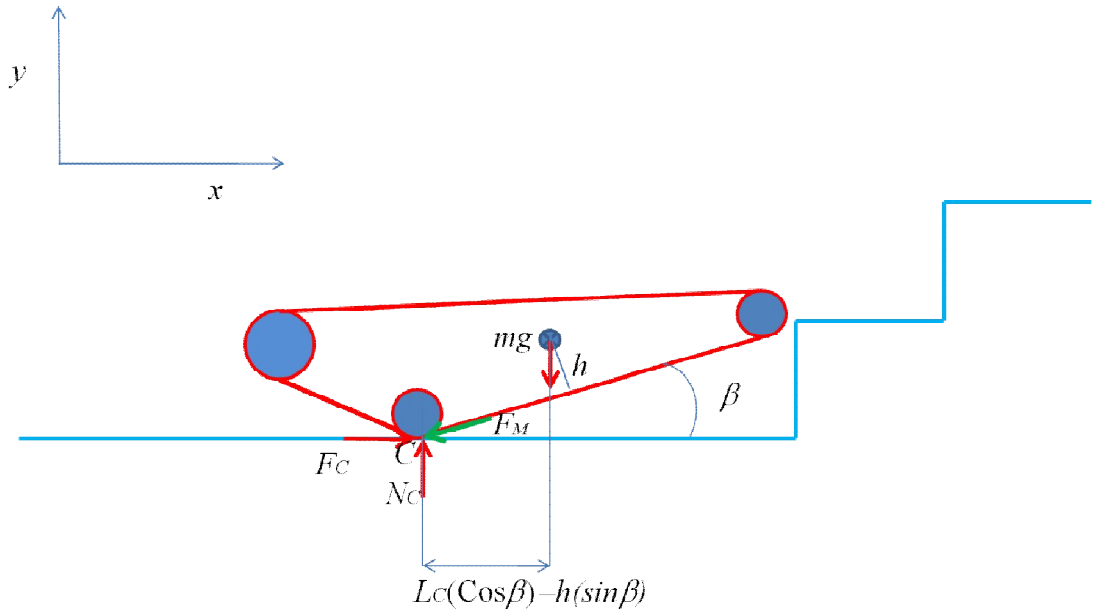


Figure C.12 : Robot poses, in *Phase*18

With reference to Figure C.13, the equations of motions of the robot are given by:

$$\sum F_x = m\frac{d^2x}{dt^2} = m(a_{cm})_x = -F_M\cos(\beta) + F_C \qquad C.\ 60$$

$$\sum F_y = m\frac{d^2y}{dt^2} = m(a_{cm})_y = -F_M\sin(\beta) + N_C - mg \qquad C.\ 61$$

$$\sum M_{Cm} = I\Omega = F_M\{\cos(\beta)\,[(L_C\cos\beta - h\sin\beta)] - \sin\beta\,(L_C\sin\beta + h\cos\beta)\} + N_C\{(L_C\cos\beta - h\sin\beta) - \mu(L_C\sin\beta + h\cos\beta)\} \qquad C.\ 62$$

The acceleration of the center of mass with respect to the acceleration of the front of the tracked robot is given by:

$$(a_{CM})_x i + (a_{CM})_y j = (a_c)_x i + (a_c)_y + \begin{vmatrix} i & j & k \\ 0 & 0 & \Omega \\ (L_C\cos\beta - h\sin\beta) & (L_C\sin\beta - h\cos\beta) & 0 \end{vmatrix} \qquad C.\ 63$$

$$(a_{CM})_x = (a_c)_x - \Omega[(L_C\sin\beta - h\cos\beta)] \qquad C.\ 64$$

$$(a_{CM})_y = [\Omega(L_A\cos\beta - h\sin\beta)] \qquad C.\ 65$$

Appendix D

D.1 Simulation of the Tracked Robot with Flat Belts While Climbing Stairs during Phases 1 through 18

The plots presented below show the results of the simulations of the robot equipped with a flat track while it navigates an ascending and descending staircase. These figures mirror those presented in the main body of the thesis.

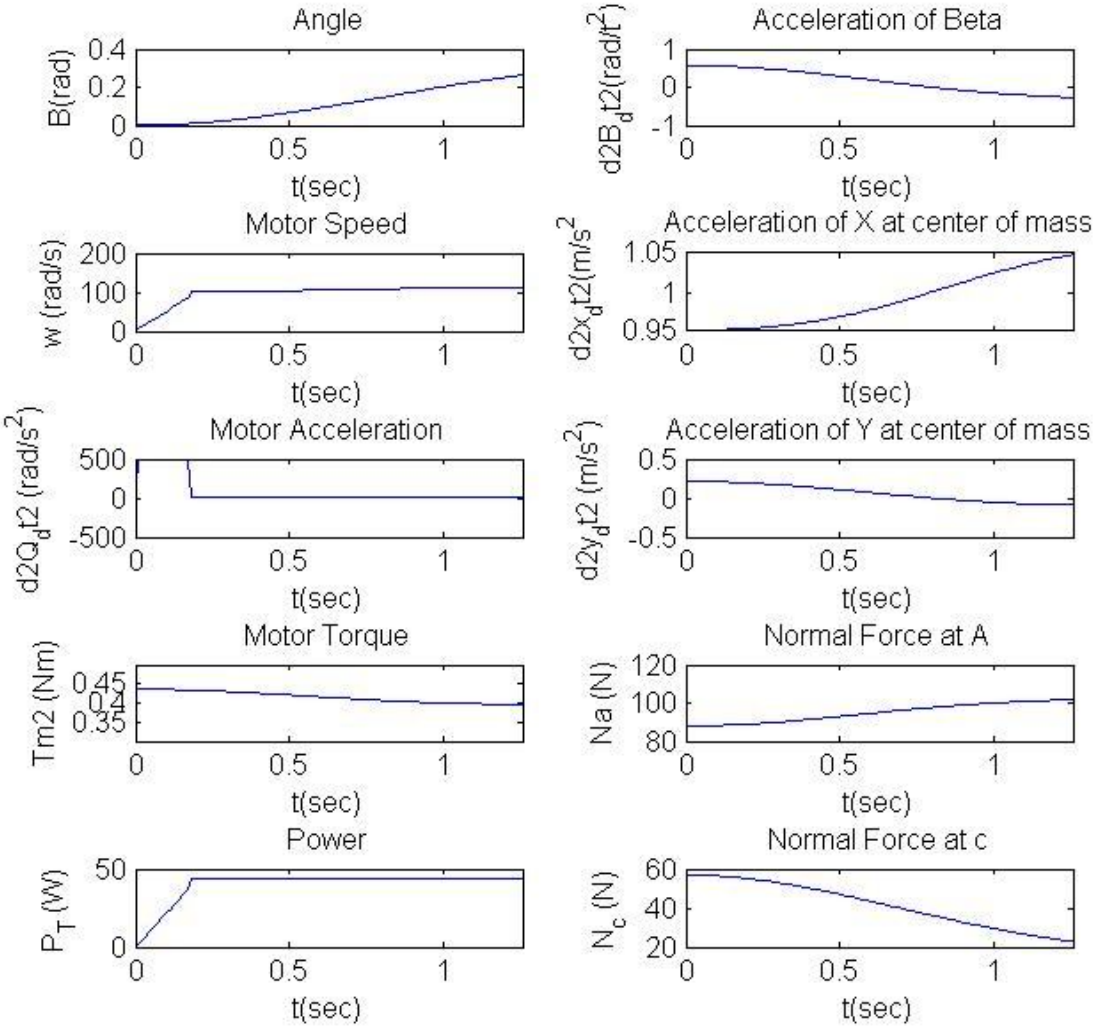


Figure D.1 : Simulation of the Tracked Robot during *Phase* 1

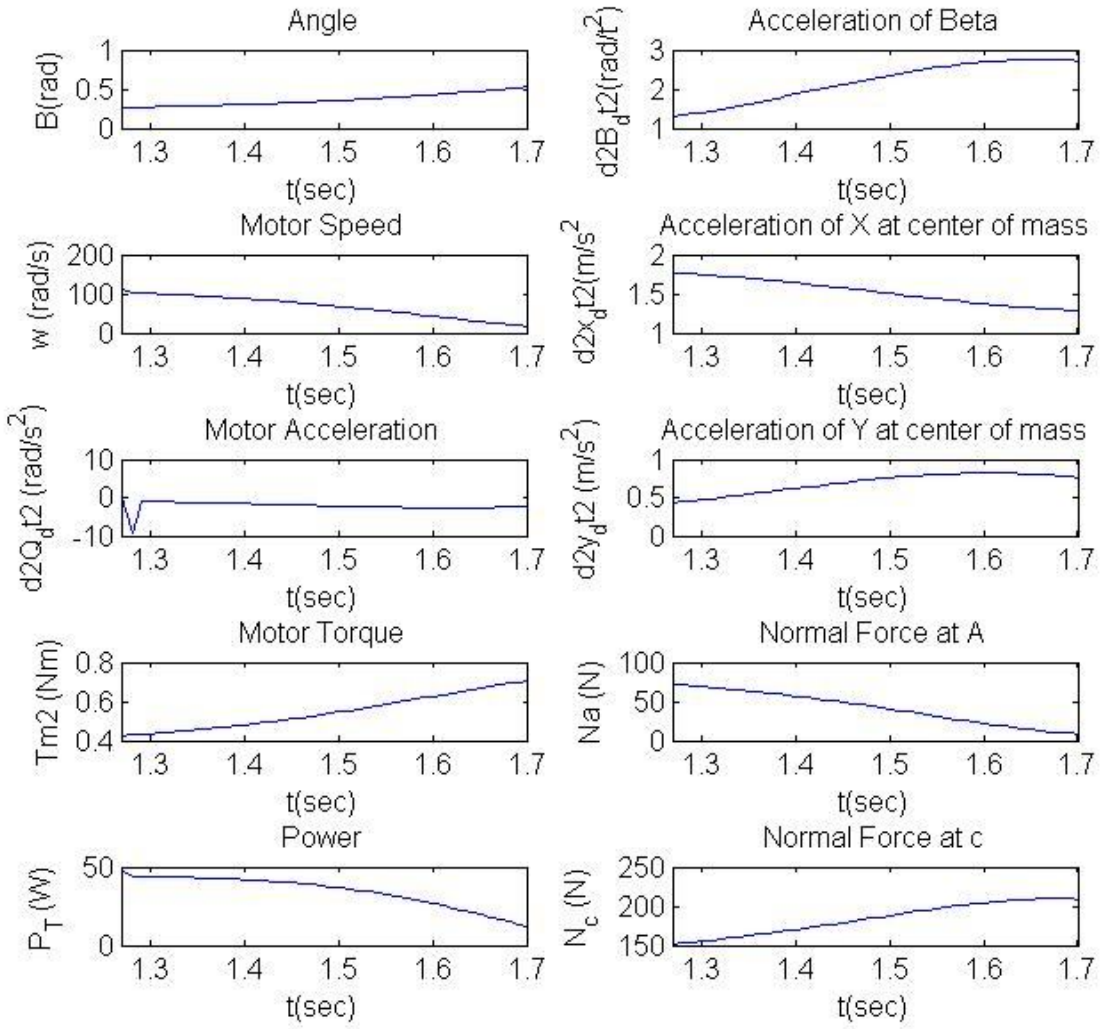


Figure D.2 : Simulation of the Tracked Robot during *Phase* 2

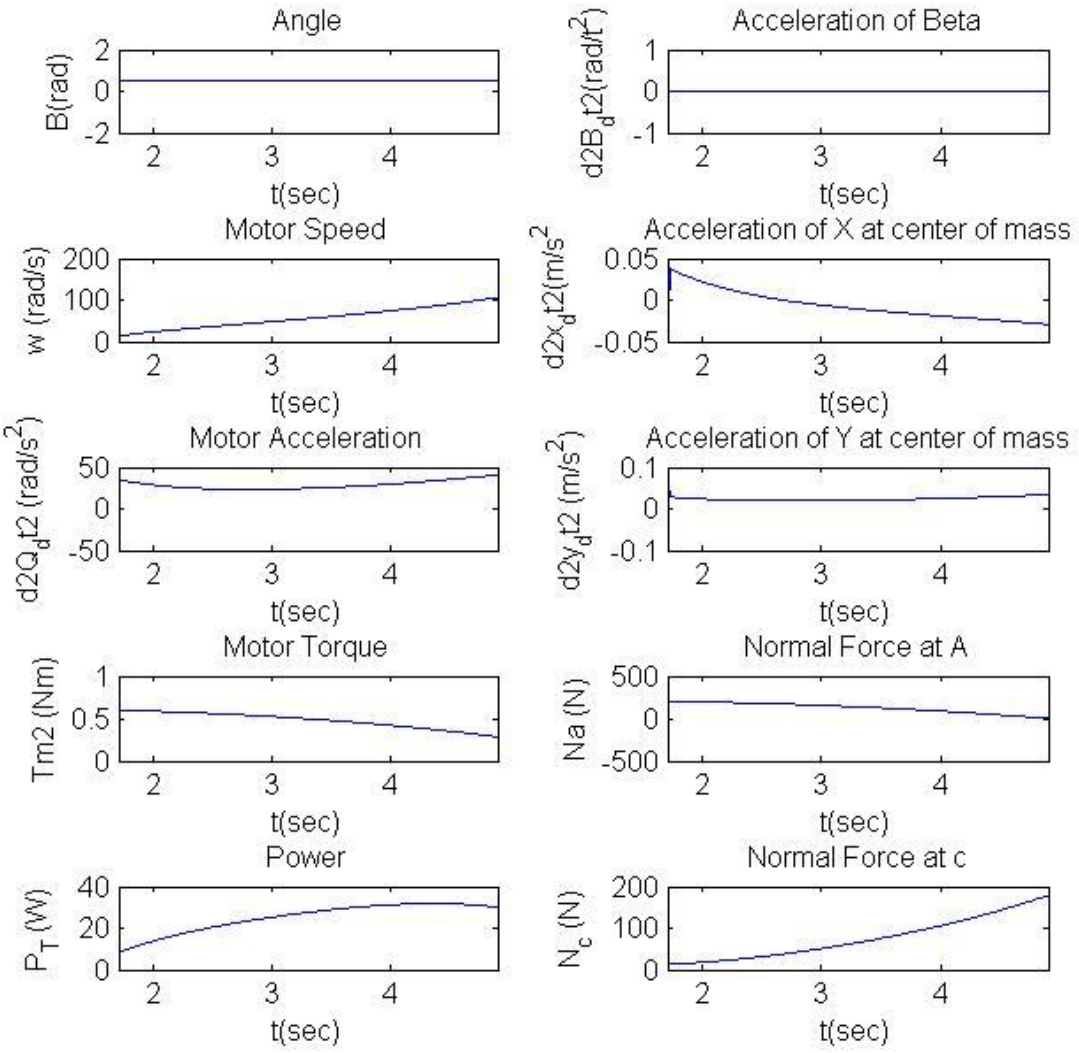


Figure D.3 : Simulation of the Tracked Robot during *Phase* 3

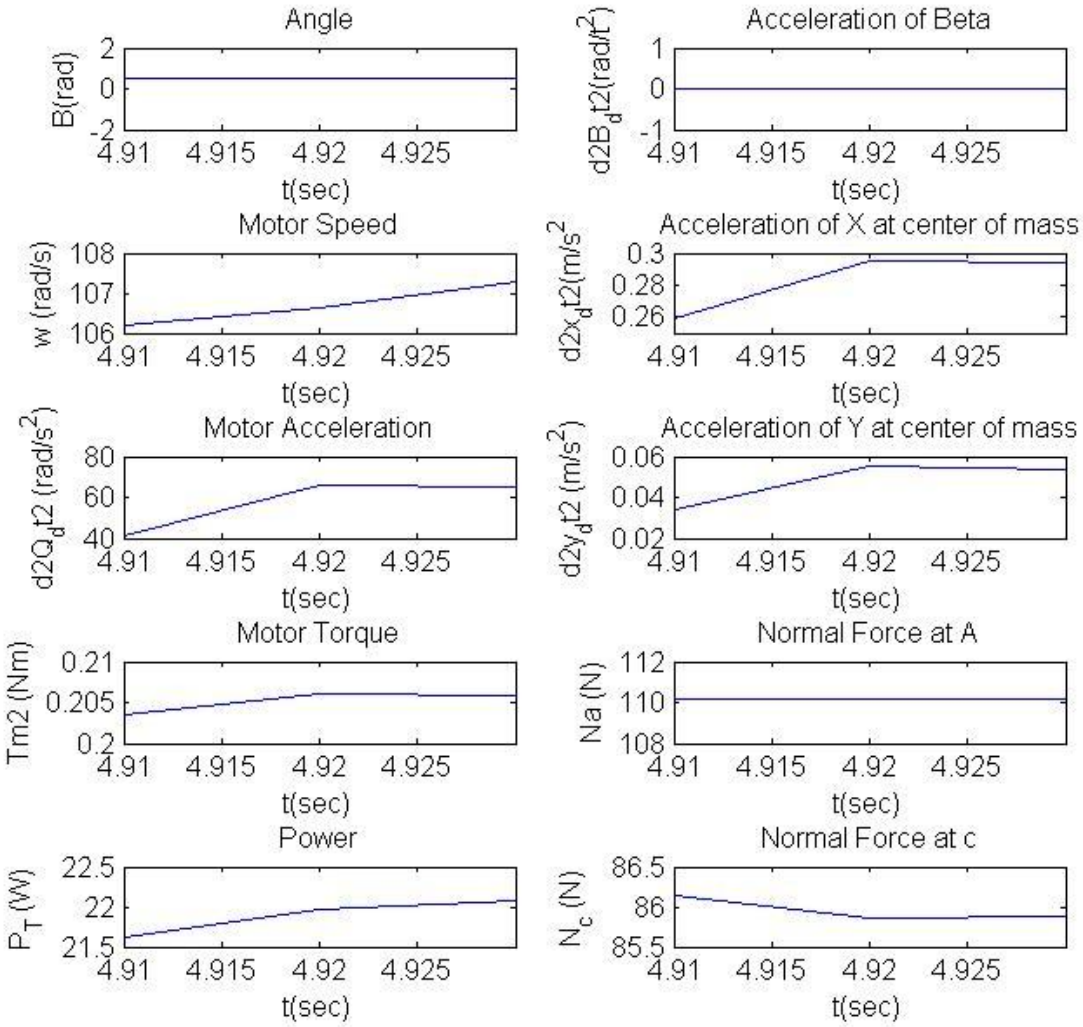


Figure D.4 : Simulation of the Tracked Robot during *Phase* 4

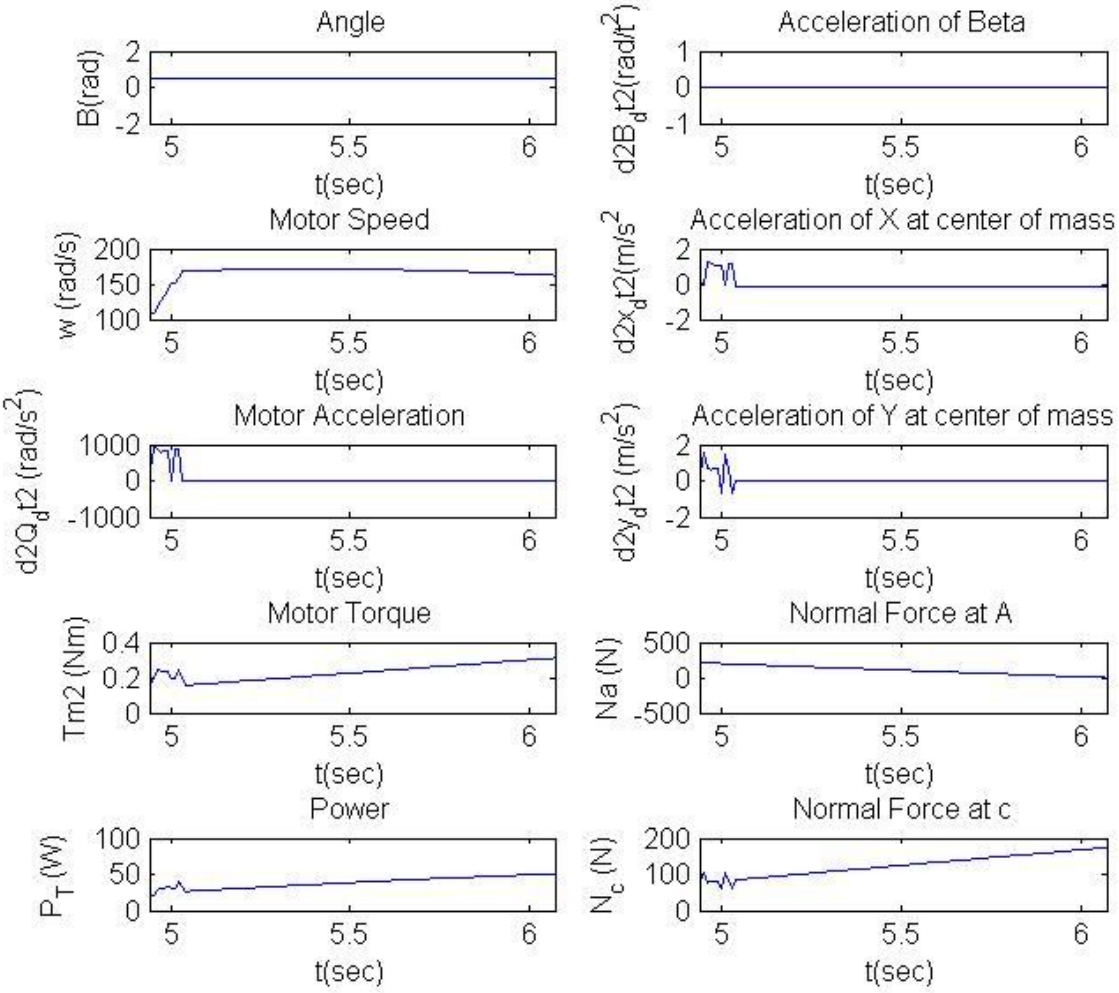


Figure D.5 : Simulation of the Tracked Robot during *Phase* 5

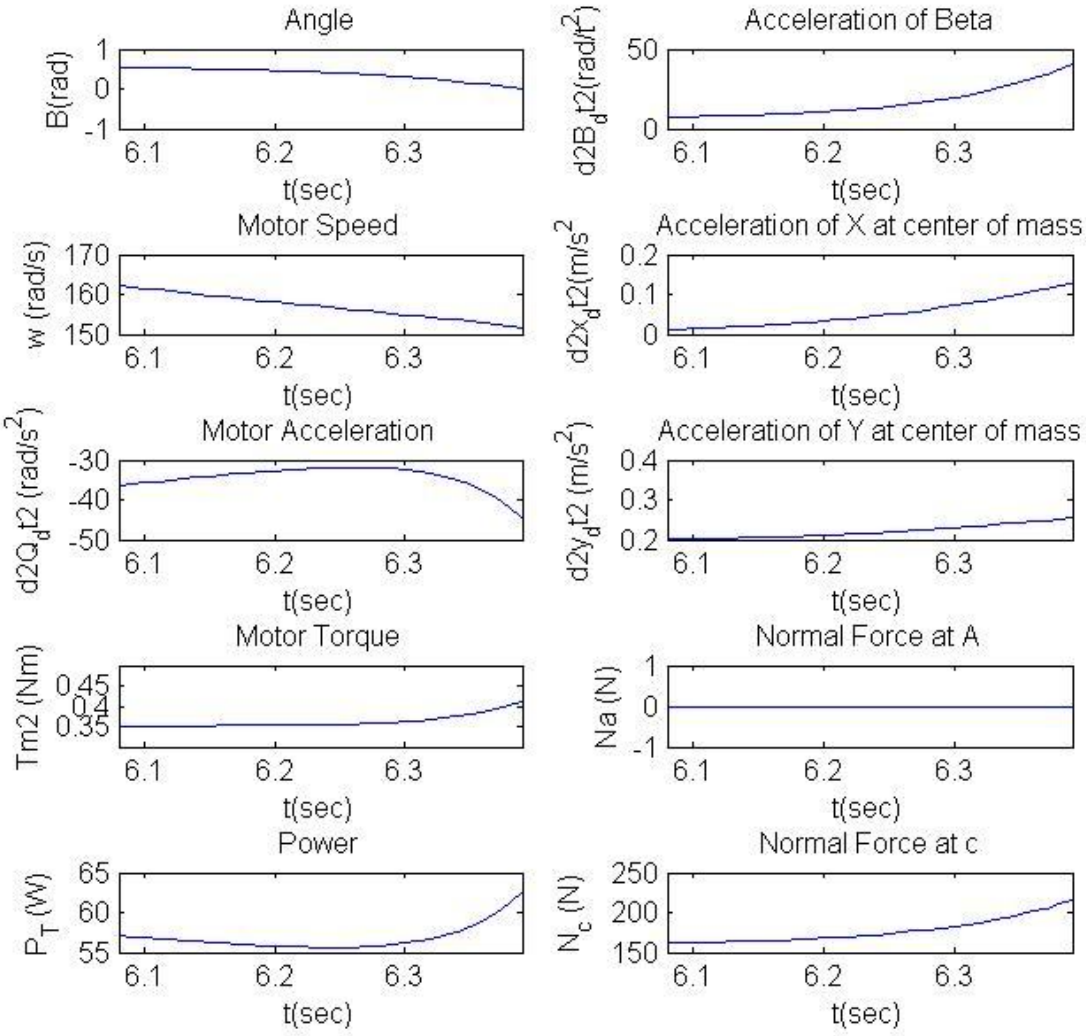


Figure D.6 Simulation of the Tracked Robot during *Phase* 6

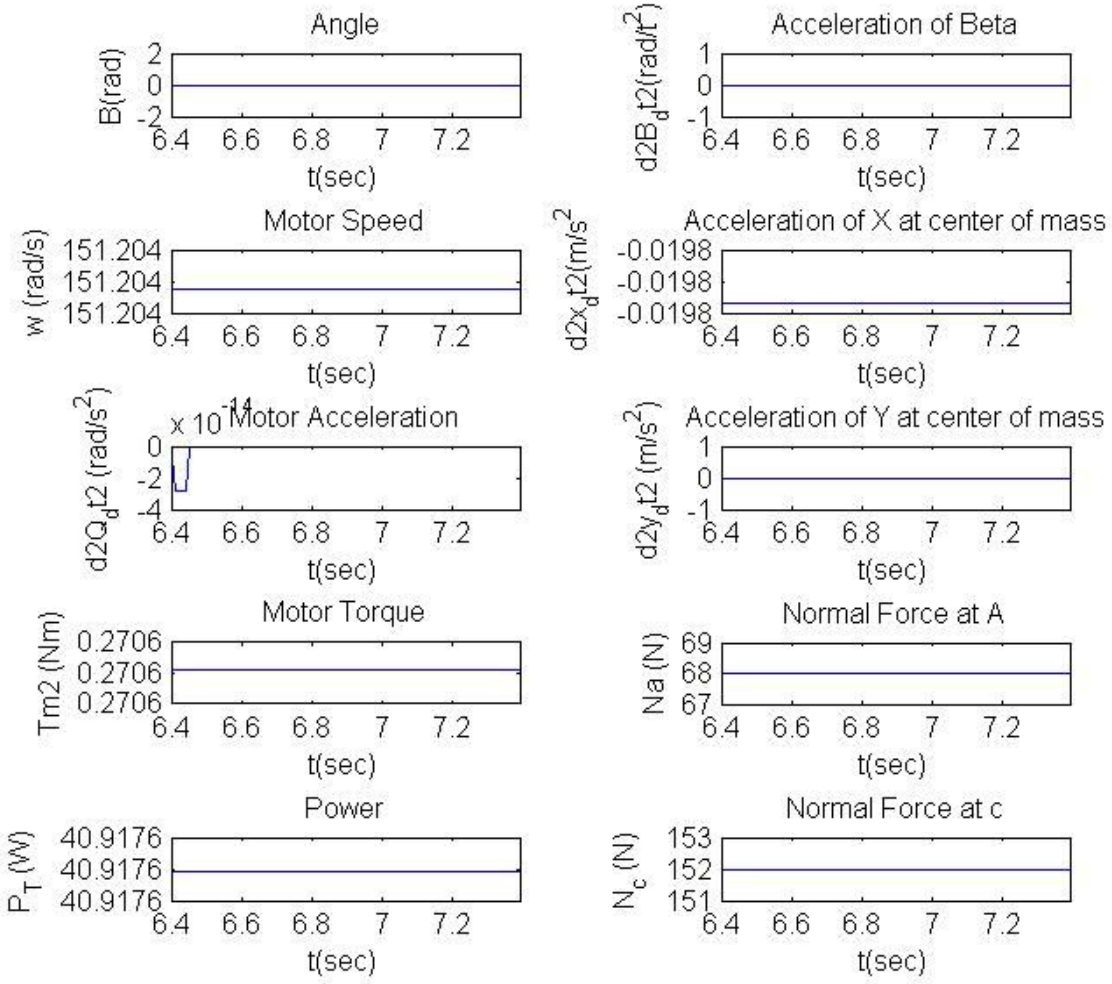


Figure D.7 Simulation of the Tracked Robot during *Phase* 7and 8

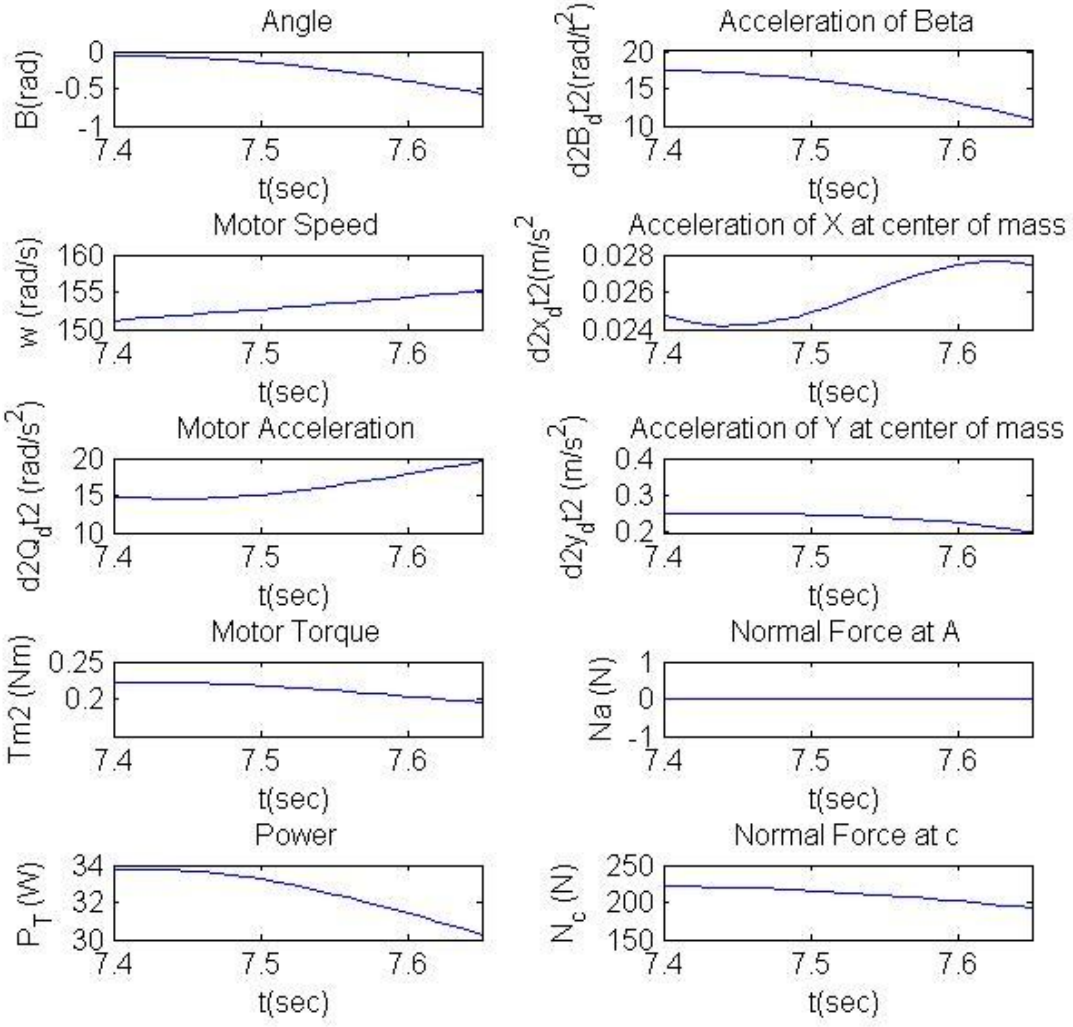


Figure D.8 Simulation of the Tracked Robot during *Phase* 9

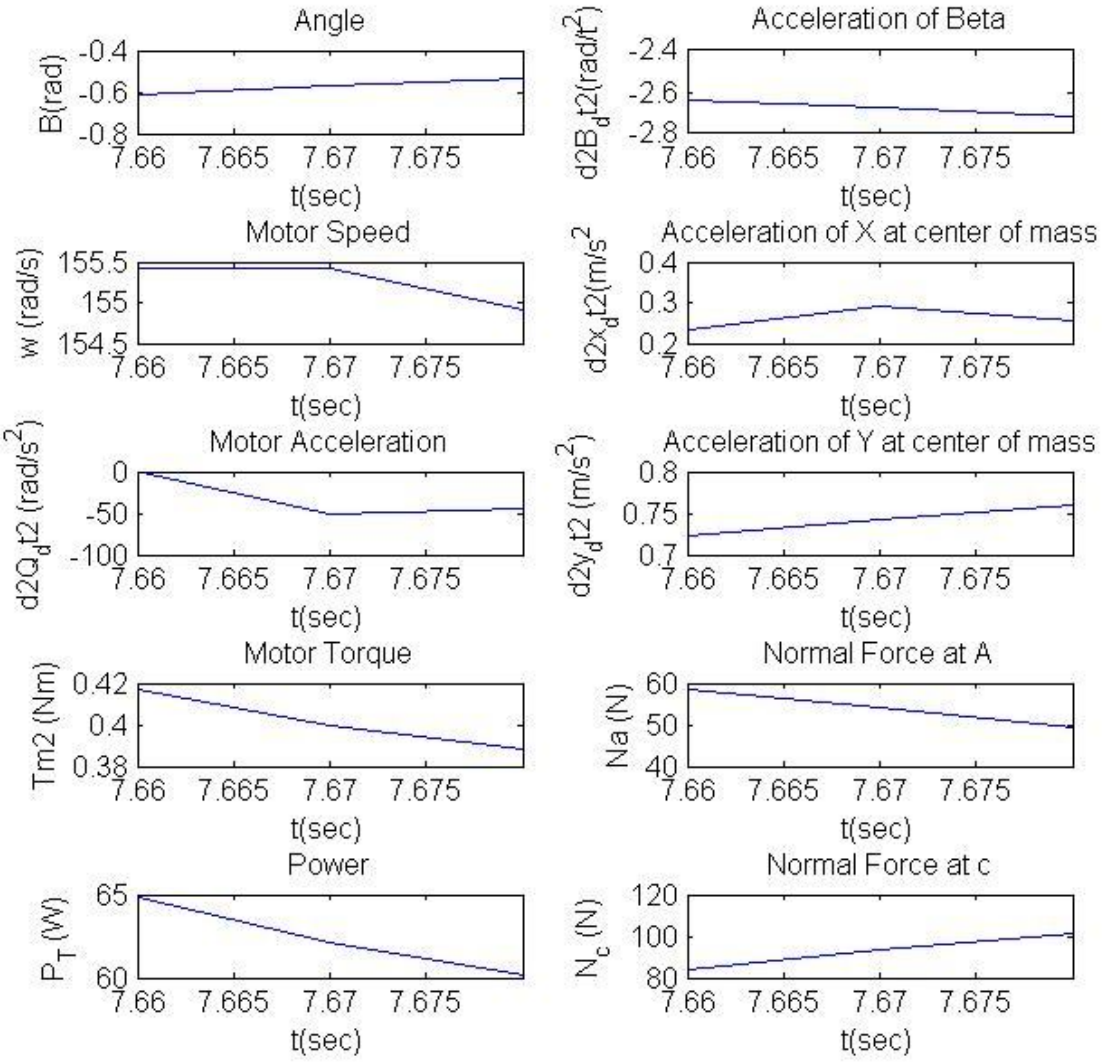


Figure D.9 Simulation of the Tracked Robot during *Phase* 10

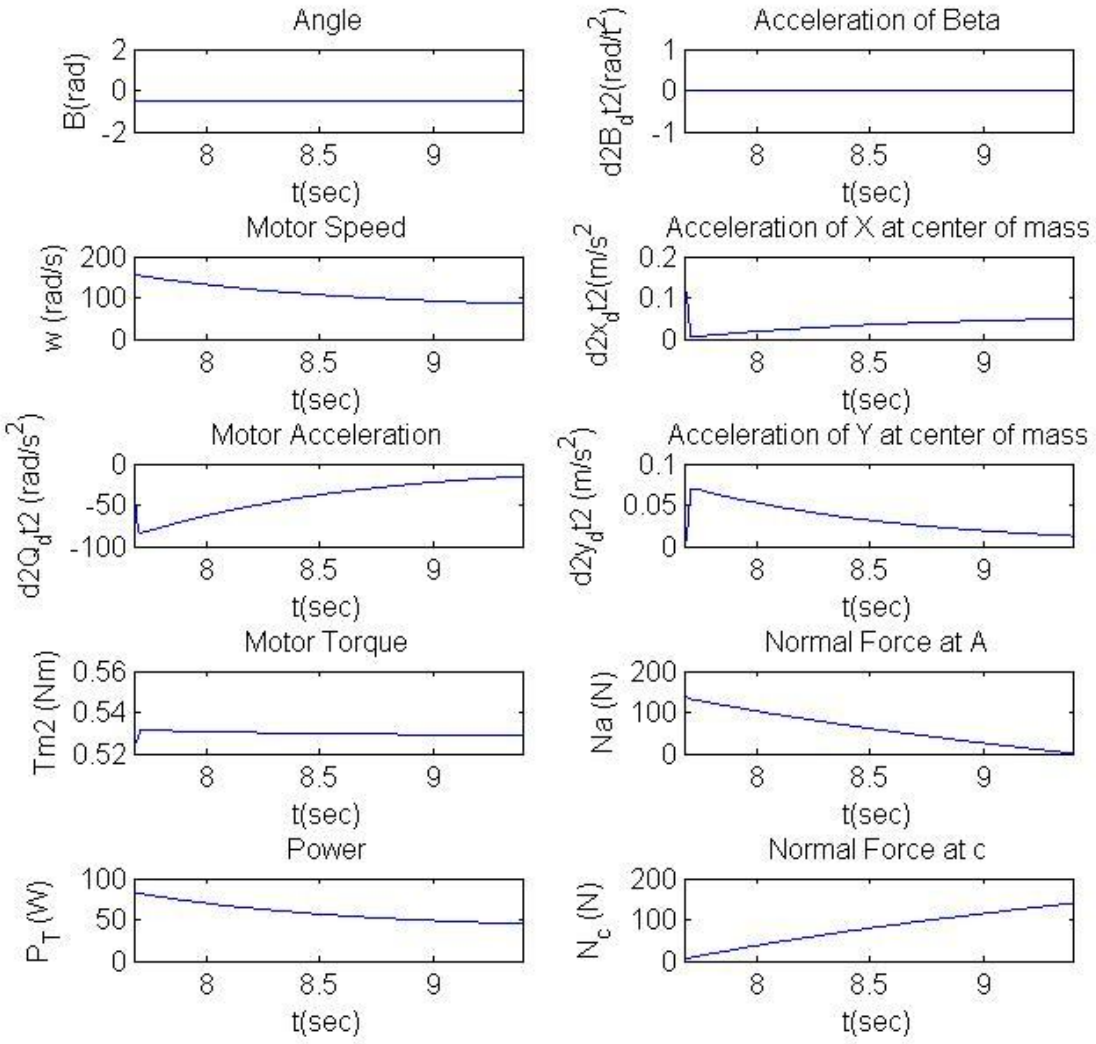


Figure D.10 : Simulation of the Tracked Robot during *Phase* 11

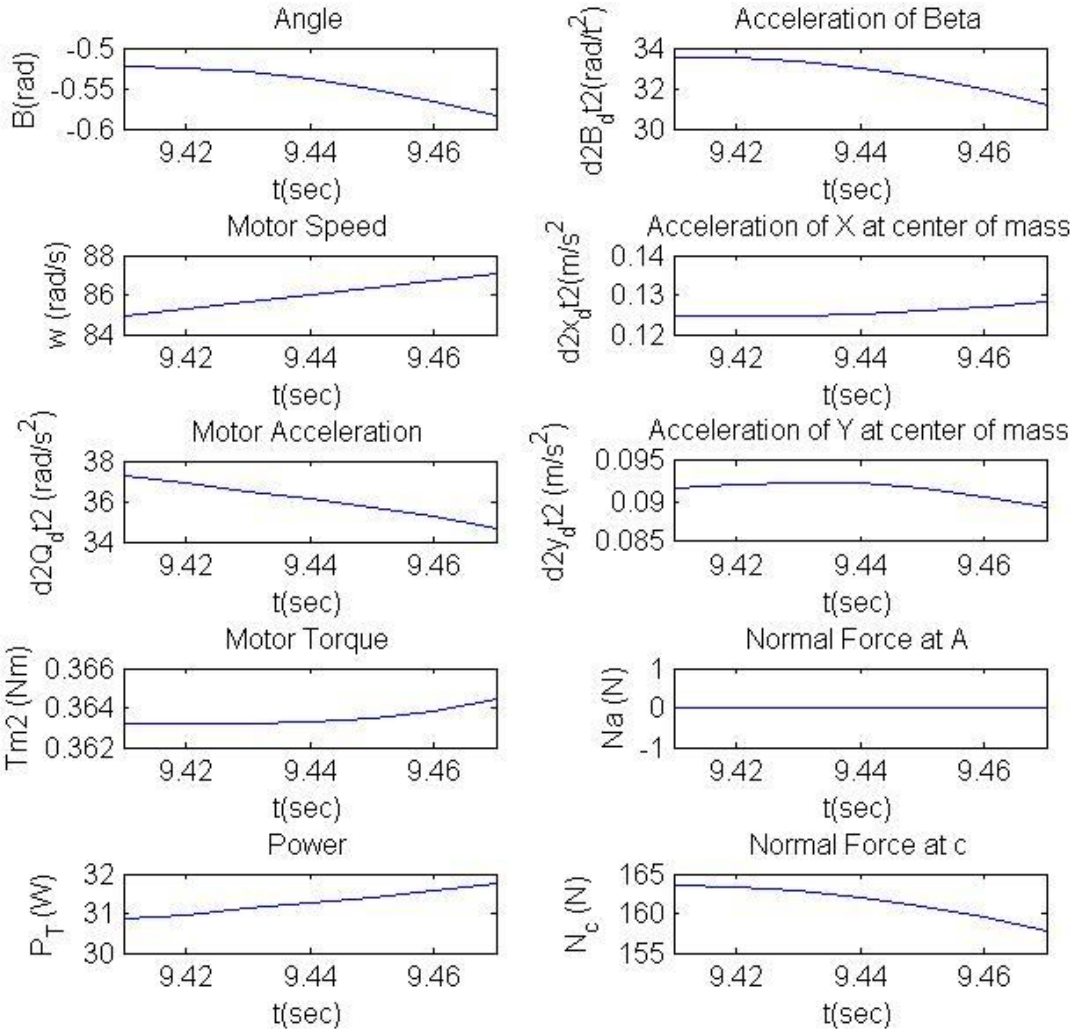


Figure D.11 : Simulation of the Tracked Robot during *Phase* 12

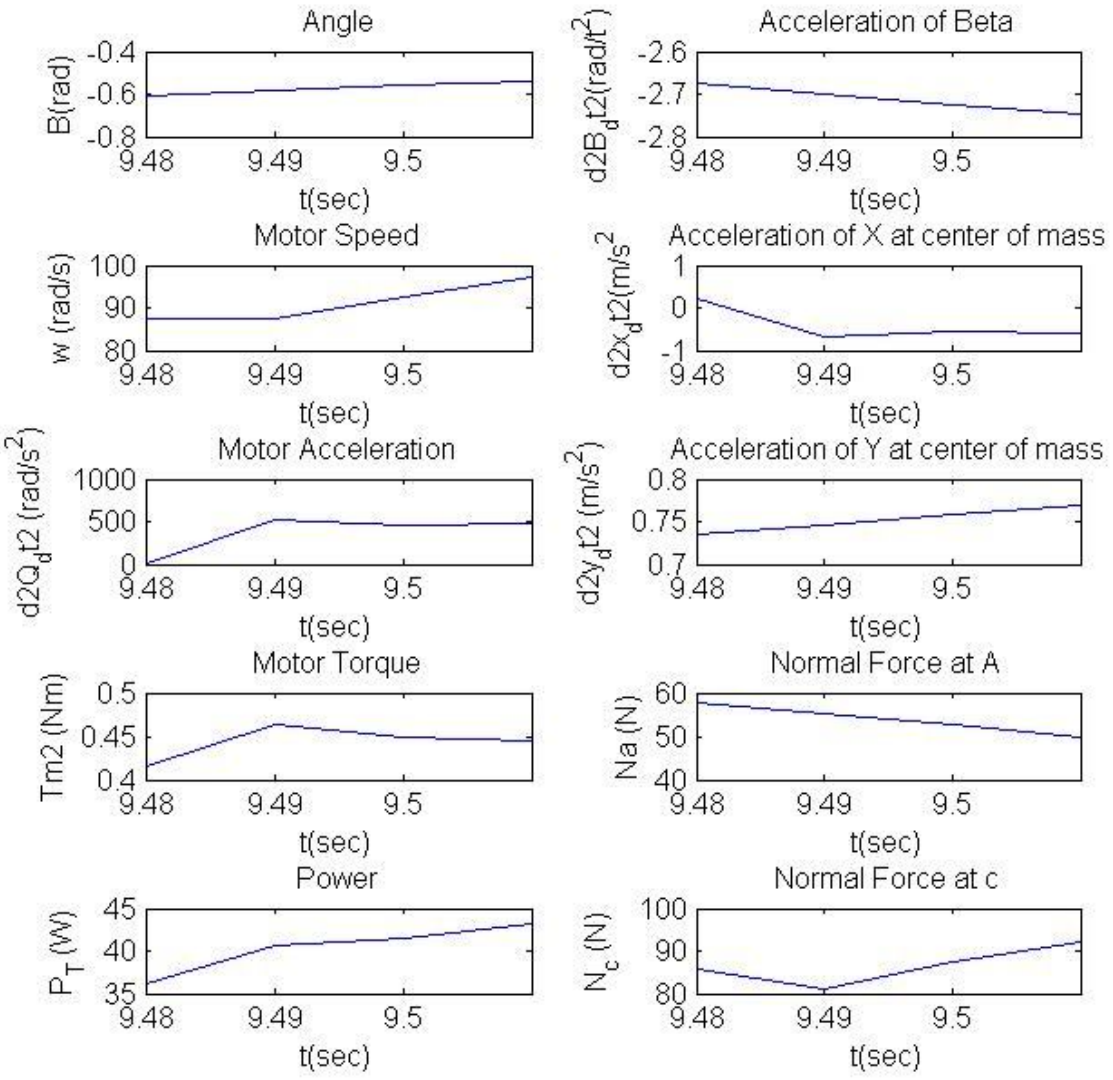


Figure D.12 : Simulation of the Tracked Robot during *Phase* 13

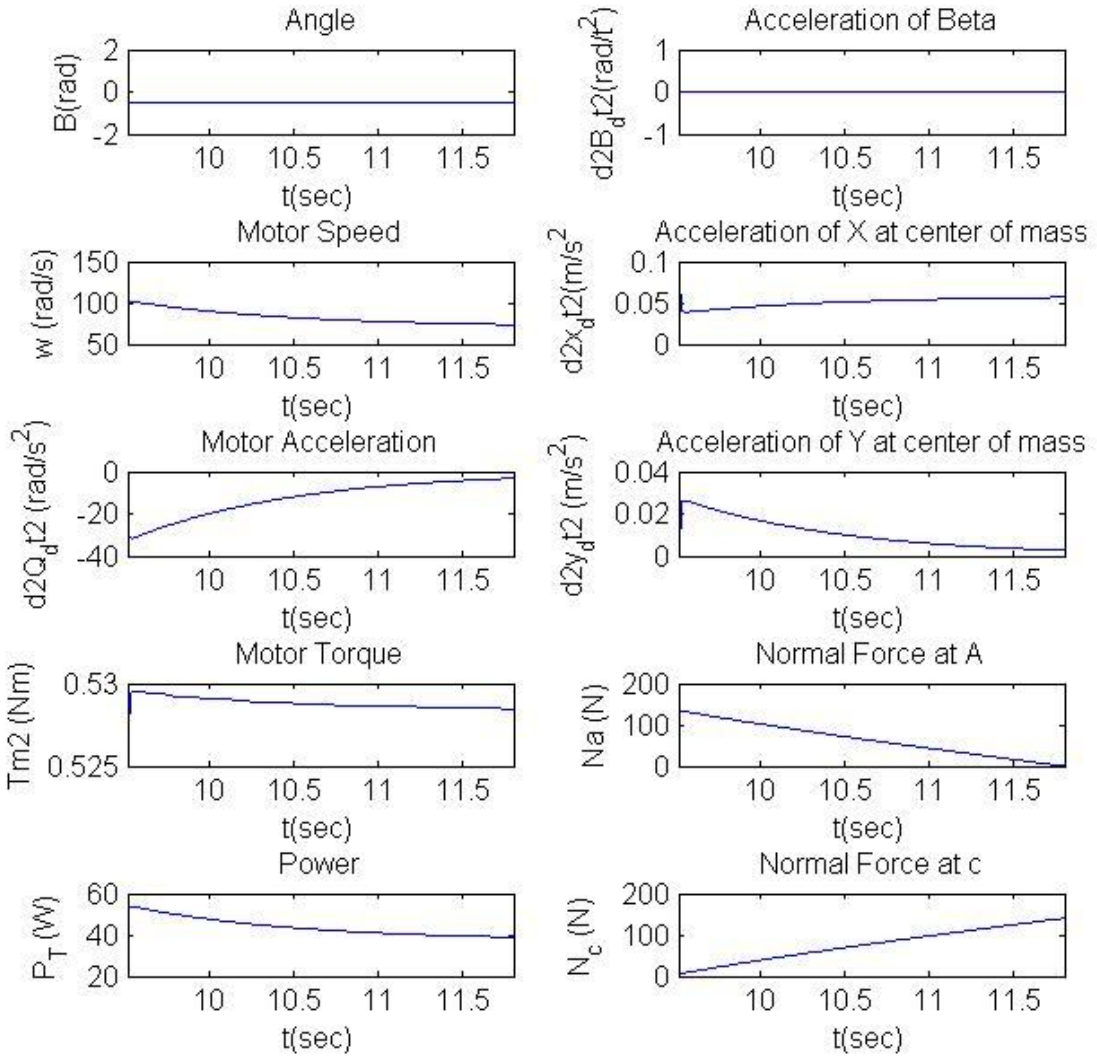


Figure D.13 Simulation of the Tracked Robot during *Phases* 14and 15

The time of Phase 14 is one “Time step”, therefore it can be considered as one point and combined with *Phase* 15.

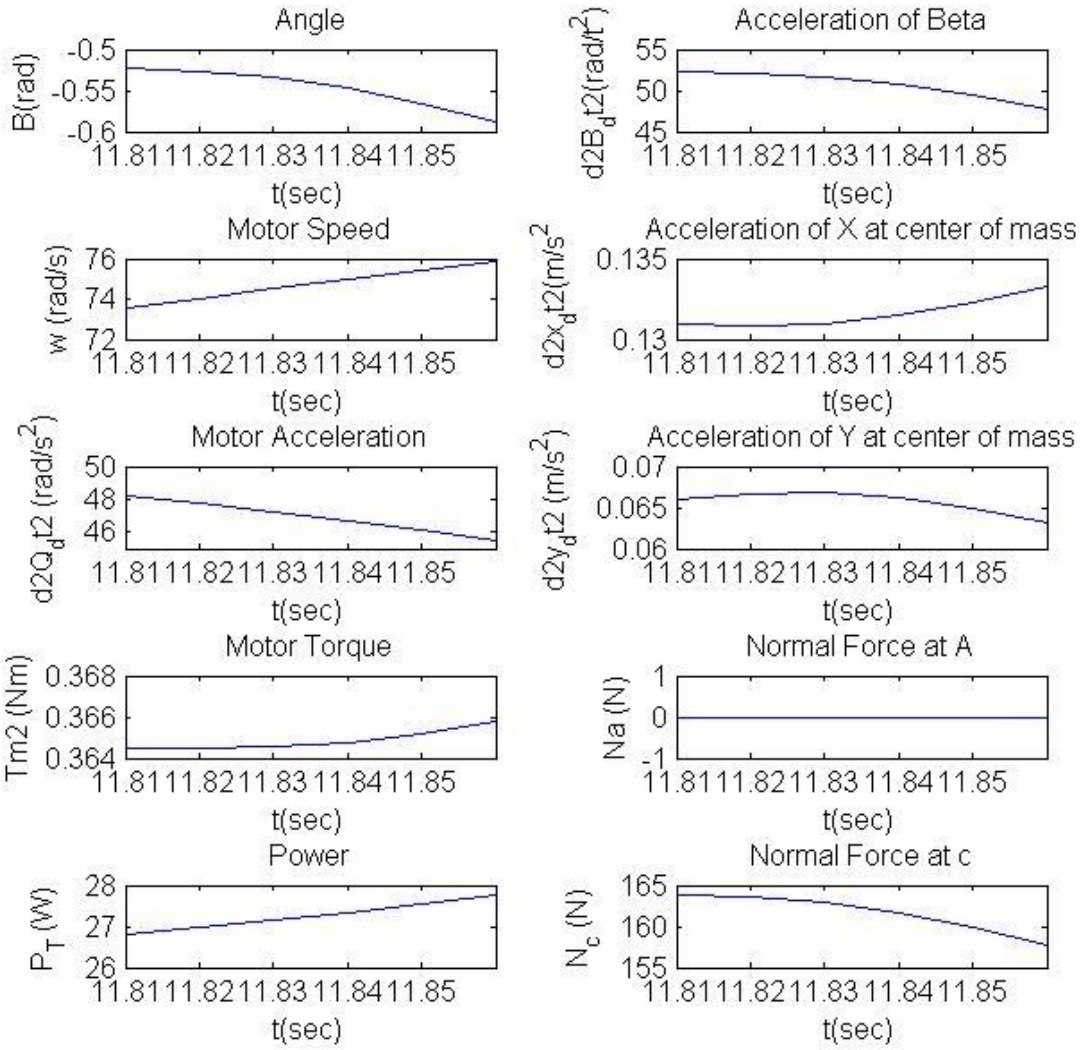


Figure D.14 Simulation of the Tracked Robot during *Phase* 16

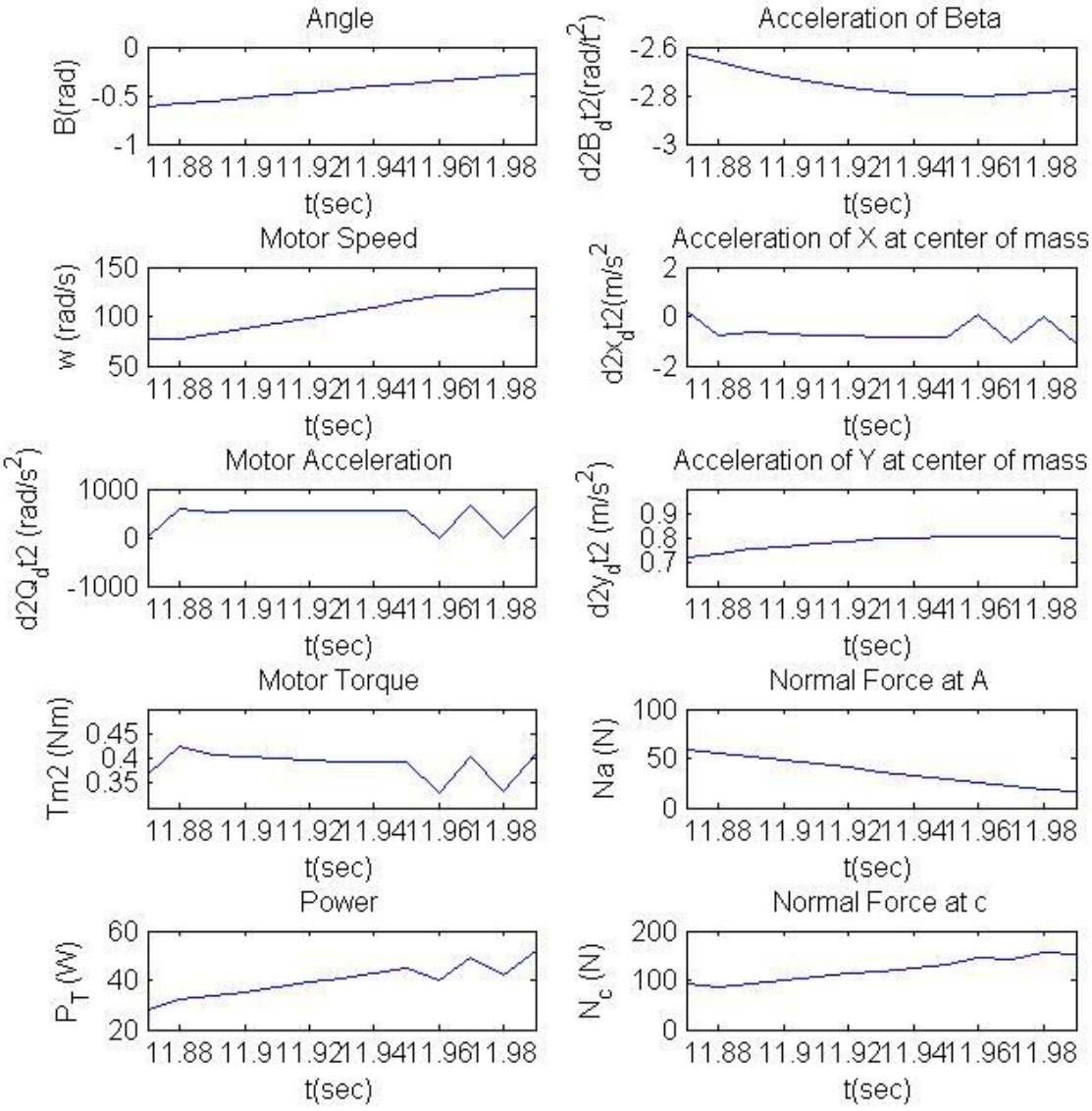


Figure D.15 Simulation of the Tracked Robot during *Phase* 17

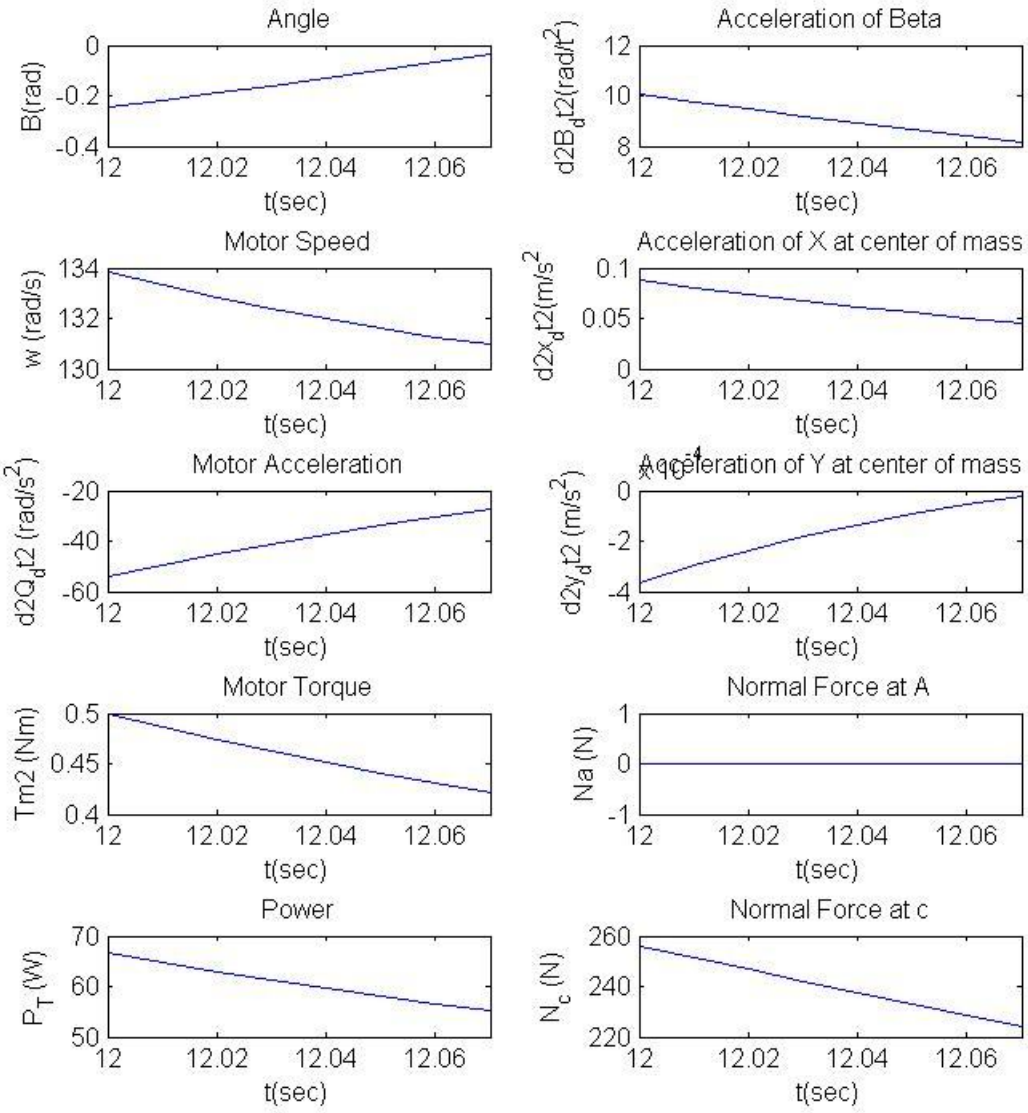


Figure D.16 Simulation of the Tracked Robot during *Phase* 18

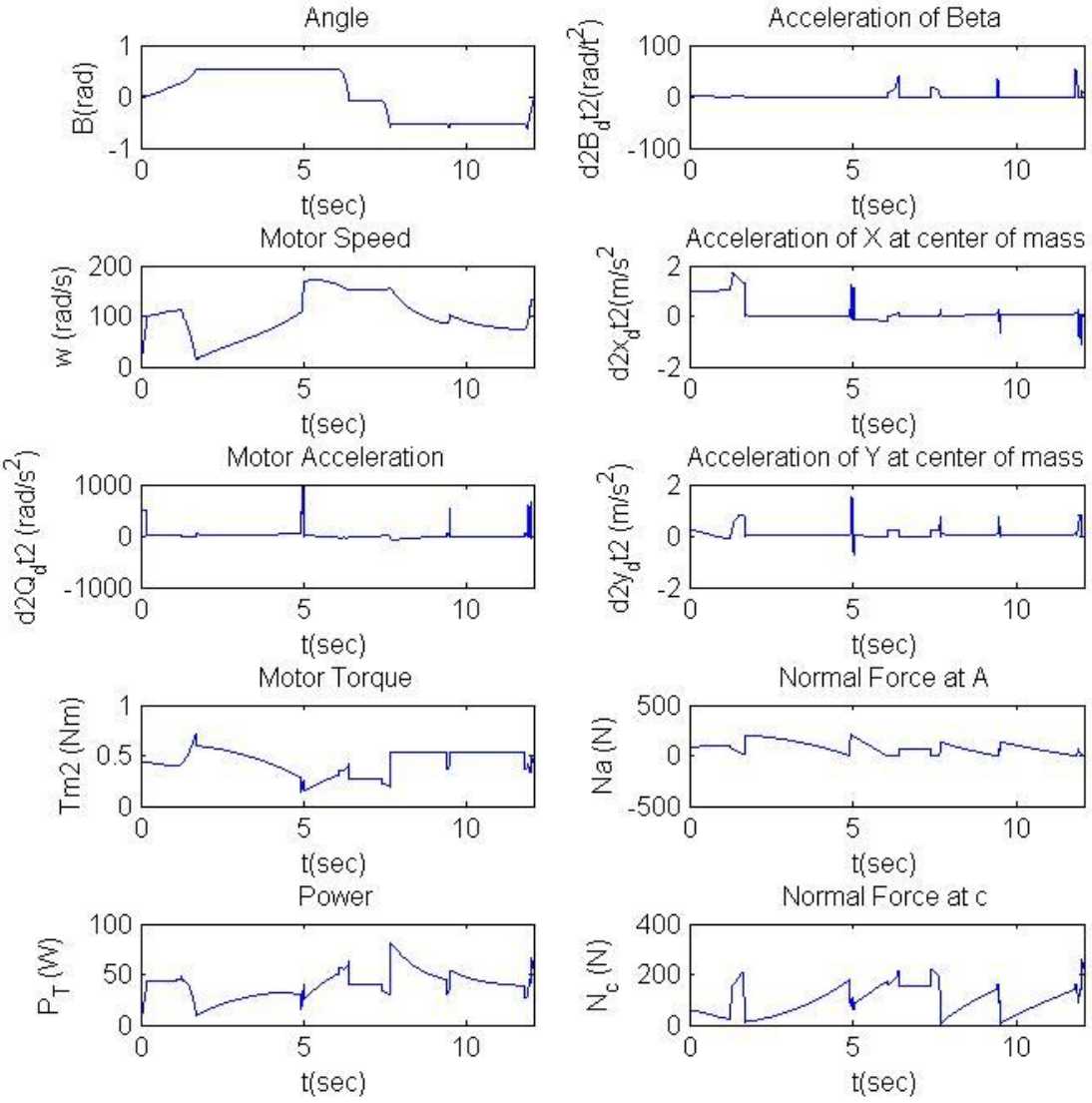


Figure D.17 : Simulation of the Tracked Robot from *Phase*1 to *Phase* 18.

Appendix F

F.1 Simulation codes

```
m=22;                                          %kg;
mg=220;                                        %kgf
L=0.65;                                        %Meter
La=0.374;
Lc=0.275;
Kapa=0.17;                                     %Meter
h=0.085;                                       %Meter
Theda=45*pi/180;                               %Radian
r=0.1;                                         %Meter
ti=0.01;
D=15.16*pi/180;
Muu=0.56;
Eff=0.5;
G=60;
V_rated=30; %Volts
T_stall_rated=1.13;
n_noload_rated=3380;
Bb=0.279;                                       %Meter
V=20; %Volts
T_stall=T_stall_rated*V/V_rated;
n_noload=n_noload_rated*V/V_rated;
w_noload=n_noload*2*pi/60;
t_a=0.2;
a_noload=w_noload/t_a;
w_slope=-w_noload/T_stall;
a_slope=-a_noload/T_stall;
I=22;
j=1;

% Preallocate for speed
i_final=1500;
```

```
B=zeros(1,i_final);
DeltaL=zeros(1,i_final);
L_1=zeros(1,i_final);
L_2=zeros(1,i_final);
L_T=zeros(1,i_final);
N_c=zeros(1,i_final);
Na=zeros(1,i_final);
Rr3=zeros(1,i_final);
Tm1=zeros(1,i_final);
Tm2=zeros(1,i_final);
d2A_dt2=zeros(1,i_final);
d2B_dt2=zeros(1,i_final);
d2Q_dt2=zeros(1,i_final);
d2x_dt2=zeros(1,i_final);
d2y_dt2=zeros(1,i_final);
dA_dt=zeros(1,i_final);
dB_dt=zeros(1,i_final);
dx_dt=zeros(1,i_final);
dy_dt=zeros(1,i_final);
q=zeros(1,i_final);
time=zeros(1,i_final);
w=zeros(1,i_final);
w_max=zeros(1,i_final);

%%                                                  PART 1
i=1;
time(i)=0;
B(i)=0.001;                                      %RadianD
dB_dt(i)=0;
w(i)=0;    %GuessW
X(i)=0;
 y(i)=0;
dx_dt(i)=0;
dy_dt(i)=0;
d2Q_dt2(i)=0;
L_2(i)=L/2;
L_T(i)=0.02;
L_1(i)=L/2;
d2A_dt2(i)=0;
dA_dt(i)=0;
DeltaL(i)=0;
DDD(i)=0;
check(i)=0;

while(B(i)<D)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)= Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i))-20)
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)= w_max(i);
```

```
            d2Q_dt2(i+1)=(w(i+1)-w(i));

        end
        q(i)=(Kapa-L*sin(B(i))/tan(Theda+B(i)));
        RR=(Lc+Kapa-L*sin(B(i))-q(i)*tan(B(i))+cos(B(i))*q(i));
        Na(i)=(mg*(Lc*cos(B(i))))/(cos(B(i))*L+q(i));

        d2B_dt2(i) = (Na(i)*(-Muu*Kapa-
L*cos(B(i))+q(i))+mg*(Lc*cos(B(i))+q(i)))/(I+m*RR^2);

        d2x_dt2(i) =1-(d2B_dt2(i))*((La).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =d2B_dt2(i)*(La*cos(B(i))-h*sin(B(i)));
        N_c(i) = ((m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(Theda+B(i))-
Muu*Na(i)))/(0.4*Muu);
        Tm2(i) = (d2y_dt2(i)*m+mg-Na(i)-N_c(i))/((sin(Theda+B(i))*G*Eff)/r);
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);

    end
    dB_dt(i+1)= d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;

    dx_dt(i+1)=d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;

    L_2(i)=L/2;
    L_T(i)=0.01;
    L_1(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;
    i=i+1;

end
i_end(j)=i-1;
j=j+1;

BB1=B(i(end)-1);
BB2=B(i(end));
E= 30*pi/180;

% % % %                                                          PART 2
% %
while(BB1<BB2&&BB2<E)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
```

```
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));

        end

        q(i)=(L*cos(B(i))-Kapa/tan(B(i)));

        RR(i)=((Lc-q(i)/cos(B(i)))^2+ (Lc*sin(B(i))+h*cos(B(i))-Kapa)^2)^0.5;

        Na(i)=mg*(L/2-q(i))/(L-1.5*q(i));
        d2B_dt2(i) = (Na(i)*(-Muu*Kapa-(L*cos(B(i))-q(i)))+mg*(Lc*cos(B(i)-
q(i))))/(I+m*RR(i)^2);
        d2x_dt2(i) =1-(d2B_dt2(i))*((La).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =d2B_dt2(i)*(La*cos(B(i))-h*sin(B(i)));
        N_c(i) = (1/Muu)*((m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(B(i))-
Muu*Na(i)));
        Tm2(i) = (d2y_dt2(i)*m+mg-Na(i)-N_c(i))/((sin(B(i))*G*Eff)/r);
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);




    end


    w(i+1)=d2Q_dt2(i)*ti+w(i);
    w_max(i)=Tm2(i)*w_slope+w_noload;
    if(w(i)<(w_max(i)))
        d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
        w(i+1)=d2Q_dt2(i+1)*ti+w(i);
    else
        w(i+1)=w_max(i);
        d2Q_dt2(i+1)=(w(i+1)-w(i));

    end

    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;
    L_2(i)=L/2;
    L_T(i)=0.02;
    L_1(i)=L/2;
    BB2=B(i(end));
    i=i+1;

end
```

```
i_end(j)=i-1;
j=j+1;
% % % % %%                                                    PART 3
L_2(i)=L_2(i(end)-1);
L_2(i)=L/2*cos(B(i));
L_T(i)=0.01;
L_1(i)=0;
B(i)=30*pi/180;
while(L_2(i-1)>0)
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)-50))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);

        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));


        end
        d2A_dt2(i+1)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i)=L/2+L_T(i);
        L_2(i)=L/2-L_T(i);
        d2y_dt2(i)=(r/60)* d2Q_dt2(i)-sin(B(i))*d2A_dt2(i);
        d2x_dt2(i)=cos(B(i))*d2A_dt2(i)-dA_dt(i)*sin(B(i));

    Na(i)=mg*(L_2(i))/(L/2*cos(B(i))*(cos(B(i))-
Muu*sin(B(i)))+Kapa*(Muu*cos(B(i))+sin(B(i))));

N_c(i)=mg+m*d2y_dt2(i)-Na(i)*(cos(B(i))-Muu*sin(B(i)))-
(Tm2(i)*(G*Eff/r))*sin(B(i));




Tm2(i)=(m*d2x_dt2(i)+Muu*N_c(i)+(Muu*cos(B(i))+sin(B(i)))*Na(i))/(G*(Eff/r))*
cos(B(i));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    d2B_dt2(i)=0;
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=30*pi/180;
```

```
    dx_dt(i+1)=d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;


    i=i+1;
end
i_end(j)=i-1;
j=j+1;
% % %                                                                PART 4

B(i)=30*pi/180;

aaa=1;
while(aaa<4);
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
            check(i)=1;
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));


        end

        d2A_dt2(i)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i)=L/2+L_T(i);
        L_2(i)=L/2-L_T(i);
        L_1(i)=L/2;
        L_2(i)=L/2;
        N_c(i)=(((Tm2(i)*(G*Eff/r))*((cos(B(i))*Muu*Kapa-
(L)*sin(B(i)))))+mg*(L_1(i)))/((Kapa*Muu+(L)*cos(B(i))));

        Na(i)=(mg*L_2(i)*cos(B(i))+Kapa*Muu)/((L)*cos(B(i)));

        d2y_dt2(i)=sin(B(i))*d2A_dt2(i);
        d2x_dt2(i)=0.2+cos(B(i))*d2A_dt2(i);
        Tm2(i)=(m*d2x_dt2(i)+Muu*N_c(i)-0.01*Na(i))/(G*(Eff/r)*(cos(B(i))));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end


    w_max(i)=Tm2(i)*w_slope+w_noload;
    if(w(i)<(w_max(i)-100))
```

```
        d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
        w(i+1)=d2Q_dt2(i+1)*ti+w(i);
    else
        w(i+1)=w_max(i);
        d2Q_dt2(i+1)=(w(i+1)-w(i));

    end
    w(i+1)= d2Q_dt2(i)*ti+w(i);

    aaa=aaa+1;
    dB_dt(i)=0;
    d2B_dt2(i)=0;
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
     B(i+1)=30*pi/180;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;

    B(i+1)=E;
    time(i+1)=time(i)+ti;
    i=i+1;

end
i_end(j)=i-1;
j=j+1;
% %%                                                          PART 5
L_2(i)=L_2(i(end)-1);
L_2(i)=L/2;
L_T(i-1)=0.01;
L_T(i)=0.01;
L_1(i)=L/2;

B(i)=30*pi/180;
while(L_2(i-1)>0)
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i))-20)
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);

        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));


        end
        d2A_dt2(i+1)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i)=L/2+L_T(i);
        L_2(i)=L/2-L_T(i);
```

```
        d2y_dt2(i)=(r/60)* d2Q_dt2(i)-sin(B(i))*d2A_dt2(i);
        d2x_dt2(i)=cos(B(i))*d2A_dt2(i)-dA_dt(i)*sin(B(i));

         Na(i)=mg*(L_2(i))/(L/2*cos(B(i))*(cos(B(i))-
Muu*sin(B(i)))+Kapa*(Muu*cos(B(i))+sin(B(i))));

N_c(i)=mg+m*d2y_dt2(i)-Na(i)*(cos(B(i))-Muu*sin(B(i)))-
(Tm2(i)*(G*Eff/r))*sin(B(i));

        Tm2(i)=(m*d2x_dt2(i)+Muu*N_c(i)+0.01*Na(i))/(G*(Eff/r));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    d2B_dt2(i)=0;
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
     B(i+1)=30*pi/180;
    dx_dt(i+1)=d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;


    i=i+1;
end
i_end(j)=i-1;
j=j+1;
%
%
%%                                                              part6
BB3 = B(i(end)-1);
BB4 = B(i(end));
K =0;
while(K<BB4&&BB4<(30.01*pi/180))

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)-200))
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));

        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)=0.2+ d2Q_dt2(i)*cos(B(i))*r/60-w(i)*r*sin(B(i))/60;
```

```
        dy_dt(i)= r*cos(B(i))*w(i)/60+w(i)*r*cos(B(i))/60;
        d2y_dt2(i)= d2Q_dt2(i)*sin(B(i))*r/60+w(i)*r*cos(B(i))/60;
        DeltaL(i)= 0.5*d2x_dt2(i)*time(i)^2+dx_dt(i)*time(i);

        N_c(i)=(mg-m*d2y_dt2(i)-Tm1(i)*(G*Eff/r*sin(B(i))));

        d2B_dt2(i)= ( (G*Eff/r)*Tm1(i)*( sin(B(i))*DeltaL(i)-cos(B(i))*( (
h/cos(B(i)) ) + DeltaL(i)*tan(B(i)) ) )...
            +N_c(i)*( DeltaL(i) - Muu*( ( h/cos(B(i)) )+DeltaL(i)*tan(B(i)) )
) )/I;

        Na(i)=0;

        Tm2(i)=(N_c(i)*Muu+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= B(i)- dB_dt(i)*ti - 0.5*d2B_dt2(i)*ti^2;
        dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
        X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
        dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
        y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;

        time(i+1)=time(i)+ti;
        BB4=B(i(end));
    end







    d2A_dt2(i)=0;
    dA_dt(i)=0;
    i=i+1;
end

i_end(j)=i-1;
j=j+1;
%%                      part 7,8

RT = i+100;
while(i<RT)


    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    w_max(i)= Tm2(i)*w_slope+w_noload;
    w(i+1)= w_max(i);
    d2Q_dt2(i+1)=(w(i+1)-w(i));

    Na(i)=68;
    N_c(i) = mg-Na(i);
```

```
    d2B_dt2(i) =0 ;
    d2y_dt2(i) =0;
    d2x_dt2(i) = (Tm2(i)*(G*Eff/r)-0.3*mg)/m;

    Tm1(i)=Tm2(i);
    dB_dt(i+1)= 0;
    B(i+1)=B(i);
    dx_dt(i+1)=d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=0;
    y(i+1)=0;

    time(i+1)=time(i)+ti;
    L_2(i)=L/2;
    L_T(i)=0.01;
    L_1(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;
    i=i+1;

end
i_end(j)=i-1;
j=j+1;
%                                                          PART 9
while(B(i)>(-34.34*pi/180))
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));

        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)= d2Q_dt2(i)*cos(B(i))*r/60-w(i)*r*sin(B(i))/60;
        dy_dt(i)= r*cos(B(i))*w(i)/60+w(i)*r*cos(B(i))/60;
        d2y_dt2(i)= d2Q_dt2(i)*sin(B(i))*r/60+w(i)*r*cos(B(i))/60;
        DeltaL(i)= 0.5*d2x_dt2(i)*time(i)^2+dx_dt(i)*time(i);

        N_c(i)=(mg-m*d2y_dt2(i)+Tm1(i)*G*Eff/r*sin(B(i)));

        d2B_dt2(i)= ( (G*Eff/r)*Tm1(i)*( sin(B(i))*DeltaL(i)-cos(B(i))*( (
h/cos(B(i)) ) + DeltaL(i)*tan(B(i)) ) )...
            +N_c(i)*( DeltaL(i) + Muu*( ( h/cos(B(i)) )+DeltaL(i)*tan(B(i)) )
) )/I;

        Na(i)=0;

        Tm2(i)=(N_c(i)*Muu+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
```

```
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= (B(i)- dB_dt(i)*ti - 0.5*d2B_dt2(i)*ti^2);
        dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
        X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
        dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
        y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
        time(i+1)=time(i)+ti;
%         BB6=B(i(end));
    end
    L_T(i)=0;
    L_1(i)=L/2;
    L_2(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;
    i=i+1;
end

ff= -30*pi/180;
i_end(j)=i-1;
j=j+1;
% % %                                                          PART 10
while(B(i)<ff)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));
        end

        Rr3(i)=((h)^2+((Kapa/cos(B(i)))-Lc)^2)^0.5;

Na(i)= -mg*(Lc*Muu*cos(B(i))-h*(sin(B(i)))-h*0.5-
Muu*Kapa)/(Kapa*((1/tan(B(i)))*(+Muu*sin(B(i))+cos(B(i)))+(sin(B(i))+Muu*cos(
B(i))))) ;

d2B_dt2(i)=(((-Na(i)*Kapa)*((1/tan(B(i)))*(Muu*sin(B(i))+cos(B(i)))+(-
sin(B(i))+Muu*cos(B(i))))+(Lc-h*sin(B(i)))*mg)/(I+m*(Rr3(i))^2));
        d2x_dt2(i) =-(r/60)*
d2Q_dt2(i)+(d2B_dt2(i))*((Lc).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =d2B_dt2(i)*(Lc*cos(B(i))-h*sin(B(i)));

        N_c(i) = (m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(B(i))-
Muu*Na(i)*cos(B(i))+Na(i)*sin(B(i)))/Muu;
```

```
        Tm2(i) = (d2y_dt2(i)*m-
mg+Na(i)*cos(B(i))+N_c(i))/((sin(B(i))*G*Eff)/r);
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;
    w(i+1)=d2Q_dt2(i)*ti+w(i);
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;
    i=i+1;

end
i_end(j)=i-1;
j=j+1;
%                                                          PART 11

B(i)=-30*pi/180;
L_1(i)=L/2;
L_2(i)=L/2;
L_T(i-1)=0.01;
 while(L_1(i)>0)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
            check(i)=1;
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));

        end
        d2A_dt2(i+1)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i+1)=L/2-L_T(i);
        L_2(i+1)=L/2+L_T(i);
        d2y_dt2(i)=sin(B(i))*d2A_dt2(i);
        d2x_dt2(i)=cos(B(i))*d2A_dt2(i)-dA_dt(i)*sin(B(i));

%
        Na(i)=mg*L_1(i)*cos(B(i))/(((L/2)*cos(B(i)))*(+Muu*cos(B(i))-
sin(B(i)))-Kapa*(-Muu*cos(B(i))+sin(B(i))));
        N_c(i)=mg+d2y_dt2(i)+Tm2(i)*(G*(Eff/r)*sin(B(i)))-
Na(i)*(Muu*cos(B(i))+sin(B(i)))/(Muu*cos(B(i))+sin(B(i)));
```

```
        Tm2(i)=(m*d2x_dt2(i)+(+Muu*cos(B(i))-
sin(B(i)))*N_c(i)+(Muu*cos(B(i))-sin(B(i)))*Na(i))/(G*(Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    d2B_dt2(i+1)=0;
    dB_dt(i+1)=0;
    B(i+1)=-30*pi/180;
    dx_dt(i+1)=d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    i=i+1;
 end
i_end(j)=i-1;
j=j+1;
% %                                                    PART 12

while(B(i)>(-34.34*pi/180))
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i))-200)
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));

        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)= d2Q_dt2(i)*cos(B(i))*r/60-w(i)*r*sin(B(i))/60;
        dy_dt(i)= r*cos(B(i))*w(i)/60+w(i)*r*cos(B(i))/60;
        d2y_dt2(i)= d2Q_dt2(i)*sin(B(i))*r/60+w(i)*r*cos(B(i))/60;
        DeltaL(i)= 0.5*d2x_dt2(i)*time(i)^2+dx_dt(i)*time(i);

        N_c(i)=(mg-m*d2y_dt2(i)+Tm1(i)*G*Eff/r*sin(B(i)));

        d2B_dt2(i)= (- (G*Eff/r)*Tm1(i)*( sin(B(i))*DeltaL(i)-cos(B(i))*( (
h/cos(B(i)) ) + DeltaL(i)*tan(B(i)) ) )...
            +N_c(i)*( DeltaL(i) + Muu*( ( h/cos(B(i)) )+DeltaL(i)*tan(B(i)) )
) )/I;

        Na(i)=0;

        Tm2(i)=(N_c(i)*Muu+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
```

```
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= (B(i)- dB_dt(i)*ti - 0.5*d2B_dt2(i)*ti^2);
        dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
        X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
        dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
        y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
        time(i+1)=time(i)+ti;

    end
    L_T(i)=0;
    L_1(i)=L/2;
    L_2(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;

    i=i+1;
end
fff= -30*pi/180;
i_end(j)=i-1;
j=j+1;
%                                                          PART 13
while(B(i)<fff)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));
        end


Na(i)= -mg*(Lc*Muu*cos(B(i))-h*(sin(B(i)))-h*0.5-
Muu*Kapa)/(Kapa*((1/tan(B(i)))*(+Muu*sin(B(i))+cos(B(i)))+(sin(B(i))+Muu*cos(
B(i))))) ;

d2B_dt2(i)=(((-Na(i)*Kapa)*((1/tan(B(i)))*(Muu*sin(B(i))+cos(B(i)))+(-
sin(B(i))+Muu*cos(B(i))))+(Lc-h*sin(B(i))-0.0425)*mg)/(I+m*(Rr3(i))^2));
        d2x_dt2(i) =-(r/60)*
d2Q_dt2(i)+(d2B_dt2(i))*((Lc).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =-d2B_dt2(i)*(Lc*cos(B(i))-h*sin(B(i)));

        N_c(i) = (m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(B(i))-
Muu*Na(i)*cos(B(i))+Na(i)*sin(B(i)))/Muu;
```

```
        Tm2(i) = (d2y_dt2(i)*m-
mg+Na(i)*cos(B(i))+N_c(i))/((sin(B(i))*G*Eff)/r);
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;
    w(i+1)=d2Q_dt2(i)*ti+w(i);
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;
    i=i+1;

end
i_end(j)=i-1;
j=j+1;

% % % % % %                                                          PART
15

B(i)=-30*pi/180;




L_1(i)=L/2;
L_2(i)=0;
L_T(i-1)=0.01;
 while(L_1(i)>0)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
            check(i)=1;
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));

        end
        d2A_dt2(i+1)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i+1)=L/2-L_T(i);
        L_2(i+1)=L/2+L_T(i);
        d2y_dt2(i)=sin(B(i))*d2A_dt2(i);
```

```
        d2x_dt2(i)=cos(B(i))*d2A_dt2(i)-dA_dt(i)*sin(B(i));
        Na(i)=mg*L_1(i)*cos(B(i))/(((L/2)*cos(B(i)))*(+Muu*cos(B(i))-
sin(B(i)))-Kapa*(-Muu*cos(B(i))+sin(B(i))));
        N_c(i)=mg+d2y_dt2(i)+Tm2(i)*(G*(Eff/r)*sin(B(i)))-
Na(i)*(Muu*cos(B(i))+sin(B(i)))/(Muu*cos(B(i))+sin(B(i)));

        Tm2(i)=(-m*d2x_dt2(i)+(+Muu*cos(B(i))-
sin(B(i)))*N_c(i)+(Muu*cos(B(i))-sin(B(i)))*Na(i))/(G*(Eff/r)*cos(B(i)));
error=(Tm2(i)-Tm1(i))/Tm2(i);
Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    d2B_dt2(i+1)=0;
    dB_dt(i+1)=0;

    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
      B(i+1)=-30*pi/180;
      i=i+1;
 end
i_end(j)=i-1;
j=j+1;
% % % % %                                                          PART 16
%
while(B(i)>(-34.34*pi/180))
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i))-200)
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));

        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)= d2Q_dt2(i)*cos(B(i))*r/60-w(i)*r*sin(B(i))/60;
        dy_dt(i)= r*cos(B(i))*w(i)/60+w(i)*r*cos(B(i))/60;
        d2y_dt2(i)= d2Q_dt2(i)*sin(B(i))*r/60+w(i)*r*cos(B(i))/60;
        DeltaL(i)= 0.5*d2x_dt2(i)*time(i)^2+dx_dt(i)*time(i);

        N_c(i)=(mg-m*d2y_dt2(i)+Tm1(i)*G*Eff/r*sin(B(i)));

        d2B_dt2(i)= ( (G*Eff/r)*Tm1(i)*( sin(B(i))*DeltaL(i)-cos(B(i))*( (
h/cos(B(i)) ) + DeltaL(i)*tan(B(i)) ) )...
```

```
            +N_c(i)*( DeltaL(i) + Muu*( ( h/cos(B(i)) )+DeltaL(i)*tan(B(i)) )
) )/I;

        Na(i)=0;

        Tm2(i)=(N_c(i)*Muu+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= (B(i)- dB_dt(i)*ti - 0.5*d2B_dt2(i)*ti^2);
        dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
        X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
        dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
        y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
        time(i+1)=time(i)+ti;
%         BB6=B(i(end));
    end
    L_T(i)=0;
    L_1(i)=L/2;
    L_2(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;

    i=i+1;
end
f= -15.16*pi/180;
i_end(j)=i-1;
j=j+1;
% %                                                          PART 17
while(B(i)<f)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));
        end

        Rr3(i)=((h)^2+((Kapa/cos(B(i)))-Lc)^2)^0.5;

Na(i)= -mg*(Lc*Muu*cos(B(i))-h*(sin(B(i)))-h*0.5-
Muu*Kapa)/(Kapa*((1/tan(B(i)))*(+Muu*sin(B(i))+cos(B(i)))+(sin(B(i))+Muu*cos(
B(i))))) %#ok<NOPTS>

d2B_dt2(i)=(((-Na(i)*Kapa)*((1/tan(B(i)))*(Muu*sin(B(i))+cos(B(i)))+(-
sin(B(i))+Muu*cos(B(i))))+(Lc-h*sin(B(i))-0.0425)*mg)/(I+m*(Rr3(i))^2));
```

```
        d2x_dt2(i) =-(r/60)*
d2Q_dt2(i)+(d2B_dt2(i))*((Lc).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =-d2B_dt2(i)*(Lc*cos(B(i))-h*sin(B(i)));

        N_c(i) = (m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(B(i))-
0.2*Na(i)*cos(B(i))+Na(i)*sin(B(i)))/Muu;

        Tm2(i) = (d2y_dt2(i)*m-
mg+Na(i)*cos(B(i))+N_c(i))/((sin(B(i))*G*Eff)/r);
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;
    i=i+1;

end
Z=0;
i_end(j)=i-1;
j=j+1;
%                                          PART 18
while(B(i)<Z)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));
        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)= d2Q_dt2(i)*cos(B(i))*r/60;
        dy_dt(i)= r*cos(B(i))*w(i)/60+w(i)*r*cos(B(i))/60;
        d2y_dt2(i)= d2Q_dt2(i)*sin(B(i))*r/60/60;

     N_c(i)=(mg+m*d2y_dt2(i)-Tm1(i)*sin(B(i))*G*Eff/r);
        d2B_dt2(i)= ((( (G*Eff/r*Tm1(i)))*cos(B(i)*( Lc*cos(B(i))-
h*sin(B(i)))+sin(B(i))*( Lc*sin(B(i))-h*cos(B(i))))+N_c(i)*(( Lc*cos(B(i))-
h*sin(B(i)))-Muu*( Lc*sin(B(i))+h*cos(B(i)))))/I);

        Na(i)=0;

        Tm2(i)=(N_c(i)*Muu+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
```

```
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= (B(i)+dB_dt(i)*ti+ 0.5*d2B_dt2(i)*ti^2);
        time(i+1)=time(i)+ti;

    end

    i=i+1;
end
i_end(j)=i-1;

%                               PLOT
j=1;
while(j<18)

    if(j==1)
        i=1:i_end(j);
    elseif(j==17)
        i=1:i_end(16);
    else
        i=(i_end(j-1)+1):i_end(j);
    end


    figure(j)
    subplot(5,2,1)
    plot(time(i),B(i))
    xlabel('t(sec)')
    ylabel('B(rad)')
    if(time(max(i))>time(min(i)))
        xlim([time(min(i)) time(max(i))])
    end
    title('Angle')

    subplot(5,2,2)
    plot(time(i),d2B_dt2(i));
    xlabel('t(sec)')
    ylabel('d2B_dt2(rad/t^2)')
    if(time(max(i))>time(min(i)))
        xlim([time(min(i)) time(max(i))])
    end
    title('Angular Acceleration of Center of Mass ')

    subplot(5,2,3)
    plot(time(i), w(i));
    xlabel('t(sec)')
    ylabel('w (rad/s)')
    if(time(max(i))>time(min(i)))
        xlim([time(min(i)) time(max(i))])
    end
    title('Motor Speed')
```

```
subplot(5,2,4)
plot(time(i), d2x_dt2(i));
xlabel('t(sec)')
ylabel(' d2x_dt2(m/s^2)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Acceleration of center of mass in X Direction')

subplot(5,2,5)
plot(time(i), d2Q_dt2(i));
xlabel('t(sec)')
ylabel('d2Q_dt2 (rad/s^2)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Motor Acceleration')

subplot(5,2,6)
plot(time(i),d2y_dt2(i));
xlabel('t(sec)')
ylabel('d2y_dt2 (m/s^2)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Acceleration of  center of mass in Y Direction')

subplot(5,2,7)
plot(time(i), Tm2(i));
xlabel('t(sec)')
ylabel('Tm2 (Nm)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Motor Torque')

subplot(5,2,8)
plot(time(i), Na(i));
xlabel('t(sec)')
ylabel('Na (N)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Normal Force at A')

subplot(5,2,9)
plot(time(i),(Tm2(i).*w(i)));
xlabel('t(sec)')
ylabel('P_T (W)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Power')

subplot(5,2,10)
plot(time(i), N_c(i));
```

```
    xlabel('t(sec)')
    ylabel('N_c (N)')
    if(time(max(i))>time(min(i)))
        xlim([time(min(i)) time(max(i))])
    end
    title('Normal Force at c')

j=j+1;
end
```

```
m=22;                                          %kg;
mg=220;                                        %kgf
L=0.65;                                        %Meter
La=0.374;
Lc=0.275;
Kapa=0.17;                                     %Meter
h=0.085;                                       %Meter
Theda=45*pi/180;                               %Radian
r=0.1;                                         %Meter
ti=0.01;
D=15.16*pi/180;
Muu=0.56;
Eff=0.5;
G=60;
V_rated=30; %Volts
T_stall_rated=1.13;
n_noload_rated=3380;
Bb=0.279;                                       %Meter
V=20; %Volts
T_stall=T_stall_rated*V/V_rated;
n_noload=n_noload_rated*V/V_rated;
w_noload=n_noload*2*pi/60;
t_a=0.2;
a_noload=w_noload/t_a;
w_slope=-w_noload/T_stall;
a_slope=-a_noload/T_stall;
I=22;
j=1;

% Preallocate for speed
i_final=1500;
B=zeros(1,i_final);
DeltaL=zeros(1,i_final);
L_1=zeros(1,i_final);
L_2=zeros(1,i_final);
L_T=zeros(1,i_final);
N_c=zeros(1,i_final);
Na=zeros(1,i_final);
Rr3=zeros(1,i_final);
Tm1=zeros(1,i_final);
```

```
Tm2=zeros(1,i_final);
d2A_dt2=zeros(1,i_final);
d2B_dt2=zeros(1,i_final);
d2Q_dt2=zeros(1,i_final);
d2x_dt2=zeros(1,i_final);
d2y_dt2=zeros(1,i_final);
dA_dt=zeros(1,i_final);
dB_dt=zeros(1,i_final);
dx_dt=zeros(1,i_final);
dy_dt=zeros(1,i_final);
q=zeros(1,i_final);
time=zeros(1,i_final);
w=zeros(1,i_final);
w_max=zeros(1,i_final);

%%                                                  PART 1
i=1;
time(i)=0;
B(i)=0.001;                                       %RadianD
dB_dt(i)=0;
w(i)=0;    %GuessW
X(i)=0;
 y(i)=0;
dx_dt(i)=0;
dy_dt(i)=0;
d2Q_dt2(i)=0;
L_2(i)=L/2;
L_T(i)=0.02;
L_1(i)=L/2;
d2A_dt2(i)=0;
dA_dt(i)=0;
DeltaL(i)=0;
DDD(i)=0;
check(i)=0;
while(B(i)<D)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)= Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i))-20)

            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)= w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));

        end
        q(i)=(Kapa-L*sin(B(i))/tan(Theda+B(i)));
        RR(i)=(Lc+Kapa-L*sin(B(i))-q(i)*tan(B(i))+cos(B(i))*q(i));
        Na(i)=(220*Lc*cos(B(i))+q(i))/(cos(B(i))* L+q(i));
%         Na(i)=mg*(Lc*cos(B(i)))/(cos(B(i))*L+q(i));
%
```

```
        d2B_dt2(i) = (Na(i)*(-0.56*Kapa-L*cos(B(i))-
q(i))+mg*(Lc*cos(B(i))+q(i)))/(I+m*RR(i)^2);

        d2x_dt2(i) =1-(d2B_dt2(i))*((La).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =+d2B_dt2(i)*(La*cos(B(i))-h*sin(B(i)));

        N_c(i) =((-m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(Theda+B(i))-
0.1*Na(i)))/ (sin(B(i)+Theda)+Muu*cos(B(i)+Theda));
        Tm2(i) = (d2y_dt2(i)*m+mg-Na(i)-
N_c(i)*(0.1*sin(B(i)+Theda)+cos(B(i)+Theda)))/((sin(Theda+B(i))*G*Eff)/r);

        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);

    end
    dB_dt(i+1)= d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;
    time(i+1)=time(i)+ti;
    L_2(i)=L/2;
    L_T(i)=0.01;
    L_1(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;
    i=i+1;

end

i_end(j)=i-1;
j=j+1;

BB1=B(i(end)-1);
BB2=B(i(end));
E= 30*pi/180;

% % % %                                                          PART 2
% %
while(BB1<BB2&&BB2<E)
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);


    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)=w_max(i);
```

```
            d2Q_dt2(i+1)=(w(i+1)-w(i));

        end

        q(i)=(L*cos(B(i))-Kapa/tan(B(i)));

        RR(i)=((Lc-q(i)/cos(B(i)))^2+ (L/2*sin(B(i))+h*cos(B(i))-
Kapa)^2)^0.5;

        MM(i)=((L*cos(B(i))-h*sin(B(i)))-Muu*(Lc*sin(B(i))+h*cos(B(i))));
        NN(i)=(cos(B(i))+Muu*sin(B(i)))*(Lc*cos(B(i))-q(i));

MMM(i)=(sin(B(i))+Muu*cos(B(i)))*(h*sin(B(i))+h*sin(B(i))+Lc*sin(B(i))-Kapa);


        Na(i)=220*(NN(i)+MMM(i))/(MMM(i)+NN(i)+MM(i));
        d2B_dt2(i) = (Na(i)*(Muu*(Lc*sin(B(i))+h*cos(B(i)))-(L*cos(B(i))-
q(i)))+mg*(L/2*cos(B(i))))/(I+m*RR(i)^2);
        d2x_dt2(i) =+2- (d2B_dt2(i))*((La).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =d2B_dt2(i)*(La*cos(B(i))-h*sin(B(i)));
        N_c(i) =
(1/(sin(B(i))+Muu*cos(B(i))))*((m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(B(i))-
Muu*Na(i)));

        Tm2(i) = (d2y_dt2(i)*m+mg-Na(i)-N_c(i)*(cos(B(i))-
Muu*sin(B(i))))/((sin(B(i))*G*Eff)/r);
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);




    end


    w(i+1)=d2Q_dt2(i)*ti+w(i);
    w_max(i)=Tm2(i)*w_slope+w_noload;
    if(w(i)<(w_max(i)))
        d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
        w(i+1)=d2Q_dt2(i+1)*ti+w(i);
    else
        w(i+1)=w_max(i);
        d2Q_dt2(i+1)=(w(i+1)-w(i));

    end

    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;
```

```
    time(i+1)=time(i)+ti;
    L_2(i)=L/2;
    L_T(i)=0.02;
    L_1(i)=L/2;
    BB2=B(i(end));
    i=i+1;

end

i_end(j)=i-1;
j=j+1;
% % % % %%                                                          PART 3
L_2(i)=L_2(i(end)-1);
L_2(i)=L/2*cos(B(i));
L_T(i)=0.01;
L_1(i)=0;
B(i)=30*pi/180;
while(L_2(i-1)>0)
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);

        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));


        end
        d2A_dt2(i+1)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i)=L/2+L_T(i);
        L_2(i)=L/2-L_T(i);
        d2y_dt2(i)=(r/60)* d2Q_dt2(i)-sin(B(i))*d2A_dt2(i);
        d2x_dt2(i)=cos(B(i))*d2A_dt2(i)-dA_dt(i)*sin(B(i));

    Na(i)=mg*(L_2(i))/(L/2*cos(B(i))*(cos(B(i))-
Muu*sin(B(i)))+Kapa*(Muu*cos(B(i))+sin(B(i))));

N_c(i)=mg+m*d2y_dt2(i)-Na(i)*(cos(B(i))-Muu*sin(B(i)))-
(Tm2(i)*(G*Eff/r))*sin(B(i));




Tm2(i)=(m*d2x_dt2(i)+Muu*N_c(i)+(Muu*cos(B(i))+sin(B(i)))*Na(i))/(G*(Eff/r))*
cos(B(i));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
```

```
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    d2B_dt2(i)=0;
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=30*pi/180;
    dx_dt(i+1)=d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;


    i=i+1;
end

i_end(j)=i-1;
j=j+1;
% % %                                                                 PART 4

B(i)=30*pi/180;

aaa=1;
while(aaa<4);
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
            check(i)=1;
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));


        end

        d2A_dt2(i)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i)=L/2+L_T(i);
        L_2(i)=L/2-L_T(i);
        L_1(i)=L/2;
        L_2(i)=L/2;
        N_c(i)=(((Tm2(i)*(G*Eff/r))*((cos(B(i))*Muu*Kapa-
(L)*sin(B(i)))))+mg*(L_1(i)))/((Kapa*Muu+(L)*cos(B(i))));

        Na(i)=(mg*L_2(i)*cos(B(i))+Kapa*Muu)/((L)*cos(B(i)));
```

```
        d2y_dt2(i)=sin(B(i))*d2A_dt2(i);
        d2x_dt2(i)=0.2+cos(B(i))*d2A_dt2(i);
        Tm2(i)=(m*d2x_dt2(i)+Muu*N_c(i)-0.01*Na(i))/(G*(Eff/r)*(cos(B(i))));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end


    w_max(i)=Tm2(i)*w_slope+w_noload;
    if(w(i)<(w_max(i)-100))
        d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
        w(i+1)=d2Q_dt2(i+1)*ti+w(i);
    else
        w(i+1)=w_max(i);
        d2Q_dt2(i+1)=(w(i+1)-w(i));

    end
    w(i+1)= d2Q_dt2(i)*ti+w(i);

    aaa=aaa+1;
    dB_dt(i)=0;
    d2B_dt2(i)=0;
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
     B(i+1)=30*pi/180;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;

    B(i+1)=E;
    time(i+1)=time(i)+ti;
    i=i+1;

end
i_end(j)=i-1;
j=j+1;
% %%                                                          PART 5
L_2(i)=L_2(i(end)-1);
L_2(i)=L/2;
L_T(i-1)=0.01;
L_T(i)=0.01;
L_1(i)=L/2;

B(i)=30*pi/180;
while(L_2(i-1)>0)
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i))-20)
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
```

```
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));


        end
        d2A_dt2(i+1)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i)=L/2+L_T(i);
        L_2(i)=L/2-L_T(i);
        d2y_dt2(i)=(r/60)* d2Q_dt2(i)-sin(B(i))*d2A_dt2(i);
        d2x_dt2(i)=cos(B(i))*d2A_dt2(i)-dA_dt(i)*sin(B(i));

         Na(i)=mg*(L_2(i))/(L/2*cos(B(i))*(cos(B(i))-
Muu*sin(B(i)))+Kapa*(Muu*cos(B(i))+sin(B(i))));

N_c(i)=mg+m*d2y_dt2(i)-Na(i)*(cos(B(i))-Muu*sin(B(i)))-
(Tm2(i)*(G*Eff/r))*sin(B(i));

                        Tm2(i)=(m*d2x_dt2(i)+Muu*N_c(i)+Muu*Na(i))/(G*(Eff/r));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    d2B_dt2(i)=0;
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
     B(i+1)=30*pi/180;
    dx_dt(i+1)=d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;


    i=i+1;
end
i_end(j)=i-1;
j=j+1;
%
%
%%                                                              part6

BB3 = B(i(end)-1);
BB4 = B(i(end));
K =0;
while(K<BB4)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
```

```
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)-100))
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));

        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)=0.2+ d2Q_dt2(i)*cos(B(i))*r/60-w(i)*r*sin(B(i))/60;
        dy_dt(i)= r*cos(B(i))*w(i)/60+w(i)*r*cos(B(i))/60;
        d2y_dt2(i)= d2Q_dt2(i)*sin(B(i))*r/60+w(i)*r*cos(B(i))/60;
        DeltaL(i)= 0.5*d2x_dt2(i)*time(i)^2+dx_dt(i)*time(i);

        N_c(i)=(mg-m*d2y_dt2(i)-Tm1(i)*(G*Eff/r*sin(B(i))));

        d2B_dt2(i)= ( (G*Eff/r)*Tm1(i)*( sin(B(i))*DeltaL(i)-cos(B(i))*( (
h/cos(B(i)) ) + DeltaL(i)*tan(B(i)) ) )...
            +N_c(i)*( DeltaL(i) - Muu*( ( h/cos(B(i)) )+DeltaL(i)*tan(B(i)) )
) )/I;

        Na(i)=0;

        Tm2(i)=(N_c(i)*Muu+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= B(i)- dB_dt(i)*ti - 0.5*d2B_dt2(i)*ti^2;
        dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
        X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
        dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
        y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;

        time(i+1)=time(i)+ti;
        BB4=B(i(end));
    end
        d2A_dt2(i)=0;
    dA_dt(i)=0;
    i=i+1;
end

i_end(j)=i-1;
j=j+1;
%%                      part 7,8

RT = i+100;
while(i<RT)


    Tm2(i)=(1/w_slope)*w(i)+T_stall;
```

```
    w_max(i)= Tm2(i)*w_slope+w_noload;
    w(i+1)= w_max(i);
    d2Q_dt2(i+1)=(w(i+1)-w(i));

    Na(i)=68;
    N_c(i) = mg-Na(i);
    d2B_dt2(i) =0 ;
    d2y_dt2(i) =0;
    d2x_dt2(i) = (Tm2(i)*(G*Eff/r)-Muu*mg)/m;

    Tm1(i)=Tm2(i);
    dB_dt(i+1)= 0;
    B(i+1)=B(i);
    dx_dt(i+1)=d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=0;
    y(i+1)=0;

    time(i+1)=time(i)+ti;
    L_2(i)=L/2;
    L_T(i)=0.01;
    L_1(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;
    i=i+1;

end
i_end(j)=i-1;
j=j+1;
%                                                          PART 9
while(B(i)>(-34.34*pi/180))
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));

        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)= d2Q_dt2(i)*cos(B(i))*r/60;
        dy_dt(i)= r*cos(B(i))*w(i)/60;
        d2y_dt2(i)= d2Q_dt2(i)*sin(B(i))*r/60+w(i)*r*cos(B(i))/60;
        DeltaL(i)= 0.5*d2x_dt2(i)*time(i)^2+dx_dt(i)*time(i);

        N_c(i)=(mg+m*d2y_dt2(i)+Tm1(i)*G*Eff/r*sin(B(i)));

        d2B_dt2(i)= ( (G*Eff/r)*Tm1(i)*(-cos(B(i)) *DeltaL(i)+sin(B(i))*( (
h/cos(B(i)) ) + DeltaL(i)*tan(B(i)) ) )...
```

```
            +N_c(i)*( DeltaL(i) +Muu*( ( h/cos(B(i)) )+DeltaL(i)*tan(B(i)) )
) )/I;

        Na(i)=0;

        Tm2(i)=(N_c(i)*0.3*cos(B(i))+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= (B(i)- dB_dt(i)*ti - 0.5*d2B_dt2(i)*ti^2);
        dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
        X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
        dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
        y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
        time(i+1)=time(i)+ti;

    end
    L_T(i)=0;
    L_1(i)=L/2;
    L_2(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;
    i=i+1;
end

ff= -30*pi/180;
i_end(j)=i-1;
j=j+1;
% % %                                                          PART 10
while(B(i)<ff)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));
        end

        Rr3(i)=((h)^2+((Kapa/cos(B(i)))-Lc)^2)^0.5;

Na(i)= -mg*(Lc*Muu*cos(B(i))-h*(sin(B(i)))-h*0.5-
Muu*Kapa)/(Kapa*((1/tan(B(i)))*(+Muu*sin(B(i))+cos(B(i)))+(sin(B(i))+Muu*cos(
B(i))))) ;

d2B_dt2(i)=-(((-Na(i)*Kapa)*((1/tan(B(i)))*(Muu*sin(B(i))+cos(B(i)))+(-
sin(B(i))+Muu*cos(B(i))))+(Lc-h*sin(B(i))-(Muu)*mg)/(I+m*(Rr3(i))^2));
```

```
        d2x_dt2(i) =-(r/60)*
d2Q_dt2(i)+(d2B_dt2(i))*((Lc).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =-d2B_dt2(i)*(Lc*cos(B(i))-h*sin(B(i)));

        N_c(i) = (m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(B(i))-
Muu*Na(i)*cos(B(i))+Na(i)*sin(B(i)))/Muu;

        Tm2(i) = (d2y_dt2(i)*m-
mg+Na(i)*cos(B(i))+N_c(i))/((sin(B(i))*G*Eff)/r);
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;
    w(i+1)=d2Q_dt2(i)*ti+w(i);
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;
    i=i+1;

end
i_end(j)=i-1;
j=j+1;
%                                                          PART 11

B(i)=-30*pi/180;
L_1(i)=L/2;
L_2(i)=L/2;
L_T(i-1)=0.01;
 while(L_1(i)>0)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
            check(i)=1;
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));

        end
        d2A_dt2(i+1)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i+1)=L/2-L_T(i);
        L_2(i+1)=L/2+L_T(i);
```

```
        d2y_dt2(i)=sin(B(i))*d2A_dt2(i);
        d2x_dt2(i)=cos(B(i))*d2A_dt2(i)-dA_dt(i)*sin(B(i));

%
        Na(i)=mg*L_1(i)*cos(B(i))/(((L/2)*cos(B(i)))*(+Muu*cos(B(i))-
sin(B(i)))-Kapa*(-Muu*cos(B(i))+sin(B(i))));
        N_c(i)=mg+d2y_dt2(i)+Tm2(i)*(G*(Eff/r)*sin(B(i)))-
Na(i)*(Muu*cos(B(i))+sin(B(i)))/(Muu*cos(B(i))+sin(B(i)));

        Tm2(i)=(-m*d2x_dt2(i)+(+Muu*cos(B(i))-
sin(B(i)))*N_c(i)+(Muu*cos(B(i))-sin(B(i)))*Na(i))/(G*(Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    d2B_dt2(i+1)=0;
    dB_dt(i+1)=0;
    B(i+1)=-30*pi/180;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    i=i+1;
 end
i_end(j)=i-1;
j=j+1;
% %                                                          PART 12

while(B(i)>(-34.34*pi/180))
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i))-200)
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));

        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)= d2Q_dt2(i)*cos(B(i))*r/60-w(i)*r*sin(B(i))/60;
        dy_dt(i)= r*cos(B(i))*w(i)/60+w(i)*r*cos(B(i))/60;
        d2y_dt2(i)= d2Q_dt2(i)*sin(B(i))*r/60+w(i)*r*cos(B(i))/60;
        DeltaL(i)= 0.5*d2x_dt2(i)*time(i)^2+dx_dt(i)*time(i);

        N_c(i)=(mg-m*d2y_dt2(i)+Tm1(i)*G*Eff/r*sin(B(i)));
```

```
        d2B_dt2(i)= ( (G*Eff/r)*Tm1(i)*( sin(B(i))*DeltaL(i)-cos(B(i))*( (
h/cos(B(i)) ) + DeltaL(i)*tan(B(i)) ) )...
            +N_c(i)*( DeltaL(i) + Muu*( ( h/cos(B(i)) )+DeltaL(i)*tan(B(i)) )
) )/I;

        Na(i)=0;

        Tm2(i)=(N_c(i)*Muu+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= (B(i)- dB_dt(i)*ti - 0.5*d2B_dt2(i)*ti^2);
        dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
        X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
        dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
        y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
        time(i+1)=time(i)+ti;

    end
    L_T(i)=0;
    L_1(i)=L/2;
    L_2(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;

    i=i+1;
end
fff= -30*pi/180;
i_end(j)=i-1;
j=j+1;
%                                                          PART 13
while(B(i)<fff)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));
        end


Na(i)= -mg*(Lc*Muu*cos(B(i))-h*(sin(B(i)))-h*0.5-
Muu*Kapa)/(Kapa*((1/tan(B(i)))*(+Muu*sin(B(i))+cos(B(i)))+(sin(B(i))+Muu*cos(
B(i))))) ;

d2B_dt2(i)=-(((-Na(i)*Kapa)*((1/tan(B(i)))*(Muu*sin(B(i))+cos(B(i)))+(-
sin(B(i))+Muu*cos(B(i))))+(Lc-h*sin(B(i))-(Muu)*mg)/(I+m*(Rr3(i))^2));
```

```
        d2x_dt2(i) =-(r/60)*
d2Q_dt2(i)+(d2B_dt2(i))*((Lc).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =-d2B_dt2(i)*(Lc*cos(B(i))-h*sin(B(i)));

        N_c(i) = (m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(B(i))-
Muu*Na(i)*cos(B(i))+Na(i)*sin(B(i)))/Muu;

        Tm2(i) = (d2y_dt2(i)*m-
mg+Na(i)*cos(B(i))+N_c(i))/((sin(B(i))*G*Eff)/r);
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;
    w(i+1)=d2Q_dt2(i)*ti+w(i);
    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;
    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;
    i=i+1;

end
i_end(j)=i-1;
j=j+1;

% % % % % %                                                                PART
15

B(i)=-30*pi/180;




L_1(i)=L/2;
L_2(i)=0;
L_T(i-1)=0.01;
 while(L_1(i)>0)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
            check(i)=1;
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));
```

```
        end
        d2A_dt2(i+1)=(r/60)* d2Q_dt2(i);
        dA_dt(i)=r*w(i)/60;
        L_T(i)=L_T(i-1)+dA_dt(i)*ti+0.5*d2A_dt2(i)*ti^2;
        L_1(i+1)=L/2-L_T(i);
        L_2(i+1)=L/2+L_T(i);
        d2y_dt2(i)=sin(B(i))*d2A_dt2(i);
        d2x_dt2(i)=cos(B(i))*d2A_dt2(i)-dA_dt(i)*sin(B(i));
        Na(i)=mg*L_1(i)*cos(B(i))/(((L/2)*cos(B(i)))*(+Muu*cos(B(i))-
sin(B(i)))-Kapa*(-Muu*cos(B(i))+sin(B(i))));
        N_c(i)=mg+d2y_dt2(i)+Tm2(i)*(G*(Eff/r)*sin(B(i)))-
Na(i)*(Muu*cos(B(i))+sin(B(i)))/(Muu*cos(B(i))+sin(B(i)));

        Tm2(i)=(-m*d2x_dt2(i)+(+Muu*cos(B(i))-
sin(B(i)))*N_c(i)+(Muu*cos(B(i))-sin(B(i)))*Na(i))/(G*(Eff/r)*cos(B(i)));
error=(Tm2(i)-Tm1(i))/Tm2(i);
Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    d2B_dt2(i+1)=0;
    dB_dt(i+1)=0;

    dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
    X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
    dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
    y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
      B(i+1)=-30*pi/180;
      i=i+1;
 end
i_end(j)=i-1;
j=j+1;
% % % % %                                                          PART 16
%
while(B(i)>(-34.34*pi/180))
    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));

        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)= d2Q_dt2(i)*cos(B(i))*r/60-w(i)*r*sin(B(i))/60;
```

```
        dy_dt(i)= r*cos(B(i))*w(i)/60+w(i)*r*cos(B(i))/60;
        d2y_dt2(i)= d2Q_dt2(i)*sin(B(i))*r/60+w(i)*r*cos(B(i))/60;
        DeltaL(i)= 0.5*d2x_dt2(i)*time(i)^2+dx_dt(i)*time(i);

        N_c(i)=(mg-m*d2y_dt2(i)+Tm1(i)*G*Eff/r*sin(B(i)));

        d2B_dt2(i)= ( (G*Eff/r)*Tm1(i)*( sin(B(i))*DeltaL(i)-cos(B(i))*( (
h/cos(B(i)) ) + DeltaL(i)*tan(B(i)) ) )...
            +N_c(i)*( DeltaL(i) + Muu*( ( h/cos(B(i)) )+DeltaL(i)*tan(B(i)) )
) )/I;

        Na(i)=0;

        Tm2(i)=(N_c(i)*Muu+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= (B(i)- dB_dt(i)*ti - 0.5*d2B_dt2(i)*ti^2);
        dx_dt(i+1)=-d2x_dt2(i)*ti+dx_dt(i);
        X(i+1)=X(i)+dx_dt(i)*ti+0.5*d2x_dt2(i)*ti^2;
        dy_dt(i+1)=d2y_dt2(i)*ti+dy_dt(i);
        y(i+1)=y(i)+dy_dt(i)*ti+0.5*d2y_dt2(i)*ti^2;
        time(i+1)=time(i)+ti;
%         BB6=B(i(end));
    end
    L_T(i)=0;
    L_1(i)=L/2;
    L_2(i)=L/2;
    d2A_dt2(i)=0;
    dA_dt(i)=0;

    i=i+1;
end
f= -15.16*pi/180;
i_end(j)=i-1;
j=j+1;
% %                                                              PART 17
while(B(i)<f)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i+1)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i+1)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i+1)=(w(i+1)-w(i));
        end

        Rr3(i)=((h)^2+((Kapa/cos(B(i)))-Lc)^2)^0.5;
```

```
Na(i)= -mg*(Lc*Muu*cos(B(i))-h*(sin(B(i)))-h*0.5-
Muu*Kapa)/(Kapa*((1/tan(B(i)))*(+Muu*sin(B(i))+cos(B(i)))+(sin(B(i))+Muu*cos(
B(i))))) %#ok<NOPTS>

d2B_dt2(i)=-(((-Na(i)*Kapa)*((1/tan(B(i)))*(Muu*sin(B(i))+cos(B(i)))+(-
sin(B(i))+Muu*cos(B(i))))+(Lc-h*sin(B(i))-(Muu)*mg)/(I+m*(Rr3(i))^2));
        d2x_dt2(i) =-(r/60)*
d2Q_dt2(i)+(d2B_dt2(i))*((Lc).*sin(B(i))+h*cos(B(i)));
        d2y_dt2(i) =-d2B_dt2(i)*(Lc*cos(B(i))-h*sin(B(i)));

        N_c(i) = (m*(d2x_dt2(i))+(G*Eff/r)*Tm2(i)*cos(B(i))-
0.2*Na(i)*cos(B(i))+Na(i)*sin(B(i)))/Muu;

        Tm2(i) = (d2y_dt2(i)*m-
mg+Na(i)*cos(B(i))+N_c(i))/((sin(B(i))*G*Eff)/r);
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)= Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
    end

    time(i+1)=time(i)+ti;

    w(i+1)=d2Q_dt2(i)*ti+w(i);

    dB_dt(i+1)=d2B_dt2(i)*ti+dB_dt(i);
    B(i+1)=B(i)+dB_dt(i)*ti+0.5*d2B_dt2(i)*ti^2;
    time(i+1)=time(i)+ti;
    i=i+1;

end
Z=0;
i_end(j)=i-1;
j=j+1;
%                                                PART 18
while(B(i)<Z)

    error=1;
    Tm2(i)=(1/w_slope)*w(i)+T_stall;
    Tm1(i)=Tm2(i);
    while((error>0.001)||(error<-0.001))
        w_max(i)=Tm2(i)*w_slope+w_noload;
        if(w(i)<(w_max(i)))
            d2Q_dt2(i)=(a_slope*Tm2(i))+a_noload;
            w(i+1)=d2Q_dt2(i)*ti+w(i);
        else
            w(i+1)=w_max(i);
            d2Q_dt2(i)=(w(i+1)-w(i));
        end
        dx_dt(i)= r*cos(B(i))*w(i)/60;
        d2x_dt2(i)=- d2Q_dt2(i)*cos(B(i))*r/60;
        dy_dt(i)= r*cos(B(i))*w(i)/60+w(i)*r*cos(B(i))/60;
        d2y_dt2(i)= -d2Q_dt2(i)*sin(B(i))*r/60/60;

     N_c(i)=(mg+m*d2y_dt2(i)-Tm1(i)*sin(B(i))*G*Eff/r);
```

```
        d2B_dt2(i)= ((( (G*Eff/r*Tm1(i)))*cos(B(i)*( Lc*cos(B(i))-
h*sin(B(i)))+sin(B(i))*( Lc*sin(B(i))-h*cos(B(i))))+N_c(i)*(( Lc*cos(B(i))-
h*sin(B(i)))-Muu*( Lc*sin(B(i))+h*cos(B(i)))))/I);

        Na(i)=0;

        Tm2(i)=(N_c(i)*Muu+m*d2x_dt2(i))/((G*Eff/r)*cos(B(i)));
        error=(Tm2(i)-Tm1(i))/Tm2(i);
        Tm2(i)=Tm1(i)+Tm2(i)*error/4;
        Tm1(i)=Tm2(i);
        w(i+1)=d2Q_dt2(i)*ti+w(i);
        dB_dt(i+1)= d2B_dt2(i)*ti + dB_dt(i);
        B(i+1)= (B(i)+dB_dt(i)*ti+ 0.5*d2B_dt2(i)*ti^2);
        time(i+1)=time(i)+ti;

    end

    i=i+1;
end
i_end(j)=i-1;

%                               PLOT
j=1;
while(j<18)

    if(j==1)
        i=1:i_end(j);
    elseif(j==17)
        i=1:i_end(16);
    else
        i=(i_end(j-1)+1):i_end(j);
    end


    figure(j)
    subplot(5,2,1)
    plot(time(i),B(i))
    xlabel('t(sec)')
    ylabel('B(rad)')
    if(time(max(i))>time(min(i)))
        xlim([time(min(i)) time(max(i))])
    end
    title('Angle')

    subplot(5,2,2)
    plot(time(i),d2B_dt2(i));
    xlabel('t(sec)')
    ylabel('d2B_dt2(rad/t^2)')
    if(time(max(i))>time(min(i)))
        xlim([time(min(i)) time(max(i))])
    end
    title('Acceleration of Beta')

    subplot(5,2,3)
    plot(time(i), w(i));
```

```
xlabel('t(sec)')
ylabel('w (rad/s)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Motor Speed')

subplot(5,2,4)
plot(time(i), d2x_dt2(i));
xlabel('t(sec)')
ylabel(' d2x_dt2(m/s^2')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Acceleration of X at center of mass')

subplot(5,2,5)
plot(time(i), d2Q_dt2(i));
xlabel('t(sec)')
ylabel('d2Q_dt2 (rad/s^2)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Motor Acceleration')

subplot(5,2,6)
plot(time(i),d2y_dt2(i));
xlabel('t(sec)')
ylabel('d2y_dt2 (m/s^2)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Acceleration of Y at center of mass')

subplot(5,2,7)
plot(time(i), Tm2(i));
xlabel('t(sec)')
ylabel('Tm2 (Nm)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Motor Torque')

subplot(5,2,8)
plot(time(i), Na(i));
xlabel('t(sec)')
ylabel('Na (N)')
if(time(max(i))>time(min(i)))
    xlim([time(min(i)) time(max(i))])
end
title('Normal Force at A')

subplot(5,2,9)
plot(time(i),(Tm2(i).*w(i)));
xlabel('t(sec)')
ylabel('P_T (W)')
```

```
    if(time(max(i))>time(min(i)))
        xlim([time(min(i)) time(max(i))])
    end
    title('Power')

    subplot(5,2,10)
    plot(time(i), N_c(i));
    xlabel('t(sec)')
    ylabel('N_c (N)')
    if(time(max(i))>time(min(i)))
        xlim([time(min(i)) time(max(i))])
    end
    title('Normal Force at c')

j=j+1;
end
```