# **BIRZEIT UNIVERSITY**

# FACULTY OF GRADUATE STUDIES

"Dynamics and Stability of Rover"



By Muhannad M. Abdo

Submitted in partial fulfillment of the requirements for the Master Degree in Scientific Computing from the Faculty of Graduate Studies at Birzeit University

> Supervised by Dr. Hassan A. Shibly

Birzeit, Palestine, June 8, 2006

# "Dynamics and Stability of Rover" By Muhannad M. Abdo

This thesis was successfully defended on June 8, 2006 and approved by:

**Committee Members** 

Signature

1. Dr. Hassan Shibly

2. Dr. Karim Tahboub

3. Dr. Mohammad Saleh

ant Mich

#### Abstract

Today's stability criteria are not capable for providing a precise definition and precaution suited for a mobile robot: traversing on unpredictable surface, exerting manipulation forces and torques, susceptible for variable ground normal forces, or subjecting to attitude orientations. Thus, this research firstly examined the dynamic effects of mobile robot traversing on different surface geometries with variable configurations and attitudes, and it secondly investigated their impact on the normal forces distribution. Finally, it reflected the influences of these factors on the dynamic stability of the rover in order to protect the rover from tumbling. This study presents a new dynamic stability criterion done on a new mechanical structure; quadruped mobile robot equipped with wheels and legs called rover.

The primary contribution of this thesis is exploiting the Denavit-Hartenburg approach for assigning the coordinate frames at link's end-terminals, and then relating between each two adjacent frames by forming homogeneous transformation matrix. Forward kinematics is exploited to relate the end-effectors (four wheels) with base frame (platform). The platform attitudes (Roll, Pitch, and Yaw) are evaluated in relative to proposed universal frame at the center of platform. The coordination between locomotion (wheels' motion) and manipulation (joints' motion) is clearly defined.

In this work, the dynamic equations of motion are driven by using Newton-Euler Recursive Relations. The kinematics of links (velocities and accelerations) are propagated in forward recursion starting from base frame and ending at the four end-effectors, link by link. As well as, the dynamics of links (generalized forces and moments) are propagated in backward recursion starting from four end-effectors frame and ending at base frame, link by link. The force and moment propagated into a base link (platform) are determined as a function of gravity forces, inertial forces, inertial toques exerted on the center of mass of links, and ground normal forces exerted on the end-effectors.

The equations of equilibrium for four legs are considered indeterminate system, thus in this thesis the normal forces are evaluated for three contact legs in the case the nonsymmetric rover. However, in the case of symmetric configurations the normal forces are distributed equally between the sides which sharing the same the inertial forces, ground geometries, and platform attitude. Thus regarding to symmetric four legs are evaluated by considering two legs sharing the same value.

A new dynamic stability criterion is presented for rover in this thesis, and it is operating on various shapes of surfaces, and variable rover configurations. In addition, this criterion provides on-line calculations for the effect of variable rover configurations, various surface geometry, platform attitudes, kinematic values, dynamic effects, and variable ground normal forces. The on-line calculations are referred relatively to the universal frame.

The simulation model is also presented for various examples using MatLab in order to provide on-line calculations for predicting the behavior of a physical system under a variety of surface geometries and rover configurations.

**Keywords:** mobile robots, center of mass, static and dynamic stability margin, forward and inverse kinematics, forward and backward dynamics, wheeled-legged manipulator, uneven terrain, inertial forces and moments, inertial acceleration, normal and frictional forces, Newton-Euler Recursive Relations.

#### ملخص

ما تم إيجاده سابقاً من معايير للثبات الستايتيكي والديناميكي لا يزود بدلائل كافية بخصوص وقاية العربات المتحركة (mobile robots) من خطر الإنقلاب، وخصوصاً في حالات السطوح الغير منتظمة وغير المستوية، المتباينة في قوى رد الفعل المتغيرة على الأرجل الملامسة لتلك السطوح. تتعرض العربة في تلك الحالات لتأثير القوى الديناميكية وما يتبعه من تغيير وضعية دوران العربة. يقدم هذا البحث مجسم ميكانيكي جديد لعربة ذو أربع أرجل مزودة بالعجلات، وتم تصميمها ودراسة تأثيرات هذه العوامل مشتركة على الإتزان الديناميكي.

هذا البحث يستخدم (Denavit and Hartenberg) لتعيين المحاور الثلاثية لأجزاء العربة الداخلية وإستخدام مصفوفة (homogeneous transformation) لإيجاد العلاقة بين المحاور المتجاورة. أما العلاقة المرتبطة ما بين القاعدة (platform) والعجلات (end-effectors) فلقد قمنا بإستخدام (forward kinematic)، كذلك تم إيجاد قيم زوايا دوران القاعدة المجهولة حول محور ثابت.

طريقة Newton-Euler Recursive Relations قد تم توظيفها لإيجاد تأثير ال Kinematics إبتداءاً من القاعدة حتى العجلات الأربع، ومن ثم تم إيجاد تأثير ديناميكية القوى والعزم على كل جزء من أجزاء العربة بداية من العجلات حتى القاعدة. قدم هذا البحث معيار جديد للإتزان الديناميكي للعربة استنادًا على قيمة العزم على سطح القاعدة المعبر عنها نسبة للمحور الثابت على أن تكون بعيدة عن القيمة الحرجة.

ولقد تم إستخدام برنامج ال Matlab للتطبيق على عدة أمثلة تحتوي على عدة متغيرات في شكل تضاريس سطح المريخ والتغيرات الحاصلة على حركة أجزاء العربة، وكذلك ضمن تأثير ديناميكية متغيرة.

## ACKNOWLEDGMENT

I would like firstly to appreciate my supervisor Dr. Hassan Shibly, because he drew my attention toward the Robotic Science. In addition, he granted me the spark to initiate the scientific research from scratch, and he provided me further valuable comments, techniques, and information in order to deeply penetrate the arduous setbacks throughout the years. As well as, I highly appreciate Dr. Hassan for his advices regarding to my future.

I would like also to thank my parents, brothers, and sisters for making the house atmosphere excellent for concentration, and for encouraging me.

Furthermore, I am thankful for my close friend Mr. Wafa Al-Jamal for his encouragement, valuable information and practical experience in mobile robot.

# NOTATIONS

#### Abbreviations

- DH Denavit-Hartenburg
- RCF right conjunctional frame
- LCF left conjunctional frame
- RDF right disjunctional frame
- LDF left disjunctional frame
- RFS right front shoulder
- RRS right rear shoulder
- LFS left front shoulder
- LRS left right shoulder
- K front or rear chosen under a certain conditions
- RKIS right front/rear input system
- LKIS left front/rear input system
- GCP ground contact point

# **Coordinate frames**

- O frame
- O<sub>U</sub> universal frame
- O<sub>0</sub> base frame
- O<sub>0R</sub> right base frame
- O<sub>0L</sub> left base frame

- O<sub>1R</sub> right conjunctional frame
- O<sub>1L</sub> left conjunctional frame
- O<sub>2R</sub> right disjunctional frame
- O<sub>2L</sub> left disjunctional frame
- $O_{3RF}$  locomotive wheel frame of right front leg
- $O_{3RR}$  locomotive wheel frame of right rear leg
- O<sub>3LF</sub> locomotive wheel frame of left front leg
- $O_{3LR}$  locomotive wheel frame of left rear leg
- O<sub>4RF</sub> end-effector frame of right front leg
- $O_{4RR}$  end-effector frame of right rear leg
- $O_{4LF}$  end-effector frame of left front leg
- $O_{4LR}$  end-effector frame of left rear leg
- O<sub>WRF</sub> wheel universal frame of right front leg
- $O_{WRR}\xspace$  wheel universal frame of right rear leg
- $O_{WLF}$  wheel universal frame of left front leg
- O<sub>WLR</sub> wheel universal frame of left rear leg
- O<sub>SRF</sub> surface frame of right front leg
- O<sub>SRR</sub> surface frame of right rear leg
- O<sub>SLF</sub> surface frame of left front leg
- O<sub>SLR</sub> surface frame of left rear leg
- O<sub>G</sub> ground universal frame

## Drawings

- $\oplus$  axis is pointing in the paper
- $\otimes$  axis is pointing out the paper
- indicates that the wheel is in contact with ground.
- indicates that the wheel is not in contact with ground.
- $\mapsto \bullet$  means that the leg is in contact with ground.
- $\mapsto \circ$  means that the leg is not in contact with ground.

## Variables and constant

- A homogeneous transformation
- B generalized homogeneous transformation matrix
- q generalized coordinate
- $\theta$  variable joint
- $\alpha$  twist angle
- n normal vector
- o orientation vector
- a approach vector
- p position vector
- $r_i^{i-1}$  position vector from frame i to frame i-1
- $d_i \qquad \text{link offset of link } i.$
- $a_i$  link length of link i.
- m<sub>i</sub> mass of link i.
- m total mass of the rover

## **TERMINOLOGIES**

**Manipulator robot:** is a set of links connected with joints that executes a set of manipulations via joints and links, while the base link is fixed by stationary pillar.

**Manipulation:** is the movement of robot's components with respect a fixed base frame.

**Mobile robot:** executes a set of manipulations and locomotions during the travel, while the base link moves.

**Locomotion:** is the movement of the base frame with respect the universal frame as resulted of the movement of the locomotive device. This process requires scientific and accurate coordination between base link, robot's components, and the geometry of the ground.

**Base link:** is considered the first device of the four legged manipulators, it is not bolted with stationary pillar as Stanford, Screw, Puma, etc. Therefore, it is influenced by the configurations of the four legs, the geometry of the ground, as well as the generalized forces acting on the end-effectors.

**End-effector link:** is the last link that interacts on the surrounding environment. Its functionality integrates the manipulation and the locomotion using finger, arm, leg or wheel. In this thesis, there are four end-effectors, i.e. four wheels.

**Kinematics:** is concerned with study of motion of robot (i.e. displacements, velocities, and accelerations of links) regardless the forces that cause these motions.

Forward kinematics: is the study of position and orientation of the endeffector as a function of the joint angles, in forward manner starting from base to end-effector.

**Inverse kinematics:** is a study of the joint angles as a function of position and orientation of the end-effector.

**Dynamics:** is concern with study of forces and moments (i.e. normal forces, gravity forces, and inertial forces and moments) that cause the motion without regard to the displacements, velocities, and accelerations.

**Forward dynamics:** is the derivation of kinematics from forces and moments starting from platform and ending at wheel, link by link.

**Inverse dynamics:** is the derivation of forces and moments from the kinematics starting from wheel and ending at platform, link by link.

**Stability criterion:** is a concept or a technique made to prevent the robot from turning over.

**Static stability:** is a study concerns in mobile robot moving with zero or constant velocity (acceleration = zero) in the absence of inertial forces, under the effect of ground geometries, normal forces exerted on end-effectors, and gravity forces exerted on center of masses of links. It discusses the support polygon where the line of gravity will fall inside. Thus, the mechanical system is more stable and comfortable with using more legs.

**Dynamic stability:** is the study that concerns in mobile robot moving with regularly linear velocity (constant acceleration). The additional effects added to static case are the influences of frictional forces, and inertial forces and torques

acting on center of masses of links. This study requires arduous control and numerical computations in order to achieve on-line calculations.

**Center of mass:** is a single point around which the total mass of the rover is balanced in all direction.

**Support polygon:** is the polygon area delimited by the projections of supported legs onto horizontal plane.

Ground contact points: are the numbers of landing legs on the surface.

**Generalized coordinates:** are used to describe the geometric configurations or the degrees of freedom for mechanical system.

**Generalized forces:** are the forces and moments acting by actuator on joints in the direction of the generalized coordinates.

# List of Figures

Figure 1.1. Mars Pathfinder mission settled on Ares Vallis on July 4, 1997 [17]	3
Figure 1.2. Spirit and opportunity robot [19].	4
Figure 1.3. Rover components composed of four wheeled-legged manipulators	7
Figure 1.4. Top view shows the support polygon and pattern onto horizontal plane	. 16
Figure 1.5. Energy Stability Margin	. 17
Figure 2.1. Joint axes assignments and frame numbering for the Rover	. 35
Figure 2.2. x <sub>i</sub> -axis setting up. a. In parallel joint axis, x <sub>i</sub> axis is in the line directed from	1
frame O <sub>i-1</sub> to O <sub>i</sub> . <b>b.</b> In intersecting joint axis, x <sub>i</sub> axis is perpendicular to the plane of	or
parallel to the vector cross product $\pm z_{i-1} \times z_i$ .	. 36
Figure 2.3. Assignments of coordinate frame on the form of home position	. 37
Figure 2.4. Pairs of two adjacent links.	. 39
Figure 2.5. Position vector from base to end-effector frame	. 44
Figure 2.6. Differential gear joint	. 45
Figure 2.7. Two frames attached at the base link.	. 45
Figure 2.8. Transform graph for the four legs, starting from platform frame to end-	
effector frame	. 48
Figure 2.9. The frames for the four legs,	. 48
Figure 2.10. The coordinate frame of manipulated wheel.	. 50
Figure 2.11. Wheel angular movement tracked linearly on ground	. 51
Figure 2.12. Two opposite wheels enabling the rover for rotating forward, the arc length	th
of the wheel is tracked on ground, from start to finish of the travel	. 54
<b>Figure 2.13.</b> Body attitude provides three rotational degrees-of-freedom $(\phi, \theta, \psi)$ ,	
assuming congruent frames for platform and universal frame at the beginning	. 58
<b>Figure 2.14.</b> Roll motions about $y_U$ axis by $\phi$ angle, Pitch motions about $z_U$ axis by $\theta$	
angle, and Yaw motions about $x_U$ axis by $\psi$ angle	. 58
Figure 2.15. Transform graph of universal frame and two bases frames.	. 60
Figure 2.16. The transform graph of rover frames.	. 61
Figure 2.17. The geometric configurations and ground input Systems relative to the tw	0
legged manipulators.	. 63
Figure 2.18. The black circle indicates for supported legs and white circle indicates for	5
not supported legs with the ground.	. 64
Figure 2.19. Pitch angle.	. 65
Figure 2.20. Rotation about lateral axis of universal frame by $\theta$	. 66
Figure 2.21. Elevation difference	. 67

# XIII

Figure 2.22. Front view shows the platform rotating about longitudinal axis of univers	sal
frame by $\phi$ . y <sub>0</sub> -axis and y <sub>U</sub> -axis are pointing out of paper	69
Figure 2.23. Pythagorean relations.	70
<b>Figure 2.24.</b> The altitude of RCF $O_{1R}$ to ground frame $O_{G}$	72
Figure 2.25. Coordinate frames of $O_U$ , $O_{0R}$ , $O_{0L}$ and $O_{1L}$ .	75
<b>Figure 2.26.</b> Rotation about vertical axis of universal frame by $\psi$	78
Figure 2.27. Side view shows the difference in the locations of LC and RC, and top view	iew
shows the rotational vaw angle $\psi$ occurred between the universal and platform	
frame.	79
<b>Figure 2.28. a.</b> Arc path occurs when $a_A \theta_{AB} \neq a_A \theta_{AI}$	80
<b>Figure 2.29.</b> Front view shows the four less open by an angle about $\mathbf{z}_{1P}$	82
<b>Figure 2.30.</b> Platform universal frame wheel universal frame and ground universal	02
frame are contingent frames for being having the same orientations	85
Figure 2.31. Completed transform graph	86
Figure 2.32. End-effector pose	
Figure 2.33. Surface frame	91
Figure 3.1. Dynamics propagations	96
Figure 3.2. Recursive Newton-Euler Formulation notations on the base of the standard	d of
the DH convention	97
Figure 4.1. Normal forces acting on wheels perpendicular to surface.	122
Figure 4.2. External forces and moments exerted by ground on end-effector projected	in
frame O <sub>4</sub>	125
Figure 5.1. Four critical moments	138
Figure 5.2. Platform attitude.	142
Figure 5.3. Normal forces	143
Figure 5.4. Universal moments and critical moments about z <sub>U</sub> axis	145
Figure 5.5. Universal moments and critical moments about y <sub>U</sub> axis.	146
Figure 5.6. Propagated torques about (x <sub>U</sub> ,y <sub>U</sub> ,z <sub>U</sub> ) axes	147
Figure 5.7. Propagated moment of gravity forces about (x <sub>U</sub> ,y <sub>U</sub> ,z <sub>U</sub> ) axes	147
Figure 5.8. Propagated moment of inertial forces about (x <sub>U</sub> ,y <sub>U</sub> ,z <sub>U</sub> ) axes	148
Figure 5.9. Propagated moment of normal forces about (x <sub>U</sub> ,y <sub>U</sub> ,z <sub>U</sub> ) axes	148
Figure 5.10. Rover posture on step flat-inclined surface	149
Figure 5.11. Platform attitude.	150
Figure 5.12. Normal forces.	151
Figure 5.13. Universal moments and critical moments about y <sub>U</sub> axis	152
Figure 5.14. Propagated torques about (x <sub>U</sub> ,y <sub>U</sub> ,z <sub>U</sub> ) axes	153
Figure 5.15. Propagated moment of gravity forces about $(x_U,y_U,z_U)$ axes	153
Figure 5.16. Propagated moment of inertial forces about $(x_U, y_U, z_U)$ axes	154
Figure 5.17. Propagated moment of normal forces about $(x_U,y_U,z_U)$ axes	155

### Figure 5.19. Platform attitude 157 Figure 5.32. Zooming for universal moments and critical moments about z<sub>U</sub> axis ...... 170 Figure A.1. Two adjacent links [65]. Figure E.1. Transform graphs for four legged manipulators starting from universal to end-effectors. 221 Figure E.2. Forces and moments exerted on link 4RF, 4RR, 4LF, and 4LR...... 222 Figure E.3. Forces and moments exerted on link 3RF, 3RR, 3LF, and 3LR...... 223

#### XIV

# List of Tables

<b>Table 2.1.</b> Kinematic parameters table based on DH convention.	
<b>Table 5.1.</b> conf_1 $\rightarrow$ conf_2	
Table 5.2. Torques exerted on wheel.	
<b>Table 5.3.</b> $\operatorname{conf}_0 \to \operatorname{conf}_0$ , $\operatorname{conf}_0 \to \operatorname{conf}_1$ , $\operatorname{conf}_0 \to \operatorname{conf}_2$	
<b>Table 5.4.</b> $\operatorname{conf}_0 \rightarrow \operatorname{conf}_0$ , $\operatorname{conf}_0 \rightarrow \operatorname{conf}_1$ , $\operatorname{conf}_0 \rightarrow \operatorname{conf}_2$	
<b>Table 5.5.</b> $\operatorname{conf}_0 \to \operatorname{conf}_0$ , $\operatorname{conf}_1 \to \operatorname{conf}_1$ , $\operatorname{conf}_2 \to \operatorname{conf}_2$	
<b>Table 5.6.</b> $\operatorname{conf}_0 \to \operatorname{conf}_0$	
<b>Table 5.7.</b> $\operatorname{conf}_0 \to \operatorname{conf}_1$	
Table A.1. DH explanation.	193
Table C.1. Dynamic parameters table.	

# Table of contents

Abstract .		I			
ACKNO	WLEDGMENT	V			
NOTATIONS					
TERMIN	TERMINOLOGIES				
Chapter (	One	1			
1. Introduction					
1.1.	Mechanical structures for mobile robots				
1.1.1. Manipulation system					
1.1.2	9				
1.2.	Stability criterion				
1.2.1	. Static stability margin				
1.2.2	2. Previous work on static stability				
1.2.3	<ol> <li>Dynamic stability margin</li> </ol>				
1.2.4	Previous work on dynamic stability				
1.3.	Computational kinematics and dynamics				
Chapter 7	ſwo				
2. Kine	ematics of the rover				
2.1.	Coordinate frames				
2.2.	Denavit-Hartenburg parameters				
2.3.	Homogeneous transformation:				
2.4.	Forward kinematics				
2.5.	Base frame				
2.6.	Platform frame				
2.7.	Wheel kinematics	49			
2.8.	Platform kinematics	53			
2.9.	Platform universal frame	57			
2.9.1	. Attitude angles				
2.10.	Rover transform graph				
2.10	1. Ground universal frame				
2.10	.2. Universal wheel frame				
2.10	.3. Surface geometries				
Chapter 7	Three				
3. Con	3. Computational Dynamics				
3.1.	3.1. Dynamic equations of motion				
3.2. Output generalized coordinates					
3.3.	Newton-Euler Recursive Relations				

# XVII

	3.3.1.	Forward recursion		
	3.3.2.	Backward recursion	112	
Cha	118			
4.	System f	orces and moments	118	
	4.1.1.	System weight		
	4.1.2.	Normal force		
	4.1.3.	Frictional force	123	
	4.1.4.	Wheeled motor torque	124	
	4.1.5.	Constraints		
Chaj				
5. Results and discussion				
Chapter Six				
6. Conclusion				
References				
Арр	endices		192	
App	endix A: 1	Denavit-Hartenburg Convention	192	
App	endix B: 1	nverse kinematics	196	
Арр	endix C: 1	Kinematic and dynamic parameters	207	
App	endix D:	Newton-Euler Recursive Formulation		
App	endix E: I	Free-Body Diagram for four manipulators	221	
Appendix F: Universal forces and moments			227	
Matlab Code				

# **Chapter One**

### 1. Introduction

Mobile robots play a major role in development of our real life in different areas. In the wake of the growing speed of technologies and explorations, the human beings face challenges to accomplish specific missions in dangerous environments safely in meaningful and purposeful manner. For example, the explorations taken place inside volcanoes [1], nuclear reactors [2], mining fields [3], construction and forestry industries [4], or planetary missions [5, 6]. Those missions confront arduous processes and endanger the workers' life to reach over a forbidden location entries. In fact, those explorations are highly in need of replacement of direct human intervention with accessible mobile machines, which achieve incorporation between the manipulation and locomotion automatically. Therefore, countless efforts [7, 8, 9, 10, 11] have been focused on autonomous mobile robots in order to avoid the human operators from the dangerous environment.

Since 1960 [12], there have been a growing international interests in the Mars exploration where the absence of life assurance. The scientists [13, 14] have interested in Martian surface geology, topology, mineralogy, morphology, geochemistry, and atmospheric environment. Indeed, they have drawn the world's attention for three main necessities throughout exploring those Martian properties and characteristics: in order to make sure of the probability of lastpresent life existence, understand the climate history, and search for what resources can be benefited from over there. However, the indirect contact of scientists with Mars from earth throughout telecommunication systems yields uncompleted results. This reason has enforced the scientists' needs directing toward mobile robot capable for gaining sufficient amount of samples of sands and rocks and subjecting these samples under tests and experiments on the earth. Therefore, the planetary scientists have opened their eyes on the use of small mobile robots since 1996 [15, 16], which are capable for traversing random Martian terrain stably and smoothly for longer traverses and time. This mission requires studies for: firstly, an efficient mechanical structure. Secondly, effective dynamic stability criterion. Thirdly, mathematical analysis and simulation for kinematics and dynamics in computational manner. Fourthly, surface geometry and its dynamic disturbances.

### 1.1. Mechanical structures for mobile robots

The first micro-rover, named Sojourner [14], was launched aboard the Mars Pathfinder spacecraft in 1996 and landed on Mars in July 4, 1997 [17], see Figure 1.1. However, the Sojourner was designed for a very limited mission distance and time; it traversed 100 meters as a total distance during its elapsed time "83 sol" over there, while the average speed was 2.7 meters per traverse day [17, 18].



a. Lander and rover.



**b.** Sojourner rover roamed on the Martian surface.

Figure 1.1. Mars Pathfinder mission settled on Ares Vallis on July 4, 1997 [17].

In 2003, NASA's Mars Exploration Rover mission sent two identical six-wheeled mobile robots; the first was named Spirit and the second was named Opportunity. The Spirit and Opportunity landed on opposite sides on the surface of Mars and completed the mission in January 2004 [16]. They both can move on terrain with five centimeters per second as top speed, and can traverse 40 meters in Martian daytime, and the mission life was no more than 90 days and 1000 meter as total distance [19]. Robotic arm was attached on platform for testing Martian rocks and soil as shown Figure 1.2.



Figure 1.2. Spirit and opportunity robot [19].

However, the previous traditional rovers have maintained stable in short traverses and time, with slow and constant velocity, and in relatively benign terrain [20] due to their inheritance of same mechanical characteristics, ignoring rover kinematics and dynamics, neglecting the idea of existing of inertial effects and unpredictable environments, imposing quasi-static motion, and disability to define precise static and dynamic stability criterion capable for functioning in all rover mechanical structures and surface geometries. In future Mars exploration missions, there will highly be interests for autonomous mobile robots that will broaden the range of exploration for long distance and time in challenging terrain and obstacles more than encountered by previous rovers [14, 21, 22].

Therefore, this work evades the idea of adopting any one of the past mechanical characteristics, and it started from scratch in creating a new mechanical model composed of four manipulator wheeled-legs sharing the same platform as moving base link. The presented rover should maintain statically and dynamically stable during the locomotion to accomplish Mars mission. This is also the main issue in which this work treated and focused in computational manner.

This work exhibits a new mechanical design for a quadruped mobile robot. The four identical wheeled legs are gaining high level coordination between manipulation and locomotion in various aspects, because the four legs share symmetric mechanism and coordinate frames. The design here executes a set of manipulations and locomotions integrated at the same time in algorithmatic control for providing the dynamic stability. This feature contributes in increasing the rover speed stably and smoothly on uneven terrain.

#### 1.1.1. Manipulation system

The rover is simply composed of common platform connected with four wheeled-legged manipulators by differential joint. Notations are distributed on right side, left side, front side, and rear side. Each wheeled-legged manipulator connected with common platform will be represented as right front leg, right rear leg, left front leg, and left rear leg, as shown in Figure 1.3. Each leg is considered as a combination of five links and four joints, starting from platform base link 0, and ended with end-effector link 4. The right side and left side share the differential joint, joint 1, mounted above the mobile platform. At the edge of platform in each side, each two legs share the joint 2 and it is named conjunctional joint. Joint 3 divides those for two independent legs, i.e. front shoulder and rear shoulder. Finally, Joint 4 connects the locomotive wheel. The revolute joints are utilized here for controlling the mobility and posture of the rover. Furthermore, the joints enable the end-effectors to select the footholds on ground, control the distributions of ground normal forces, and delimit the area of support polygon.



Figure 1.3. Rover components composed of four wheeled-legged manipulators.

The first joint, differential joint, rotates around the lateral axis of the universal axis. The second joint, conjunctional joint, rotates around the longitudinal axis of the platform edge. The third joint, disjunctional joint, rotates around the lateral axis of the platform edge. Finally, a wheel is connected by the fourth joint to provide protection from tipping onto its side and to propel the entire rover on ground. Each wheel is equipped with DC motor for the actuating motion. The rover has no breaking system, but the motors provide the feature of selflocking system; so that if the motors of the four wheels are locked, the rover will stop.

The number of degrees of freedom of each leg depends on the number of joints in the rover. Usually in robotics science, each joint provides one degree of freedom either for revolute or prismatic motion, unlike human joints. Moreover, the platform will be susceptible to a sequence of changes in configurations during the motion, while this mobility of platform will provide the system with three degrees of freedom, i.e. three  $(\phi, \theta, \psi)$  related to orientation of the platform represented in roll, pitch, and yaw.

The rover overall weighs was chosen to be 12kg, which is distributed such that the platform weighs approximately 4kg,  $m_1$ = 1kg,  $m_2$ = 0 (by approximation),  $m_3$ = 1kg, and  $m_4$  (wheel) weighs= 0.5kg. The length and the width of the platform respectively are 60cm and 40cm. The length of the each shoulder (link 3) is 40cm. The inner and outer radii of each wheel are 3 cm and 5cm, respectively.

#### 1.1.2. Locomotion system

The locomotion of mobile robot is defined as the movement of the whole robot on the ground by employing either wheels or/and legs. Most mobile robots use the wheels, which are easier to control and manoeuvre, maintain stable, consume less energy, and move faster than legs on an even terrain. However, the wheels cannot operate on uneven terrain efficiently, because the wheels diameters have to be larger than the obstacles to overcome and the rolling contact of such rovers on uneven terrain are susceptible to complex wheeled-ground interactions [23] with the physical soil properties: rocks distribution, friction characteristics and soft terrain. In addition, the heavy-wheels or their payload may plow the soft terrain causing friction forces and terrain damage thwarting the whole mission. Look at the practical prototypes as in SOLERO [24], and CEDRA [25].

In contrast, the legs are capable to select footholds above discontinuous ground, in which benefits the locomotion to traverse on an uneven terrain, that comprises the capability of avoiding the obstacles and holes, walking up and down the steps, overcoming the soft ground sinking and causing less terrain damages, and controlling the distribution of forces. In addition, the positive advantage of legs can add that they are omnidirectional, as it can provide for directional movements forward, backward, sideways, or turn on the spot as shown through the quadruped robot WARP1 [26]. However, legged mobiles have many degrees of freedom that make it difficult to design and control. Moreover, they are relatively slow speed and energy inefficient. In addition, at least six legs are required for static walking, while three wheels are required for static rolling. The practical examples on this type are Quadruped Aibo ERS-210 robot [27], WARP1 [26], TITAN VIII [28] or SILO6 [3].

Therefore, the mobile robot will be much more productive if it is equipped with legs and wheels to over come the most challenges mentioned previously. These wheeled-legged properties mentioned above were implied from practical experiences taken place in several mobile robots; for example in the case of Sojourner, Spirit, Opportunity, or Rocky 7 [29].

This thesis inherits the advantages and eliminates the drawbacks of both legged and wheeled locomotion in computational manner, for being equipped with four wheeled-legged manipulators. Thus, the platform a base link will smoothly rotate in relative to configurations of four wheeled-legged manipulators and surface geometries. In addition, the presented rover will overcome obstacles, traverse uneven terrain at higher velocity in stable form and with less power consumption. Furthermore, it will provide a reliable passive mechanism for supporting the weight of the rover at inclined surface. In addition, it can locomot forward, backward, and sideways.

This rover executes a set of manipulations and locomotions during the travel. The supported legs of rover will be susceptible for discrete changes when the legs are lifted or placed on variable surface geometries. This yield a change on rover attitude with respect to universal frame, because the body's attitude is influenced by configurations of joints and surface geometries subjected on the supported wheeled. These kinds of control, irrespective of manipulation or locomotion, are required a computational stability measured criterion that maintains the rover stable with different terrain types. However, till now there is no precise static and dynamic criterion that can be common for all different mechnical structures and surface geometries. The loss of stability may lead to tipping over and then the mission will fail completely.

#### 1.2. Stability criterion

Generally, stability is defined as the tendency of a robot to return to its original equilibrium state after being influenced by a disturbance. This work studies the stability of rover on the surface of Mars throughout overcoming any perturbation that could enforce the rover to turn over. Many scientific authors have dealt with specific definition which states that the turnover occurs when the center of mass of rover undergoes a rotation about one of its edges of support polygon. This rotation yields a reduction in the number of ground contact points and a decrease in the boundary of the support polygon. The remaining contact points will finally lie on a single line as axis of rotation. Moreover, the moment acting around this single edge of support polygon could enforce the rover to tumble making the system statically unstable. These sentences have been formulated mathematically in order to relate the geometrical shape of the ground directly with the manipulation and locomotion of the rover.

There are two general classifications for rover stability; namely static stability and dynamic stability. In 1968, McGhee and Frank were the first who put forward the static stability criterion for an ideal machine moving at slow and constant speed on even terrain. In 1976, Orin et. al were the first who proposed dynamic stability criterion on the presence of inertial forces. Later, several researchers have either extended the previous criteria or proposed new stability criteria for both static and dynamic. Unfortunately, different applications may require different stability margin criteria. Even if the stability criterion is better evaluated, the mechanism of rover will be optimized in order to cope with different terrain situations [30]. Thus, the criteria founded before were insufficient to remain the most rovers upright or stable [31]. In any way, it should be necessary to pay attention for the definitions for both static and dynamic systems, and the previous criteria done in previous works.

#### 1.2.1. Static stability margin

The static stability was traditionally determined by the support polygon and the projection of the center of mass. These two parameters can formulate a simple definition for static stability: "occurs when center of mass is above the support polygon regardless of the effect of inertial and normal forces". However, this requires computational control for the legged configuration, ground elevations, ground contact points, and the body attitude. The legged configuration [26] studies the sequence and time in which the legs are lifted and placed in ground and in which joint angles are manipulated. The ground contact points, which delimit the support polygon, are chosen by the landing legs on ground. The ground elevation is the input system at each supported wheel.

Conditionally, the minimum requirements demanded for static stability are three legs on contact with the ground, forming the support polygon at all the times. The static stability requirements must enable the vertical projection of the center of mass to be inside the boundary of the support polygon. Otherwise, there will be moment acting around an edge of support pattern that could enforce the rover to tumble, making the system statically unstable.

Tricycle has three contact points on the ground, and the boundary of support polygon is delimited in a triangle area connecting the three contact points. If the vertical projection of center of mass is fallen inside the boundary of the support polygon, then the tricycle is characterized statically stable for keeping itself upright. In contrast, Bicycle has two contact points on the ground, and the boundary of support polygon is restricted in a single line connecting the two contact points, and the center of mass is either above or outside the line. Thus, the bicycle is always characterized statically unstable and cannot keep itself upright at rest or constant speed.

The requirements for static stability mentioned above have been formulated since 1968 in different theorems and for variant mechanisms by several researchers. They have provided an indication for the probability of better static stability by keeping the vertical projection of the center of mass at the middle of support polygon. So that, they have designed the mobile robots with big boundary of support polygon and low height of the center of mass.

#### 1.2.2. Previous work on static stability

McGhee and Frank [32] were the first who introduced the idea of Static Stability Margin criterion, based on an ideal insect locomotion system. They defined it as the shortest horizontal distance from the vertical projection of the center of gravity to the nearest border of the support pattern formed by the contact points of legs with ground, called horizontal support polygon. If the ideal machine is statically stable, the margin will be positive. Otherwise, it will be negative. As shown in Figure 1.4, the black circle indicates for supported leg and white circle indicates for not contacted legs with the ground. The static stability margin of a. is positive, and of b. negative.



Figure 1.4. Top view shows the support polygon and pattern onto horizontal plane.

However, McGhee et al dealt with a rigid body with mass-less legs moving in a straight line, on an even terrain, and in steady-state constant speed locomotion. In addition, this criterion is geometric and independent of the height of center of mass. Moreover, it does not **encompass** kinematic configurations, dynamic effects or normal forces [30].

Messuri and Klein [33] proposed Energy Stability Margin for rough terrain, which evaluates the minimum potential energy or work needed for turning the center of mass of the mobile robot around the edge of the support polygon. In other words, during the rotation of the center of mass on a circular path around the edge, this criterion measures the vertical distance between the maximum height of center of mass at a critical point above the edge and the current height of center of mass multiplying with the weight of the mobile robot as shown in Figure 1.5.



Figure 1.5. Energy Stability Margin

Nagy et al [34] extended the Energy Stability criterion to Compliant Energy Stability Margin to overcome the foot sinkage on compliant terrain. However, the stability margin, which takes into energy consideration, is an inaccurate measure because it changes with respect to the weight of mobile robot at the same posture, i.e. it maximizes the probability of stability for the heavier robot at same posture. Hirose, et al [35][36] eliminated the effect of the weight making the margin in dimensional-length expression by normalizing the Energy Stability Margin to the weight of mobile robot.

However, the static stability does not deal with conditions when the rover is subjected to the inertial forces and moments and ground normal forces [31].
Thus, the static stability can prove its functionality in the case of mass-less legs with imposing limitations on rover's motion by keeping it moves at slow and constant speed to resist the inertial effect [37]. When the moving mobile robot possesses considerable mass legs, the stability must be defined in the dynamic approach. The current efforts of researchers have concentrated on the confrontation of these dynamic effects that can restrict the stability of mobile robots and mission performance during the motion on the base of dynamic stability principle.

#### 1.2.3. Dynamic stability margin

The mobile robot must meet the conditions for dynamic stability throughout accelerated motion with taking into consideration the high effects of the inertial forces and moments, dynamics disturbances from irregular terrain, and variable normal and frictional forces. The dynamically stable rover is considered faster than in the case of statically stable form. Support polygon, legged configurations, center of mass projection, inertial forces and moments, accelerated motion, frictional forces, and normal forces were traditionally considered the main parameters for dynamic stability. As noted previously, the dynamic stability provides more comprehensive definition as if the static study is a part of dynamics. However, the rover may be dynamically stable without being statically stable or vice versa, i.e. the moving bicycle is dynamically stable, since it easy to remain upright and hard to flip during accelerated motion; and it is statically unstable in the roll direction, since it cannot remain upright at rest or slow motions.

#### 1.2.4. Previous work on dynamic stability

Orin et al [38] provided the first dynamic stability margin called Center of Pressure for a six-legged robot vehicle as an extension for center of mass projection idea. This criterion states that a mobile robot is dynamically stable if the projection of the center of mass along the direction of the resultant force remains inside the boundaries of the support polygon.

Vukobratovic and his colleagues [39] proposed Zero Moment Point criterion, which is helpful for biped locomotion only on an even and flat terrain. In any way, this criterion claims the dynamically stability for the rover if the ZMP remains inside the boundary of the support polygon. Zero Moment Mass relies on the concept that states the sum of all forces and the sum of all moments of the rover body on the support polygon are equal to zero. Kang and his colleagues [40] proposed Effective Mass Center based on Zero Moment Mass for a quadruped-walking robot subjected to external forces. They have claimed that the effect of external forces on the real center of mass yields deviation of ZMP from the real center of mass called effective mass center. For finding the walking robot stability, this deviated point can be considered as the real center of mass as if there are no external forces. Thus, the dynamic stability of the quadruped robot can be conventionally found if this point is located inside the support polygon. They attached force sensors to each leg's tip of the quadruped-walking robot in order to find the reaction forces then directly in mathematical equation they substituted these values to evaluate this deviation. However, this criterion is invalid in uneven terrain [31, 23].

Lin and Song [41] proposed Dynamic Stability Margin, which is defined as the smallest of resultant moments around edges of axes of rotation, due to normal forces, gravitational forces, and inertial forces and moment acting on center of mass, normalized by the total weight of the system. The positive moment explicitly counteracts the occurrence of instability otherwise the rover will turn over. However as a result of normalizing the moment by weight; this criterion implicitly conducts the unit of length that should remain positive.

Yoneda and Hirose [42, 23] proposed Tumble Stability Criterion for mass-less legs. This criterion investigates for the mobile robot's stability when all legs become without contact with the ground except two legs forming two ground contact points as a line segment, and the mobile robot will start to tumble by rotating around the single line segment. They claimed that there must be a supporting force for any non-contact point capable for overcoming the tumbling. Furthermore, it evaluates the absolute value of the moment around the rotation axis divided by its weight, which generates around the line segment to withstand the tumbling. This stability criterion is evaluated not only on ground surface, but also on wall and ceiling surfaces in which these surfaces will provide the support forces for legs of the mobile robot. However, it does not take into the consideration of dynamic effects of legged motions when the legs are considerable masses.

Zhou [43] proposed Leg-End Supporting Moment, which is defined similar as previous criteria as the leg-end supporting moment divided by the weight of the mobile robot. If the moment is greater than zero, the mobile robot will remain in stable state. They used the internal robot sensors for finding forces and distances online to have precise measurements.

Papadopoulos and Rey [44, 45] proposed Force-Angle stability measure, which evaluates minimum angle between the net force vector acting on the center of mass and each of the tipover axis normals. The mobile robot is in critical stability when this angle approaches to zero. The zero angle takes place at the time the net force becomes coplanar with any tipover axis normals, or when the net force becomes zero. This criterion shows that the mobile robot's instability takes place if the net force vector directs outside any one of tipover axis normals. Therefore, this criterion takes geometric measure into consideration, and it is sensitive to the effect of center of mass height, whereas the raising of the center of mass height will minimize the probability of keeping the mobile robot in stable situation. Furthermore, they claimed that it operates on uneven terrain because the support pattern, formed by ground contact points, is not restricted in a horizontal plane. However, Garcia [46, 47] proved throughout experiments that this criterion has poor accuracy when manipulation effects arise during walking over an uneven terrain.

Ghasempoor and Sepheri [48] proposed Dynamic Energy Stability Margin. They take into consideration the dynamic effects to the Energy Stability Margin including the inertial and normal forces that encountered during the motion of the mobile robot on rugged terrain.

Garcia and Gonzalez [47] improved the Energy stability Margin to Normalized Dynamic Energy Stability Margin for walking machines. This criterion is defined as the smallest of the stability levels required to tumble the robot around the support polygon, normalized to the robot weight. Furthermore, it shows that the walking machines can remain dynamically stable during motions under dynamic effects if each momentum around its edge of support polygon, generated from robot-ground forces and moments, is positive or in the clockwise direction. It is considered the optimal accuracy from the energy point of view.

However, the stability conditions mentioned above are not adequate to guarantee the safty for whole mobile robots from turnover. If optimum criterion is defined, the robot manipulation and locomotion can also be optimized. Beside, random surface types can be faced and it should be aware of variable normal forces that can suddenly appear and effect on the rover stability, because dynamic disturbances at the wheels generate large moment about the platform link expressed in universal frame, tending to rotate the mobile robot and losing its stability. The net moment that is capable for rotating the rover, which is resulted from the normal forces acted at wheels, gravity forces, inertial forces and torques exerted on the center of mass of each link, must be decomposed, studied in on-line approach, and defined as threshold limits. This requires on-line simulation for the changing occurred in rover kinematics, dynamics, configurations, and attitude. Thus, this thesis exhibits a new stability measure criterion that is sufficient for mobile robots in different surface geometries and configurations: "If the universal moment equals the critical moments, the rover will undergo to angular motion and lose its stability". The critical moment is the required moment to lose one side's connections with ground and rotate the rover about the opposite side. The stability measure criterion will be evaluated under the dynamic stability consideration for new-manufactured prototype composed of four wheeledlegged manipulators and can be generalized and common used for whole mechanical structured.

#### 1.3. Computational kinematics and dynamics

The rigid multibody system only consists of rigid links connected by actuators. To analyze and simulate the kinematics and dynamics of this system, it is necessary to study the relative motion, torques, and forces between the links. During the past researches on dynamics, the robotic system that consists of relatively small numbers of joints was analyzed using graphical and hand calculations. However, the mobile robot that consists of large number of joints and carries variable load will negatively effect on the joint motions, in such a way, the joint's speed either decreases or increases along a planned path. The dynamic characteristics for the manipulators are highly nonlinear system with respect to the number of links. It is highly recommended to make the calculation on-line, therefore it is required driving all its joints accurately and frequently at a sampling frequency higher than 60 Hz [49] for the Stanford arm [50], because the resonant frequency of most of the mechanical manipulators are around 10 Hz [49, 51, 52]. The advent of high-speed computers and computational methods has made it possible to analyze complex dynamic systems.

The computation proves itself efficiently when the amount of computation increases linearly with respect to the number of links, and the sampling frequency is higher than 60 Hz.

The joint space equations of motion can be driven via different approaches; i.e. Free Body Diagram, Lagrange equation, D'Alembert principle, Newton-Euler formulation, Hamilton principle, Gibbs Appell formulation and so on. The Free Body approach [53] is easiest approach for no more than two links. It draws a free body diagram of a certain manipulator including: all external forces by environment, weight exerted by the earth as attraction on the center of gravity of the body, ground reactions on supports, as well as the contact forces exerted by attached bodies on connections. However, the computation for equations of motion will be arduous process for manipulator with three or more links by using the Free Body diagram, because each link must be described to its preceding link successively while the entire system of free bodies is described in the frame work of "inertial coordinates" [49]. Therefore, the scientific researchers have focused the attention in development of advanced approach capable for treating the daily development of robotic mechanism and the increase of the number of links. The Lagrangian approach

[54] is an energy based formulation, since the equations of motion are firstly obtained by finding the kinetic and potential energies of the system, and then substituting these two results in Lagrange's equation (L = T - V). The Recursive Newton-Euler [55] is deal with kinematics and dynamics properties, since the equations of motion are firstly obtained by propagating the velocities and accelerations in forward recursion, and then propagating the input generalized forces in backward recursion.

The equations of motion of robotic manipulator are typically computed via applying either the Lagrange or the recursive Newton-Euler formulation. So a lot of researches have extended new versions for the both approaches. The comparison between two approaches can be inspired from computational complexity, execution time, symbolic simplicity, numeric manner, and accurate result. The Lagrange approach firstly consumed long execution time with complexity  $O(n^4)$  caused by Coriolis and Centrifugal force. Thus, the approximation was an improvement technique by ignoring Coriolis and Centrifugal forces and making the complexity reduced to  $O(n^3)$  caused by acceleration term. Armstrong [56] put forward the role of recursion in the

complexity reduction to O(n), and then a lot of researches have been relied on him.

In 1965, Uicker [54] was the first who introduced the Lagrange equations with high complexity  $O(n^4)$ . Then in 1969, Kahn [57] extended it for spatial open chain system using 4×4 homogeneous transformations with computational complexity  $O(n^3)$ . After that in 1976, Stepanenko and Vukobratovic [55] introduced the Newton-Euler equations for spatial open chains where each component is referred to base inertial frame with complexity  $O(n^3)$ . In addition in 1980, Luh, Walker and Paul [51] reduced the complexity of the Newton-Euler Method to O(n) by using recursive formulation, and considering each link's dynamic referenced to its own link coordinates or local coordinate system using  $3 \times 3$ homogeneous transformation. In 1980, Hollerback [58] extended the Kahn's effort and succeeded in reducing the complexity of Lagrange's approach from  $O(n^3)$  to O(n) by using recursive formulation in Lagrang. However, Silver [59] in 1982 proved that there is no difference between what were developed in these two approaches, Recursive Newton-Euler formulation and Lagrange approach.

The Lagrangian equations are considered most explicit for formulating the equations of motion symbolically, whilst the Newton-Euler is considered most efficient in formulating the dynamic equations numerically and computationally [60]. For example in order to compute all input generalized forces, the authors in [49] reduces the average execution time from 7.9 second via Lagrange approach to 0.0335 second via Newton Euler Recursive formulation using the same program and manipulators (FORTRAN program, and a Stanford manipulator arm using six joints, seven links and a gripper). However, the both approaches cannot be implemented practically on-line, since the sample frequency is less than 60 Hz, until the author rewrote the entire algorithm in assembly language and he reduced the time to 4.5 millisecond, therefore this execution time enable recursive Euler-Newton formulation to be applied online.

Walker and Orin [61, 62] extended the work of Luh *et al* and made application of the recursive Newton-Euler formulation explicit with less execution time. They formulated the equations of motion in explicit form in comparison with others; simply it will yield a set of recursive equations, which can be applied to the links sequentially to compute the generalized forces referenced in their own coordinates in a short period of time or in on-line control. This is the approach which thesis will recommend and base in calculations for computing the equation of motion.

In this work, the equations of motion are driven by using Newton-Euler Recursive Formulations. The kinematics of links (velocities and accelerations) are propagated in forward recursion started from base frame and ending at the four end-effectors, link by link. As well as, the dynamics of links (generalized forces and moments) are propagated in backward recursion started from four end-effectors frame and ending at base frame, link by link. The rover base is a driven link, and it moves as a result on the configurations of the four manipulators and ground elevation. However, The Newton-Euler Recursive Formulations were formulated and applied for various fixed robotic manufactures as Puma 560, Elbow, and Standford manipulators; where the main platform of the pervious systems is fastened with stationary pillar without being under motion and its coordinate frame is considered as universal frame. The utilization of the Newton-Euler Recursive Formulations directly is incorrect in regarding to mobile robot without taking the platform motion into account. The platform attitudes (roll, pitch, and yaw) with respect to the universal frame will also be taken into account as a function of rover configurations and surface geometries. In addition, the kinematic values (position, velocity, and acceleration) of the platform link will be compensated in equations of motion with respect the kinematics of wheels.

However, the previous works have considered the whole components of mobile robot as rigid body concentrated in center of mass. But, this work dealt with the kinematics and dynamics of each link apart, and relate between links in recursive approach, which can be applied to the links sequentially to compute the kinematics and dynamics referenced in their own coordinates in a short period of time and in on-line control.

This work exploits Denavit-Hartenburg convention to assign the coordinate frames. Besides, homogeneous transformation matrix will relate between each two adjacent coordinate frames starting from base and ending at four end-effectors. Moreover, forward kinematics will directly relate the base frame to the end-effectors. Plus, the roll, pitch and yaw angles are unknown variables and they are functions of system configurations and ground geometries. As well as, The homogeneous transformations of surface frame (contact point) with respect to wheel frame of each end-effector will also be computed as functions of joint configurations and ground geometries.

Because four legs are considered indeterminate system, in this thesis the normal forces are evaluated for three contact legs in the case the nonsymmetric rover. However, in the case of symmetric configurations the normal forces are distributed equally between the sides which sharing the same the inertial forces, ground geometries, and platform attitude. Thus, regarding to four legs the normal forces are evaluated by considering each two legs sharing the same value.

A new dynamic stability criterion is presented and operating arbitrary on various shapes of surfaces, and variable rover configurations. In addition, this criterion provides on-line calculations for the effect of rover configurations, various surface geometries, platform attitudes, kinematic values, dynamic effects, and variable ground normal forces.

# Chapter Two

## 2. Kinematics of the rover

In this chapter the coordinate frames will be assigned by using Denavit-Hartenburg convention "DH", and then DH parameters will be specified between each two adjacent frames. Besides, this chapter will relate between each two adjacent coordinate frames using homogeneous transformation matrix starting from base and ending at end-effectors. Moreover, we will directly relate the base frame to the end-effectors throughout a forward kinematics. The forward kinematics provides us the position and the orientation of the end-effectors with respect to the base frame as a function of joint configurations. After all, the platform attitude will be specified with respect to universal frame through roll, pitch, and yaw orientations. However, the roll, pitch and yaw angles are unknown variables and they are functions of system configurations and ground geometries. Therefore, we will integrate the work mentioned above for finding the attitude angles. The homogeneous transformations of surface frame (contact point) with respect to wheel frame of each end-effector will also be computed as functions of joint configurations and ground geometries.

The homogeneous transformation of the ground universal frame with respect to the platform universal frame will also be computed as functions of joint configurations and ground geometries too.

### 2.1. Coordinate frames

This work considers a reconfigurable rover. The rover has four legs, and each leg consists of five links connected through four revolute joints. The first step is to return the leg to home position where all joint angles are set to home position values. Coordinate frames are assigned according to the DH convention. The joints are labeled as i= 1 to 4, and links' end-terminal are labeled with a frame number  $O_i$  (i= 0 to 4) starting from  $O_0$  as base frame (platform) to  $O_4$  as an end-effector frame (wheel). The joint axes  $z_i$  are assigned along the axes of rotation as show in Figure 2.1:



Figure 2.1. Joint axes assignments and frame numbering for the Rover.

Based on joint axes shown in Figure 2.1, we complete the three orthonormal coordinate systems ( $x_i$ ,  $y_i$ ,  $z_i$ ). For parallel joint axes,  $z_i \times z_{i-1}=0$ ,  $x_i$  axis is assigned along the common perpendicular in the line directed from frame  $O_{i-1}$  to  $O_i$ , and for intersecting joint axes,  $x_i$  axis is perpendicular to the plane or parallel to the vector cross product  $\pm z_{i-1} \times z_i$  as shown in Figure 2.2:



**Figure 2.2**.  $x_i$ -axis setting up. **a.** In parallel joint axis,  $x_i$  axis is in the line directed from frame  $O_{i-1}$  to  $O_i$ . **b.** In intersecting joint axis,  $x_i$  axis is perpendicular to the plane or parallel to the vector cross product  $\pm z_{i-1} \times z_i$ .

The  $y_i$ -axis is defined in the direction needed to complete a right-handed orthonormal coordinate frame ( $x_i$ ,  $y_i$ ,  $z_i$ ).

 $x_1$  is perpendicular to the plane containing the two intersecting axes  $z_0$ and  $z_1$ . Then  $x_0$  is to align with  $x_1$  (of course in home position).  $x_2$  is also perpendicular to the plane containing the two intersecting axes  $z_1$  and  $z_2$  in similar way in assigning the  $x_1$ . Finally because  $z_3 \times z_2=0$ ,  $x_3$  is to be assigned along the common perpendicular between the  $z_2$  and the  $z_3$  axes. These procedures will be commonly repeated for the four wheeled-legged manipulators. For the coordinate frame of End-effector link  $(x_4, y_4, z_4)$ ,  $z_4$  is assigned in parallel to  $z_3$ .  $x_4$  is assigned along the common perpendicular between the  $z_3$ and  $z_4$ .  $y_4$  assignment is based on the right hand coordinate frame. The coordinate systems  $(x_i, y_i, z_i)$ , i=0...4, from the base frame to the end-effector frame are shown in Figure 2.3:



Figure 2.3. Assignments of coordinate frame on the form of home position.

The coordinate system  $(x_i, y_i, z_i)$  for i =1,2,3,4 is assigned at the endterminal of link i and hence it moves with link i, and  $z_i$  represents the motion of link i+1. The system  $(x_0, y_0, z_0)$  is assigned at link 0, the platform center, and hence it moves with platform, and  $z_0$  represent the motion of link 1.

### 2.2. Denavit-Hartenburg parameters

So far, we have completed the designation of coordinate frames. Currently, we need to describe the kinematics of the robot by describing the position and orientation of each link with respect to the previous link using DH approach. In a simple manner, each pair of successive joints is characterized by a link length between joint axes **a**, a twisted angle between joint axes  $\alpha$ , a link offset **d**, and a joint angle  $\theta$ . The description for these four parameters can be given as follows: joint angle,  $\theta_i$ , is a rotating angle between the **x**<sub>i-1</sub> and **x**<sub>i</sub> axes about **z**<sub>i-1</sub> axis. Link offset, **d**<sub>i</sub>, is a translating distance from **x**<sub>i-1</sub> and **x**<sub>i</sub> along **z**<sub>i-1</sub>. Link length, **a**<sub>i</sub>, is a translating distance from **z**<sub>i-1</sub> and **z**<sub>i</sub> along the **x**<sub>i</sub>. Finally, twisted angle,  $\alpha_i$ , is a rotating angle between **z**<sub>i-1</sub> and **z**<sub>i</sub> axis about **x**<sub>i</sub> axis. See Appendix A.

Applying the notations of DH parameters for one manipulator and for each adjacent joints starting from base frame  $O_0$  to end-effector frame  $O_4$  as in Figure 2.4:



Figure 2.4. Pairs of two adjacent links.

Filling the table up with DH parameters, we obtain:

Link	Joint type	Variable	Link offset d	Link length a	Twist angle α
1(0-1)	revolute	$\theta_1$	$d_1$	0	-90
2(1-2)	revolute	$\theta_2$	0	0	90
3(2-3)	revolute	θ <sub>3</sub> +180	0	a <sub>3</sub>	0
4(3-4)	revolute	$\theta_4$	0	$a_4$	0

**Table 2.1.** Kinematic parameters table based on DH convention.

Since the rover has revolute joints only, all generalized coordinate variables are rotational angles about their own rotational axes. The generalized coordinates,  $(\mathbf{q}_i = \mathbf{\theta}_i, i=1,...,4)$ , describes the motion in four-dimensional vector of each legged manipulator.

$$\mathbf{q}_{i} = \begin{bmatrix} \boldsymbol{\theta}_{1} & \boldsymbol{\theta}_{2} & \boldsymbol{\theta}_{3} & \boldsymbol{\theta}_{4} \end{bmatrix}^{\mathrm{T}}$$
(2.1)

The notations of generalized coordinates will match joint velocities and joint accelerations, respectively, as follows in equations 2.2 and 2.3:

$$\dot{\mathbf{q}}_{i} = \begin{bmatrix} \dot{\boldsymbol{\theta}}_{1} & \dot{\boldsymbol{\theta}}_{2} & \dot{\boldsymbol{\theta}}_{3} & \dot{\boldsymbol{\theta}}_{4} \end{bmatrix}^{\mathrm{T}}$$
(2.2)

$$\ddot{\mathbf{q}}_{i} = \begin{bmatrix} \ddot{\mathbf{\theta}}_{1} & \ddot{\mathbf{\theta}}_{2} & \ddot{\mathbf{\theta}}_{3} & \ddot{\mathbf{\theta}}_{4} \end{bmatrix}^{\mathrm{T}}$$
(2.3)

These velocities and accelerations will be transformed forward later in Chapter 3 using Newton-Euler Recursive Relations.

## 2.3. Homogeneous transformation:

The adjacent frames are related with each other through the use  $4 \times 4$  homogeneous transformation matrix, A, which represents the orientation and position of the coordinate system O<sub>i</sub> relative to O<sub>i-1</sub>. The A<sub>i</sub><sup>i-1</sup> transformations for the rover using DH convention are given as follows:

$$A_{1}^{0} = \begin{bmatrix} C_{1} & 0 & -S_{1} & 0 \\ S_{1} & 0 & C_{1} & 0 \\ 0 & -1 & 0 & d_{1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{2}^{1} = \begin{bmatrix} C_{2} & 0 & S_{2} & 0 \\ S_{2} & 0 & -C_{2} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{3}^{2} = \begin{bmatrix} -C_{3} & S_{3} & 0 & -a_{3}C_{3} \\ -S_{3} & -C_{3} & 0 & -a_{3}S_{3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_{4}^{3} = \begin{bmatrix} C_{4} & -S_{4} & 0 & a_{4}C_{4} \\ S_{4} & C_{4} & 0 & a_{4}S_{4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(2.7)$$

For simplicity, the notations are employed to abbreviate the cosine, sine and related trigonometric formula as follows:

$$\begin{split} S_{i} &= \sin \theta_{i} \\ C_{i} &= \cos \theta_{i} \\ S_{iJ} &= S(\theta_{i} + \theta_{j}) = S \theta_{i} C \theta_{j} + C \theta_{i} S \theta_{j} \\ C_{iJ} &= C(\theta_{i} + \theta_{j}) = C \theta_{i} C \theta_{j} - S \theta_{i} S \theta_{j} \end{split}$$

### 2.4. Forward kinematics

The forward kinematics is to find the position and the orientation of the end-effector relative to base frame if the angles of joints and geometric parameters of manipulator links are given. Mathematically, it is a chain product of successive homogeneous transformations moving forward from the base frame out to the end-effector frame.

$$\begin{split} \mathbf{A}_{4}^{0} &= \mathbf{A}_{1}^{0} \cdot \mathbf{A}_{2}^{1} \cdot \mathbf{A}_{3}^{2} \cdot \mathbf{A}_{4}^{3} \\ &= \begin{bmatrix} \mathbf{C}_{1} & 0 & -\mathbf{S}_{1} & 0 \\ \mathbf{S}_{1} & 0 & \mathbf{C}_{1} & 0 \\ 0 & -1 & 0 & \mathbf{d}_{1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{C}_{2} & 0 & \mathbf{S}_{2} & 0 \\ \mathbf{S}_{2} & 0 & -\mathbf{C}_{2} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -\mathbf{C}_{3} & \mathbf{S}_{3} & 0 & -\mathbf{a}_{3}\mathbf{C}_{3} \\ -\mathbf{S}_{3} & -\mathbf{C}_{3} & 0 & -\mathbf{a}_{3}\mathbf{S}_{3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{C}_{4} & -\mathbf{S}_{4} & 0 & \mathbf{a}_{4}\mathbf{C}_{4} \\ \mathbf{S}_{4} & \mathbf{C}_{4} & 0 & \mathbf{a}_{4}\mathbf{S}_{4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{C}_{1}\mathbf{C}_{2} & -\mathbf{S}_{1} & \mathbf{C}_{1}\mathbf{S}_{2} & 0 \\ \mathbf{S}_{1}\mathbf{C}_{2} & \mathbf{C}_{1} & \mathbf{S}_{1}\mathbf{S}_{2} & 0 \\ -\mathbf{S}_{2} & 0 & \mathbf{C}_{2} & \mathbf{d}_{1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -\mathbf{C}_{34} & \mathbf{S}_{34} & 0 & -\mathbf{a}_{4}\mathbf{C}_{34} - \mathbf{a}_{3}\mathbf{C}_{3} \\ -\mathbf{S}_{34} & -\mathbf{C}_{34} & 0 & -\mathbf{a}_{4}\mathbf{S}_{34} - \mathbf{a}_{3}\mathbf{S}_{3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{split}$$

$$= \begin{bmatrix} -C_{1}C_{2}C_{34}+S_{1}S_{34} & C_{1}C_{2}S_{34}+S_{1}C_{34} & C_{1}S_{2} & -C_{1}C_{2}(a_{4}C_{34}+a_{3}C_{3})+S_{1}(a_{4}S_{34}+a_{3}S_{3}) \\ -S_{1}C_{2}C_{34}-C_{1}S_{34} & S_{1}C_{2}S_{34}-C_{1}C_{34} & S_{1}S_{2} & -S_{1}C_{2}(a_{4}C_{34}+a_{3}C_{3})-C_{1}(a_{4}S_{34}+a_{3}S_{3}) \\ S_{2}C_{34} & -S_{2}S_{34} & C_{2} & S_{2}(a_{4}C_{34}+a_{3}C_{3})+d_{1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(2.8)

The above equation, which describes the posture of the rover, is a function of joint variables, in which they were transformed into a Cartesian frame relatively to base frame. In other words, the computed matrix  $A_4^0$  can be considered generalized matrix  $B_4^0$  providing 3×3 orientation matrix and 3×1 position vector of the last frame  $O_4$  with respect to the base frame  $O_0$ . The orientation matrix describes the approach vector **a**, the orientation vector **o**, the normal vector **n**. The position vector, **p**, is the position of the end-effector with respect to base frame.

$$\mathbf{B}_{4}^{0} = \begin{bmatrix} \mathbf{n}_{x} & \mathbf{o}_{x} & \mathbf{a}_{x} & \mathbf{p}_{x} \\ \mathbf{n}_{y} & \mathbf{o}_{y} & \mathbf{a}_{y} & \mathbf{p}_{y} \\ \mathbf{n}_{z} & \mathbf{o}_{z} & \mathbf{a}_{z} & \mathbf{p}_{z} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$
(2.9)

In order to reduce the amount of computations, the first column of  $B_4^0$ may be obtained as the vector cross product of the second and third columns

$$\mathbf{n} = \mathbf{o} \times \mathbf{a} \tag{2.10}$$

The position vector from the platform's base frame to the wheel frame is the fourth column of equation 2.8; we obtain equation 2.11 which can be denoted in Figure 2.5

$$\mathbf{r}_{4}^{0} = \begin{bmatrix} -C_{1}C_{2}(a_{4}C_{34} + a_{3}C_{3}) + S_{1}(a_{4}S_{34} + a_{3}S_{3}) \\ -S_{1}C_{2}(a_{4}C_{34} + a_{3}C_{3}) - C_{1}(a_{4}S_{34} + a_{3}S_{3}) \\ S_{2}(a_{4}C_{34} + a_{3}C_{3}) + d_{1} \end{bmatrix}$$
(2.11)



Figure 2.5. Position vector from base to end-effector frame.

### 2.5. Base frame

Our mobile platform is a base link connected to a differential gear joint. Each end-terminal of this differential gear is connected with another link as shown in Figure 2.6.



Figure 2.6. Differential gear joint

There are two base frames attached on the differential gear joint, i.e.  $O_{0L}$ 

and  $O_{0R}$ , and they are located in the central platform as shown in Figure 2.7.



Figure 2.7. Two frames attached at the base link.

From the rider's point of view,  $z_{0R}$  is the right lateral axis and  $z_{0L}$  is the left lateral axis,  $y_{0R}$  is the front longitudinal axis in the direction of travel and  $y_{0L}$  is the rear longitudinal axis, finally  $x_{0R}$  and  $x_{0L}$  are axes running as one axis vertically with respect to the platform plane.

The homogeneous transformation, from right base frame to left base frame, is simply rotation about  $x_{0R}$  axis by ±180 degree. See Figure 2.7.

$$\mathbf{A}_{0L}^{0R} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(2.12)

The homogeneous transformation, from left base frame to right base frame, is also equal the transformation matrix from right to left base frame; that is simply rotation about  $x_{0L}$  by  $\pm 180$  degree.

$$\mathbf{A}_{0R}^{0L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(2.13)

## 2.6. Platform frame

However, we can choose one of the two base frames as platform frame for simplicity. In this work, we referred the right base frame as platform frame. Applying the above rules and procedures on the four legs as shown in transformation graphs in Figure 2.8, it shows the homogeneous transformations between two adjacent links. Besides, the forward kinematics transformations from the platform frame to the four wheel frames of the four legs can be obtains respectively as,

Forward kinematics for right front leg:

$$\mathbf{A}_{4RF}^{0R} = \mathbf{A}_{1R}^{0R} \cdot \mathbf{A}_{2R}^{1R} \cdot \mathbf{A}_{3RF}^{2R} \cdot \mathbf{A}_{4RF}^{3RF}$$
(2.14)

Forward kinematics for right rear leg:

$$\mathbf{A}_{4RR}^{0R} = \mathbf{A}_{1R}^{0R} \cdot \mathbf{A}_{2R}^{1R} \cdot \mathbf{A}_{3RR}^{2R} \cdot \mathbf{A}_{4RR}^{3RR}$$
(2.15)

Forward kinematics for left front leg:

$$A_{4LF}^{0R} = A_{0L}^{0R} \cdot A_{1L}^{0L} \cdot A_{2L}^{1L} \cdot A_{3LF}^{2L} \cdot A_{4LF}^{3LF}$$
(2.16)

Forward kinematics for left rear leg:

$$\mathbf{A}_{4LR}^{0R} = \mathbf{A}_{0L}^{0R} \cdot \mathbf{A}_{1L}^{0L} \cdot \mathbf{A}_{2L}^{1L} \cdot \mathbf{A}_{3LR}^{2L} \cdot \mathbf{A}_{4LR}^{3LR}$$
(2.17)



Figure 2.8. Transform graph for the four legs, starting from platform frame to endeffector frame.



Figure 2.9. The frames for the four legs,

### 2.7. Wheel kinematics

Kinematics of the wheel is a study concerned with describing the way in which the wheel moves. In this thesis, the wheels are only employed as driven links for the purpose of locomotive propulsion. In the presence of driven system, each wheel has only one rotational degree of freedom in term of angular variable. The rotational motion of the rigid wheel occurs about a rolling axis  $z_3$  by angle value  $\theta_4$ . The rolling axis is simply a line axis going perpendicularly through the center of the wheel. The translational motion of the rigid wheel occurs on the ground and in a straight line. The mechanical purposes of the wheel link and the wheel angle can be described under two factors; manipulation and locomotion:

#### - Manipulation factor:

In the case of manipulation purposes, the wheel end-effector frame is simply a touch point with surface. It is denoted with  $O_4$  ( $x_4$ ,  $y_4$ ,  $z_4$ ) and is setting up as follows:  $x_4$  axis is normal to the rim of wheel;  $y_4$  axis is in tangent direction of the rim of wheel; and  $z_4$  axis is directed perpendicularly to wheel plane. See Figure 2.10 which shows three selected points ( $P_1$ ,  $P_2$ , and  $P_3$ ) on the rim of wheel.



Figure 2.10. The coordinate frame of manipulated wheel.

The position vector of contact point  $O_4$  with respect to base frame  $O_0$  is dependent on the configurations of rover  $(\theta_1, \theta_2, \theta_3, \theta_4)$ . The manipulated variable  $\theta_4$  is simply arc angle which is rotating about  $x_3$  axis is dependent on differential joint angle $\theta_1$ , shoulder joint angle  $\theta_3$ , pitch angle  $\theta$ , and surface geometry  $\beta_3$ . These factors will be explained in coming sections.

The manipulation angle  $\theta_4$  (arc angle value) is considered in the following calculations:

- 1. Roll angle  $(\phi)$ .
- 2. Forward kinematics: the orientation matrix and position vector of the manipulated links with respect to base frame must treat  $\theta_4$ .
- 3. Inverse kinematics.

### - Locomotion factor:

In locomotion purposes, the wheel is considered as driven link with  $\theta_4$  which in turn is considered the angle of wheel rotation generated from motor torque. The angular acceleration of the wheel can be evaluated as,

$$\ddot{\theta}_4 = \frac{\dot{\mathbf{v}}_{t4}}{\mathbf{a}_4} \tag{2.18}$$

Then in general, the kinematic equations of the rotational wheel are determined under assumed constant angular acceleration  $\ddot{\theta}_4$  as follows

$$\dot{\theta}_4 = \dot{\theta}_{4,0} + \ddot{\theta}_4 \mathbf{t} \tag{2.19}$$

$$\theta_4 = \theta_{4,0} + \dot{\theta}_{4,0} t + \frac{1}{2} \ddot{\theta}_4 t^2$$
(2.20)

The travel length of wheel movements on ground is directly proportional to generalized joint coordinate of wheel link  $\theta_4$  and the radius of wheel  $a_4$ .



Figure 2.11. Wheel angular movement tracked linearly on ground.

The locomotive angle  $\theta_4$  (rotation angle value) is considered in these calculations:

- 1. Linear displacement, velocity, and acceleration of wheel tracked linearly on the ground.
- 2. Linear displacement, velocity and acceleration of platform frame with respect to universal frame expressed in universal frame, i.e.  $v_{U}^{U}$  and  $\dot{v}_{U}^{U}$  respectively.
- 3. Generalized coordinates of angular displacement, velocity and acceleration  $(q_4, \dot{q}_4, \ddot{q}_4)$  those are substituted in forward recursion.
- 4. In addition, yaw angle  $\psi$  resulted from variance of wheels' velocities.

(2.21)

### 2.8. Platform kinematics

The above mentioned can be extended by including a coordination between the locomotion and manipulation. Each two legged manipulators on both sides of the rover are locomoted by two wheels at same velocity. However, each side is locomoted at different velocity relatively to the opposite side. These differences in velocities between two opposite sides will rotate the faster side around the slower side, and in result these will rotate the entire rover about the yaw axis  $x_U$ .

The rover moves in forward and backward direction according to the fixed rotation of the wheels on ground, and rotates on right and left direction according to the difference of wheels' velocities. Thus, we will define the relationship between the angular velocity of the wheels and the travel path of the vehicle body on the ground. In addition, we will coordinate the processes of locomotion with manipulation expressed in the universal frame.

We will make our calculations dependent on the contact wheels with ground. Moreover, we will choose kinematic values of one wheel from each
side, i.e.  $\theta_{4R}$  and  $\theta_{4L}$ , even if the all wheels are in contact with ground. On the right side  $\theta_{4R}$  rotates about  $z_{3R}$ , and on the left side  $\theta_{4L}$  rotates about  $z_{3L}$  in term of counter clock wise direction. However  $z_{3L}$  has inverse direction relatively to  $z_{3R}$  as shown in Figure 2.12.  $z_{3L}$  can be transformed to be pointing to the direction of  $z_{3R}$  by multiplying  $\theta_{4L}$  by negative sign.

In order for moving forward, it is required to manipulate the configurations of the wheels in adequate angles and direction.  $\theta_{4R}$  must rotate in counter clockwise direction in positive valued and  $\theta_{4L}$  must rotate in clockwise direction in negative value as shown in the following Figure 2.12



Figure 2.12. Two opposite wheels enabling the rover for rotating forward, the arc length of the wheel is tracked on ground, from start to finish of the travel.

Therefore, the travel path of right wheel and left wheel on ground can be obtained respectively as

$$\mathbf{d}_{\mathrm{R}} = \mathbf{a}_4 \cdot \mathbf{\theta}_{4\mathrm{R}} \tag{2.22}$$

$$\mathbf{d}_{\mathrm{L}} = \mathbf{a}_4 \cdot \mathbf{\theta}_{4\mathrm{L}} \tag{2.23}$$

The rover travel is generated from the linear movement of wheels on the ground. The travel path of the rover body is the average of the travel lengths of right and left wheels.

$$d_{\rm B} = \frac{(d_{\rm R} + d_{\rm L})}{2} = \frac{a_4 \cdot \theta_{4\rm R} + a_4 \cdot \theta_{4\rm L}}{2}$$
(2.24)

The instantaneous linear velocity of the wheels is equal the derivation of travel path with respect to time, or in other words, the rate of change in the travel path with respect to time.

$$\mathbf{v}_{t4R} = \mathbf{a}_4 \cdot \dot{\mathbf{\theta}}_{4R} \tag{2.25}$$

$$\mathbf{v}_{\mathrm{tLR}} = \mathbf{a}_4 \cdot \dot{\mathbf{\theta}}_{\mathrm{4L}} \tag{2.26}$$

The robot's velocity is the rate of change in the robot's position with respect to time. Thus, linear velocity of robot body is rate of change of the average of the wheeled travel lengths with respect to time.

$$\mathbf{v}_{\mathrm{B}} = \frac{\mathbf{a}_{4} \cdot \dot{\mathbf{\theta}}_{4\mathrm{R}} + \mathbf{a}_{4} \cdot \dot{\mathbf{\theta}}_{4\mathrm{L}}}{2} \tag{2.27}$$

The linear acceleration of the wheel is the rate of change of the velocity of the wheel with respect to time

$$\dot{\mathbf{v}}_{t4R} = \mathbf{a}_4 \cdot \ddot{\mathbf{\theta}}_{4R} \tag{2.28}$$

$$\dot{\mathbf{v}}_{\mathrm{tLR}} = \mathbf{a}_4 \cdot \ddot{\mathbf{\theta}}_{\mathrm{4L}} \tag{2.29}$$

The robot's acceleration is the rate of change in the robot's velocity with respect to time

$$\dot{\mathbf{v}}_{\mathrm{B}} = \frac{\mathbf{a}_{4} \cdot \ddot{\boldsymbol{\theta}}_{4\mathrm{R}} + \mathbf{a}_{4} \cdot \ddot{\boldsymbol{\theta}}_{4\mathrm{L}}}{2} \tag{2.30}$$

In any way, the rover motion on the non-flat surface and the links motions about their joint axes yield a change in the orientations of the platform frame. This different platform's attitude will be referred with respect to the universal frame. Both universal frame and platform frame have same origin on the center of platform (no translation), but different orientations as shown in Figure 2.8. These orientations can be described in different techniques, e.g. Roll Pitch and Yaw, Euler Angle representation, or Directional Cosine representation. This work chose Roll, Pitch and Yaw method.

### 2.9. Platform universal frame

The rover motions are referred with respect to right-hand orthogonal coordinate frame, called the universal frame  $O_U(x_U, y_U, z_U)$ . In other words, the coordinate frames and equations of motion of each link are considered with respect to. It is located at the center of platform forming the horizontal plane,  $y_U$ - $z_U$ , parallel to ground plane, and  $x_U$  axis is normal to ground plane directed upward as shown in Figure 2.13.

The Orientations are measured from the attitude of the body in three dimensions (Roll, Pitch and Yaw). These independent motions cause three rotational degrees-of-freedom as shown in Figure 2.13. In our case, zero translation is between the two frames. The body's attitude, which is referred about the universal frame, can be broken down into: roll corresponds to a rotation  $\phi$  about the longitudinal y<sub>U</sub>-axis, pitch corresponds to a rotation  $\theta$ about the lateral z<sub>U</sub>-axis and yaw corresponds to a rotation  $\psi$  about the normal x<sub>U</sub>-axis. As supposed the sequential order of rotations is as following:

- Rotation of  $\psi$  about  $x_U$ -axis.
- Rotation of  $\theta$  about  $z_U$ -axis.
- Rotation of  $\phi$  about  $y_U$ -axis.



**Figure 2.13.** Body attitude provides three rotational degrees-of-freedom ( $\phi$ ,  $\theta$ ,  $\psi$ ), assuming congruent frames for platform and universal frame at the beginning.

In simpler manner, any 3-Dimensional rotation is conventionally defined as a rotation in 2-Dimensional counter-clockwise direction along positive axis of rotation. So firstly we specify the axes of rotations about universal frame, and secondly the rotation angles in radian as shown in Figure 2.14:



**Figure 2.14.** Roll motions about  $y_U$  axis by  $\phi$  angle, Pitch motions about  $z_U$  axis by  $\theta$  angle, and Yaw motions about  $x_U$  axis by  $\psi$  angle.

These series of body's rotations, around the universal frame, can be described in three matrices. Moreover, these matrices can be combined by multiplications with each other orderly as follows:

 $A_{0R}^{U} = RPY(\phi, \theta, \psi) = Rot(y_{U}, \phi) Rot(z_{U}, \theta) Rot(x_{U}, \psi)$ 

=	$\begin{bmatrix} \cos \phi \\ 0 \\ -\sin \phi \\ 0 \end{bmatrix}$	0 1 0 0	$     \sin \phi \\     0 \\     \cos \phi \\     0 $	$\begin{bmatrix} 0\\0\\0\\1\end{bmatrix} . \begin{bmatrix} 0\\1\\1\end{bmatrix}$	$   \cos \theta $ $   \sin \theta $ $   0 $ $   0 $	$-\sin \theta$ $\cos \theta$ $0$ $0$	0 0 1 0	0 0 0 1	1 0 0 0	$ \begin{array}{c} 0\\ \cos\psi\\ \sin\psi\\ 0 \end{array} $	$\begin{array}{c} 0 \\ -\sin \psi \\ \cos \psi \\ 0 \end{array}$	0 0 0 1		
=	$\begin{bmatrix} c\phi c\theta \\ s\theta \\ -s\phi c\theta \\ 0 \end{bmatrix}$	-c	¢sθcy cθ φsθcy (	ν+sφs cψ ν+cφs )	sψ Ψ -	cφsθsψ –ct –sφsθsy (	y+s θsψ ψ+e	φςψ , ςφςψ	0 0 0 1				(2	2.31)

 $A_{0R}^{U}$  is a homogeneous transformation from the universal frame  $O_{U}$  to the body frame  $O_{0R}$ . These three attitude angles can be generated as a result of influences of geometric configurations of the manipulators and ground geometries. However, the order of rotations is an important, which means it is not commutative. Thus, the sequential order of rotations is not a matter of suppositions, but it is definitely subjected to the orders of sudden changes in ground geometries and joint configurations that will cause platform's orientations. If the platform is congruent to a universal base frame, the homogeneous transformation from the universal frame  $O_U$  to the body frame  $O_{0R}$  will be unity matrix:

$$\mathbf{A}_{0R}^{U} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(2.32)

The homogeneous transformation between universal and left base frame is post-multiplying  $A_{0R}^U$  by  $A_{0L}^{0R}$ 

$$A_{0L}^{U} = A_{0R}^{U} \cdot A_{0L}^{0R}$$

$$= \begin{bmatrix} c\phi c\theta & c\phi s\theta c\psi - s\phi s\psi & -c\phi s\theta s\psi - s\phi c\psi & 0\\ s\theta & -c\theta c\psi & c\theta s\psi & 0\\ -s\phi c\theta & -s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi - c\phi c\psi & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(2.33)$$

$$O_{U}$$

$$A_{0L}^{U}$$

$$A_{0R}^{U}$$

$$O_{0R}$$

Figure 2.15. Transform graph of universal frame and two bases frames.



Figure 2.16. The transform graph of rover frames.

#### 2.9.1. Attitude angles

The previous roll, pitch, and yaw angles of platform frame with respect to universal frame are influenced by joint configurations of the manipulators and geometric ground input systems. The geometric configurations of the manipulators are function of joint variables  $(\theta_1, \theta_2, \theta_3, \theta_4)$  that formulate the rover posture.  $\theta_4$  must be taken into accounts that where it must be treated as manipulation purpose or locomotive purpose. The ground input systems are functions of altitudes from ground universal level to wheeled-ground contact points.

The calculation of stability measure must meet conditions required when at least three legs are in contact with the ground surface all the time. Meanwhile, one supported leg from each side (one from left side and the second from right side) is enough for covering the required calculations as shown in Figure 2.17.



Figure 2.17. The geometric configurations and ground input Systems relative to the two legged manipulators.

The robot remains stable with three supported legs on ground while the fourth leg remains without contact. One supported leg from each side is chosen for our computations and we will remark it by K. K is stands for the chosen leg and it is either front leg or rear leg under conditions of connectivity with ground as shown bellow

$$K = \begin{cases} F; & if(F \mapsto \bullet) | (R \mapsto \bullet) \\ R; & if(F \mapsto \circ) \& (R \mapsto \bullet) \end{cases}$$

leg  $\mapsto \bullet$  means that the leg is in contact with ground.

leg  $\mapsto \circ$  means that the leg is not in contact with ground.

As explained in this example: assume right front leg is on air without contact with ground as shown in the following Figure 2.18



Figure 2.18. The black circle indicates for supported legs and white circle indicates for not supported legs with the ground.

On right side, the front leg is denoted with RK; and on the left side, the rear leg is represented with LK.

#### 2.9.1.1 Pitch angle

Pitch angle corresponds to a rotation of the platform by  $\theta$  about the lateral  $z_U$  axis as a result of differential joint rotation, rover configurations and surface geometries.  $z_{0R}$  axis and  $z_U$  axis are contingent and pointing toward the right lateral side of the platform.  $z_{0L}$  axis is in opposite direction of  $z_U$  axis pointing toward the left lateral side.

The pitch angle is firstly resulted from the difference average between the angle of right rotary link  $\theta_{1R}$  and the angle of left rotary link  $\theta_{1L}$  as shown in Figure 2.19:

$$\theta^{(1)} = \frac{\theta_{1R} - \theta_{1L}}{2}$$

$$Z_{0L}$$

$$\theta_{1L}$$

$$Differential joint$$

$$\theta_{1R}$$

$$\theta_{2U}, Z_{0R}$$

$$(2.34)$$

Figure 2.19. Pitch angle.

 $\theta$  is valued a positive angle about  $z_U$ -axis, when the platform rotates in counter-clock wise direction, or on other word, when  $\theta_{1R}$  is positively greater than  $\theta_{1L}$ .



Figure 2.20. Rotation about lateral axis of universal frame by  $\theta$ .



## Figure 2.21. Elevation difference

And secondly it resulted from the elevation difference between the front and rear legs and rover configurations. Applying Pythagorean relations, the trigonometric sine function is

$$\sin \theta^{(2)} = \frac{\left(r_{4RR}^{G}\right)_{x_{U}} - \left(r_{4RF}^{G}\right)_{x_{U}}}{\left(r_{4RR}^{OR}\right)_{y_{0R}} - \left(r_{4RF}^{OR}\right)_{y_{0R}}}$$
$$\theta^{(2)} = \sin^{-1} \left(\frac{\left(r_{4RR}^{G}\right)_{x_{U}} - \left(r_{4RF}^{G}\right)_{x_{U}}}{\left(r_{4RR}^{O}\right)_{y_{0R}} - \left(r_{4RF}^{O}\right)_{y_{0R}}}\right)$$
(2.35)

Where,

67

$$\left(r_{4RR}^{0R}\right)_{y_{0R}} = -S_{1R}C_{2R}\left(a_{4}C_{34RF} + a_{3}C_{3RF}\right) - C_{1R}\left(a_{4}S_{34RF} + a_{3}S_{3RF}\right)$$
(2.36)

$$\left(r_{4RR}^{0R}\right)_{y_{0R}} = -S_{1R}C_{2R}\left(a_{4}C_{34RR} + a_{3}C_{3RR}\right) - C_{1R}\left(a_{4}S_{34RR} + a_{3}S_{3RR}\right)$$
(2.37)

Finally the pitch angle is equal to

$$\theta = \theta^{(1)} + \theta^{(2)} \tag{2.38}$$

68

## 2.9.1.2 Roll angle

The roll angle corresponds to a rotation of the platform by  $\phi$  about the longitudinal  $y_U$  axis.  $y_{0R}$  axis and  $y_U$  axis will be contingent and pointing toward the front longitudinal view of the platform if and only if the rover is manipulated at symmetric configurations and moving on flat surface:



**Figure 2.22.** Front view shows the platform rotating about longitudinal axis of universal frame by  $\phi$ . y<sub>0</sub>-axis and y<sub>U</sub>-axis are pointing out of paper.

The wheel frame  $O_{4RK}$  is assigned at the contact point located as an endeffector where  $\theta_{4RK} = -\theta_{1R} - \theta_{3RK} + \beta - \theta$ . The roll angle about  $y_U$  axis is computed by using Pythagorean relations; the opposite side is the altitude difference between the RCF and LCF altitudes; the hypotenuse side is the lateral length of the platform.



Figure 2.23. Pythagorean relations.

The trigonometric sine function is

$$\sin\phi = \frac{\text{RCF altitude} - \text{LCF altitude}}{\text{platform lateral length}}$$
(2.39)

Where,

RCF altitude =  ${}^{\rm U} \left( {\mathbf{r}_{\rm G}^{\rm 1R}} \right)_{{\mathbf{x}_{\rm U}}}$ 

LCF altitude =  ${}^{\rm U}(\mathbf{r}_{\rm G}^{\rm 1L})_{\mathbf{x}_{\rm II}}$ 

Platform lateral length =  $2d_1$ 

 $\phi$  is valued a positive angle about  $y_U$  axis, when the platform rotates in counterclockwise direction, or on other words, when the RCF altitude is higher than the LCF altitude.

### - Right conjunctional Altitude:

Mathematically, the altitude from  $O_{1R}$  to ground frame  $O_G$  is equal the summation of the altitude from  $O_{1R}$  to  $O_{4RK}$ , and the altitude from  $O_{4RK}$  to ground frame  $O_G$ 

These calculations have to be referred to universal frame as shown in Figure 2.24:



Figure 2.24. The altitude of RCF  $O_{1R}$  to ground frame  $O_G$ 

The mathematical subtraction of  $r_{4RK}^U$  and  $r_{1R}^U$  will provide us  $(r_{4RK}^{1R})_U$ , where  $(r_{1R}^U)_{x_U}$  is the first row and fourth column of the homogeneous transformation matrix of frame  $O_{1R}$  with respect to universal frame  $O_U$ 

$$\begin{split} \mathbf{A}_{1R}^{U} &= \mathbf{A}_{0R}^{U} \cdot \mathbf{A}_{1R}^{0R} \\ &= \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{C}_{1R} & 0 & -\mathbf{S}_{1R} & 0 \\ \mathbf{S}_{1R} & 0 & \mathbf{C}_{1R} & 0 \\ 0 & -1 & 0 & \mathbf{d}_{1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\phi \mathbf{C}_{1R} & -\sin\phi & -\cos\phi \mathbf{S}_{1R} & \mathbf{d}_{1}\sin\phi \\ \mathbf{S}_{1R} & 0 & \mathbf{C}_{1R} & 0 \\ -\sin\phi \mathbf{C}_{1R} & -\cos\phi & \sin\phi \mathbf{S}_{1R} & \mathbf{d}_{1}\cos\phi \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{split}$$
(2.41)

Thus,

$$\left(\mathbf{r}_{1\mathrm{R}}^{\mathrm{U}}\right)_{\mathbf{x}_{\mathrm{U}}} = \mathbf{d}_{1}\sin\phi \tag{2.42}$$

And  $(r_{4RK}^{U})_{x_{U}}$  is the first row and fourth column of homogeneous transformation matrix from universal frame  $O_{U}$  to end-effector frame  $O_{4RK}$ ,

$$(\mathbf{r}_{4RK}^{U})_{x_{U}} = \cos\phi \cdot (-C_{1R}C_{2R}(a_{4}C_{34RK} + a_{3}C_{3RK}) + S_{1R}(a_{4}S_{34RK} + a_{3}S_{3RK})) + \sin\phi \cdot (S_{2R}(a_{4}C_{34RK} + a_{3}C_{3RK}) + d_{1})$$
(2.43)

Finally as mentioned,

$$\left( \mathbf{r}_{G}^{1R} \right)_{x_{U}} = \cos \phi \cdot \left( -C_{1R} C_{2R} \left( a_{4} C_{34RK} + a_{3} C_{3RK} \right) + S_{1R} \left( a_{4} S_{34RK} + a_{3} S_{3RK} \right) \right) + \\ \sin \phi \cdot \left( S_{2R} \left( a_{4} C_{34RK} + a_{3} C_{3RK} \right) \right) + \left( \mathbf{r}_{G}^{4RK} \right)_{x_{U}}$$

$$(2.44)$$

## - Left conjunctional Altitude

Mathematically, The altitude from frame  $O_{1R}$  to frame  $O_{G}$  is mathematical summation of  $x_{U}$ -component of position vectors from LCP frame  $O_{1L}$  to GCP,  $(r_{4LK}^{1L})_{x_{U}}$ , and system input from GCP to ground frame,  $(r_{G}^{4LK})_{x_{U}}$ :

Where, mathematical vectors subtraction of  $\mathbf{r}_{4LK}^U$  and  $\mathbf{r}_{1L}^U$  will provide us  $(\mathbf{r}_{4LK}^{1L})_U$ ;  $\mathbf{r}_{1L}^U$  is the fourth column of the homogeneous transformation matrix of frame  $O_{1L}$  with respect to universal frame  $O_U$ :

$$\begin{aligned} \mathbf{A}_{1L}^{U} &= \mathbf{A}_{0R}^{U} \cdot \mathbf{A}_{0L}^{0R} \cdot \mathbf{A}_{1L}^{0L} \\ &= \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C_{1L} & 0 & -S_{1L} & 0 \\ S_{1L} & 0 & C_{1L} & 0 \\ 0 & -1 & 0 & d_{1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\phi C_{1L} & \sin\phi & -\cos\phi S_{1L} & -d_{1} \sin\phi \\ -S_{1L} & 0 & -C_{1L} & 0 \\ -\sin\phi C_{1L} & \cos\phi & \sin\phi S_{1L} & -d_{1} \cos\phi \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$
(2.46)

Thus,

$$\left(\mathbf{r}_{1L}^{\mathrm{U}}\right)_{\mathbf{x}_{\mathrm{U}}} = -\mathbf{d}_{1}\sin\phi \tag{2.47}$$



Figure 2.25. Coordinate frames of  $O_U$ ,  $O_{0R}$ ,  $O_{0L}$  and  $O_{1L}$ 

And  $(r_{4LK}^{U})_{x_{U}}$  is the first row and fourth column of homogeneous transformation matrix from frame  $O_{U}$  to end-effector frame  $O_{4LK}$ ,

$$(\mathbf{r}_{4LK}^{U})_{x_{U}} = \cos\phi \cdot (-C_{1L}C_{2L}(a_{4}C_{34LK} + a_{3}C_{3LK}) + S_{1L}(a_{4}S_{34LK} + a_{3}S_{3LK})) - \sin\phi \cdot (S_{2L}(a_{4}C_{34LK} + a_{3}C_{3LK}) + d_{1})$$
(2.48)

Finally,

$$\left( \mathbf{r}_{G}^{1L} \right)_{x_{U}} = \cos \phi \cdot \left( -C_{1L} C_{2L} (a_{4} C_{34LK} + a_{3} C_{3LK}) + S_{1L} (a_{4} S_{34LK} + a_{3} S_{3LK}) \right) - \\ \sin \phi \cdot \left( S_{2L} (a_{4} C_{34LK} + a_{3} C_{3LK}) \right) + \left( \mathbf{r}_{G}^{4LK} \right)_{x_{U}}$$

$$(2.49)$$

The roll angle about  $y_U$ -axis can be obtained by using Pythagorean relations as shown in Figure 2.23:

The geometric sine function as explained

$$\sin \phi = \frac{\left(r_{G}^{1R}\right)_{x_{U}} - \left(r_{G}^{1L}\right)_{x_{U}}}{2d_{1}}$$

$$= \frac{\left(\cos \phi \cdot \left(-C_{1R}C_{2R}(a_{4}C_{34RK} + a_{3}C_{3RK}) + S_{1R}(a_{4}S_{34RK} + a_{3}S_{3RK}) + \right)_{R} + \left(a_{4}C_{34RK} + a_{3}C_{3RK}\right) + C_{1L}C_{2L}(a_{4}C_{34LK} + a_{3}C_{3LK}) - S_{1L}(a_{4}S_{34LK} + a_{3}S_{3LK})\right)_{R}}{\sin \phi \cdot \left(S_{2R}(a_{4}C_{34RK} + a_{3}C_{3RK}) + S_{2L}(a_{4}C_{34LK} + a_{3}C_{3LK})\right) + \left(r_{G}^{4RK}\right)_{x_{U}} - \left(r_{G}^{4LK}\right)_{x_{U}}\right)}$$

$$= \frac{2d_{1}}{2d_{1}}$$
(2.50)

$$2 d_1$$

Where,

$$\begin{pmatrix} r_{4RK}^{0R} \end{pmatrix}_{z_{0R}} = S_{2R} (a_4 C_{34RK} + a_3 C_{3RK}) + d_1$$

$$S_{2R} (a_4 C_{34RK} + a_3 C_{3RK}) = (r_{4RK}^{0R})_{z_{0R}} - d_1$$

$$\begin{pmatrix} r_{4LK}^{0R} \end{pmatrix}_{z_{0R}} = -S_{2L} (a_4 C_{34LK} + a_3 C_{3LK}) - d_1$$

$$S_{2L} (a_4 C_{34LK} + a_3 C_{3LK}) = -(r_{4LK}^{0R})_{z_{0R}} - d_1$$

and,

$$-C_{1R}C_{2R}(a_{4}C_{34RK}+a_{3}C_{3RK})+S_{1R}(a_{4}S_{34RK}+a_{3}S_{3RK})=(r_{4RK}^{0R})_{x_{0R}}$$
$$-C_{1L}C_{2L}(a_{4}C_{34LK}+a_{3}C_{3LK})+S_{1L}(a_{4}S_{34LK}+a_{3}S_{3LK})=(r_{4LK}^{0R})_{x_{0R}}$$

Therefore, this trigonometric equation can be simplified as fellows

$$\sin\phi = \frac{\cos\phi \cdot \left( \left( r_{4RK}^{0R} \right)_{x_{0R}} - \left( r_{4LK}^{0R} \right)_{x_{0R}} \right) + \sin\phi \cdot \left( \left( r_{4RK}^{0R} \right)_{z_{0R}} - \left( r_{4LK}^{0R} \right)_{z_{0R}} - 2d_1 \right) + \left( r_G^{4RK} \right)_{x_U} - \left( r_G^{4LK} \right)_{x_U} - 2d_1 \right)}{2d_1}$$
(2.51)

Rearranging and simplifying the above equation,

$$\cos\phi \cdot \left( \left( r_{4\,RK}^{0\,R} \right)_{X_{0R}} - \left( r_{4\,LK}^{0\,R} \right)_{X_{0R}} \right) + \sin\phi \cdot \left( \left( r_{4\,RK}^{0\,R} \right)_{Z_{0R}} - \left( r_{4\,LK}^{0\,R} \right)_{Z_{0R}} - 4\,d_1 \right) = -\left( r_{G}^{4\,RK} \right)_{X_U} + \left( r_{G}^{4\,LK} \right)_{X_U}$$
(2.52)

Applying the trigonometric identity [63] in order to transform eq.2.52 into a basic trigonometric equation,

$$\sqrt{\left(\left(r_{4\,RK}^{0\,R}\right)_{x_{0R}} - \left(r_{4\,LK}^{0\,R}\right)_{x_{0R}}\right)^{2} + \left(\left(r_{4\,RK}^{0\,R}\right)_{z_{0R}} - \left(r_{4\,LK}^{0\,R}\right)_{z_{0R}} - 4\,d_{1}\right)^{2}}\sin(\phi + \alpha) = -\left(r_{G}^{4\,RK}\right)_{x_{U}} + \left(r_{G}^{4\,LK}\right)_{x_{U}}$$
(2.53)

The equation becomes,

$$\sin(\phi + \alpha) = \frac{-\left(r_{G}^{4RK}\right)_{x_{U}} + \left(r_{G}^{4LK}\right)_{x_{U}}}{\sqrt{\left(\left(r_{4RK}^{0R}\right)_{x_{0R}} - \left(r_{4LK}^{0R}\right)_{x_{0R}}\right)^{2} + \left(\left(r_{4RK}^{0R}\right)_{z_{0R}} - \left(r_{4LK}^{0R}\right)_{z_{0R}} - 4d_{1}\right)^{2}}}$$
(2.54)

Finally, the roll angle is

$$\phi = \sin^{-1} \left( \frac{-\left( r_{G}^{4\,RK} \right)_{x_{U}} + \left( r_{G}^{4\,LK} \right)_{x_{U}}}{\sqrt{\left( \left( r_{4\,RK}^{0\,R} \right)_{x_{0R}} - \left( r_{4\,LK}^{0\,R} \right)_{x_{0R}} \right)^{2} + \left( \left( r_{4\,RK}^{0\,R} \right)_{z_{0R}} - \left( r_{4\,LK}^{0\,R} \right)_{z_{0R}} - 4\,d_{1} \right)^{2}} \right) - \alpha$$
(2.55)

Where,

$$\alpha = \begin{cases} \tan^{-1} \left( \frac{\left( r_{4RK}^{0R} \right)_{x_{0R}} - \left( r_{4LK}^{0R} \right)_{x_{0R}}}{\left( \overline{r_{4RK}^{0R} \right)_{z_{0R}} - \left( r_{4LK}^{0R} \right)_{z_{0R}} - 4 d_1} \right); & \text{if } \left( \left( r_{4RK}^{0R} \right)_{z_{0R}} - \left( r_{4LK}^{0R} \right)_{z_{0R}} - 4 d_1 \right) > 0 \\ \pi + \tan^{-1} \left( \frac{\left( r_{4RK}^{0R} \right)_{x_{0R}} - \left( r_{4LK}^{0R} \right)_{x_{0R}}}{\left( \overline{r_{4RK}^{0R} \right)_{z_{0R}} - \left( r_{4LK}^{0R} \right)_{z_{0R}} - 4 d_1} \right); & \text{if } \left( \left( r_{4RK}^{0R} \right)_{z_{0R}} - \left( r_{4LK}^{0R} \right)_{z_{0R}} - 4 d_1 \right) < 0 \end{cases}$$

$$(2.56)$$

# 2.9.1.3 Yaw angle

The yaw angle corresponds to a rotation about the vertical axis of the platform universal frame;  $x_U$ -axis represents the axis of rotation, and  $\psi$  represents the angle value of rotation as shown in Figure 2.26:



**Figure 2.26.** Rotation about vertical axis of universal frame by  $\psi$ .

Explicitly, the human eyes can recognize the yaw rotation as a horizontal change in the positions of right and left conjunctions, as shown in the following side and front view Figure 2.27.



Figure 2.27. Side view shows the difference in the locations of LC and RC, and top view shows the rotational yaw angle  $\psi$  occurred between the universal and platform frame.

Implicitly, the yaw angle is yielded from the differential velocities between the right and lift locomotive wheels. The different wheels' velocities make the faster wheel rotating around the slower wheel, as well as make the entire rover rotating around the  $x_U$  component of the universal frame. This difference defines the relationship between the movements of the wheels and the orientation of the rover with respect to universal frame. In general, the travel of the rover is generated from the angular movements of locomotive wheels on the ground. The travel length of wheel movement (d) on ground is directly proportional to generalized joint coordinate of wheel link ( $\theta_4$ ) and the radius of wheel ( $a_4$ ).

The length of travel may be tracked along either line path if the two sides move with the same velocity or arc path if the wheels on one side move faster than that opposite side as shown in Figure 2.28:



Figure 2.28. a. Arc path occurs when  $a_4\theta_{4R} \neq a_4\theta_{4L}$ b. Line path occurs whenever  $a_4\theta_{4R} = a_4\theta_{4L}$ 

The rover posture on Figure 2.28.a shows that  $\psi$  is positive value angle rotating around  $x_U$  axis in counter clockwise direction whenever the left wheels are moving at higher speed than the right wheels.

The rover will travel forward along  $y_{0R}$  axis if  $\theta_{4R}$  rotates about  $z_{3R}$  in counter-clock wise direction and  $\theta_{4L}$  rotates about  $z_{3L}$  in clock wise direction, and vice versa.

Mathematically, we can find yaw angle through using arc laws. The arc length is the difference in the number of wheeled rotations on ground i.e. opposite =  $a_4\theta_{4L} - a_4\theta_{4R}$ , the radius is the lateral distance between the opposite wheels. As shown in Figure 2.12, the arc length for the wheels on right side and left side, respectively, can be obtain as

$$\mathbf{A} \cdot \boldsymbol{\psi} = \mathbf{a}_4 \,\boldsymbol{\theta}_{4\mathrm{R}} \tag{2.57}$$

$$\mathbf{B} \cdot \boldsymbol{\psi} = \mathbf{a}_4 \boldsymbol{\theta}_{4\mathrm{L}} \tag{2.58}$$

Subtracting the two equations from each other, we obtain

$$(\mathbf{B}-\mathbf{A}) \cdot \boldsymbol{\psi} = \mathbf{a}_{4} \,\boldsymbol{\theta}_{4\mathrm{L}} - \mathbf{a}_{4} \,\boldsymbol{\theta}_{4\mathrm{R}}$$

$$2 \,\mathbf{d}_{1} \cdot \boldsymbol{\psi} = \mathbf{a}_{4} \boldsymbol{\theta}_{4\mathrm{L}} - \mathbf{a}_{4} \boldsymbol{\theta}_{4\mathrm{R}}$$

$$\boldsymbol{\psi} = \frac{\mathbf{a}_{4} \boldsymbol{\theta}_{4\mathrm{L}} - \mathbf{a}_{4} \boldsymbol{\theta}_{4\mathrm{R}}}{2 \,\mathbf{d}_{1}}$$

$$(2.59)$$

The above equation can be extended to include the case when the rover opens its legs aside as shown in Figure 2.29



Figure 2.29. Front view shows the four legs open by an angle about  $z_{1R}$ 

$$\Psi = \frac{a_4 \theta_{4L} - a_4 \theta_{4R}}{\left(r_{4RK}^{0R}\right)_{z_{0R}} - \left(r_{4LK}^{0R}\right)_{z_{0R}}} = \frac{a_4 \theta_{4L} - a_4 \theta_{4R}}{S_{2R} (a_4 C_{34RK} + a_3 C_{3RK}) + d_1 - \left(-S_{2L} (a_4 C_{34LK} + a_3 C_{3LK}) - d_1\right)} = \frac{a_4 \theta_{4L} - a_4 \theta_{4R}}{S_{2R} (a_4 C_{34RK} + a_3 C_{3RK}) + S_{2L} (a_4 C_{34LK} + a_3 C_{3LK}) + 2d_1}$$
(2.60)

#### 2.10. Rover transform graph

So far, we find the homogeneous transformation from universal frame  $O_U$  to base frame  $O_0$ , and the forward kinematic transformations from base frame  $O_0$  to wheel frame  $O_4$ . However, the homogeneous transformations starting from wheel frame  $O_4$  passing by surface frame  $O_s$  to ground frame  $O_G$  are not yet defined. Moreover, the homogeneous transformation from platform universal  $O_U$  to ground universal  $O_G$  is also not computed. Therefore, the rover transform graph is not completed as shown in Figure 2.31.

#### 2.10.1. Ground universal frame

The platform universal frame  $O_U$  and ground universal frame  $O_G$  have the same orientations which are fixed, but there is variable position,  $r_G^U$ , separating between these two frames. Therefore, the 3×3 rotational matrix that relates these two frames,  $O_U$  and  $O_G$ , is identity matrix, and the 3×1 position vector is a function of the input system and the configurations of manipulators.

$$\mathbf{A}_{\mathrm{G}}^{\mathrm{U}} = \begin{bmatrix} \mathbf{I}_{3\times3} & \mathbf{r}_{\mathrm{G}\ 3\times1}^{\mathrm{U}} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$
(2.61)

The variable position vector,  $\mathbf{r}_{G}^{U}$ , is the summation of the position vector  $\mathbf{r}_{4}^{U}$  from platform universal frame to wheel frame and the position vector  $\mathbf{r}_{G}^{4}$  from wheel frame to ground universal frame

$$\mathbf{r}_{G}^{U} = \mathbf{r}_{4}^{U} + \mathbf{r}_{G}^{4}$$

$$= \begin{bmatrix} \left(\mathbf{r}_{4}^{U}\right)_{x_{U}} \\ \left(\mathbf{r}_{4}^{U}\right)_{y_{U}} \\ \left(\mathbf{r}_{4}^{U}\right)_{z_{U}} \end{bmatrix} + \begin{bmatrix} \left(\mathbf{r}_{G}^{4}\right)_{x_{U}} \\ -\left(\mathbf{r}_{4}^{U}\right)_{y_{U}} \\ -\left(\mathbf{r}_{4}^{U}\right)_{z_{U}} \end{bmatrix} = \begin{bmatrix} \left(\mathbf{r}_{4}^{U}\right)_{x_{U}} + \left(\mathbf{r}_{G}^{4}\right)_{x_{U}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$
(2.62)

The magnitude of the position vector  $r_G^4$ , from wheel frame to ground frame, is definitely equal to the magnitude of position vector  $r_4^G$ , from ground frame to wheel frame, but in opposite direction

$$\mathbf{r}_{\mathrm{G}}^{4\mathrm{RK}} = -\mathbf{r}_{4\mathrm{RK}}^{\mathrm{G}} \tag{2.63}$$

The input system  $(r_4^G)_{x_U}$ , is the vertical altitude from ground universal frame to wheel frame (end-effector). The position vector  $r_4^U$  is a function of joint variables that formulate the rover posture and decide the location of ground contact point.

Finally, the combination of identity orientation matrix and the computed position vector defines the homogenous transformation from the platform universal frame  $O_U$  to the ground universal frame  $O_G$ 

$$\mathbf{A}_{\mathrm{G}}^{\mathrm{U}} = \begin{bmatrix} 1 & 0 & 0 & \left(\mathbf{r}_{4}^{\mathrm{U}}\right)_{\mathbf{x}_{\mathrm{U}}} + \left(\mathbf{r}_{\mathrm{G}}^{4}\right)_{\mathbf{x}_{\mathrm{U}}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(2.64)



**Figure 2.30.** Platform universal frame, wheel universal frame, and ground universal frame are contingent frames for being having the same orientations.

## 2.10.2. Universal wheel frame

After finding  $4 \times 4$  homogeneous transformation matrix of Roll, Pitch and Yaw, and **A**'s from platform to end-effectors, we can find the pose of the wheel frame with respect to the universals;  $O_U$ ,  $O_W$ ,  $O_G$ .



Figure 2.31. Completed transform graph.

Since the platform universal, wheel universal, and ground universal have the same and fixed axes, the orientation matrix of wheel frame with respect to any one of these universal frames can be the same as

$$\mathbf{R}_{4}^{U} = \mathbf{R}_{4}^{W} = \mathbf{R}_{4}^{G} \tag{2.65}$$

The pose of wheel end-effector with respect to a certain frame is simply the study of orientations and the position vector. The orientations are defined in three angles values ( $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ ) respectively about ( $x_U$ ,  $y_U$ ,  $z_U$  axes), ( $x_W$ ,  $y_W$ ,  $z_W$  axes), or ( $x_G$ ,  $y_G$ ,  $z_G$  axes). The position vector is defined as  $r_4^0$  as shown in the following Figure 2.32:



Figure 2.32. End-effector pose

The solutions for angles are specified here using roll, pitch, and yaw approach. The roll is a rotation about y-axis by  $\alpha_2$ , the pitch is a rotation about z-axis by  $\alpha_3$ , and yaw is a rotation about x-axis by  $\alpha_3$ 

$$T_4^W = \operatorname{Rot}(y_W, \alpha_2) \operatorname{Rot}(z_W, \alpha_3) \operatorname{Rot}(x_W, \alpha_1)$$

n x	0 x	a x	p <sub>x</sub> ]	[	- cα <sub>2</sub>	0	$s\alpha_2$	0	$\int c \alpha_3$	$-s\alpha_3$	0	0	[1	0	0	0]	
ny	Оу	аy	p <sub>y</sub>	_	0	1	0	0	sα <sub>3</sub>	$c\alpha_3$	0	0	0	$c \alpha_1$	$-s\alpha_1$	0	(2.66)
nz	0 z	az	p <sub>z</sub>	_	$-s\alpha_2$	0	$c \alpha_2$	0	0	0	1	0	0	$s\alpha_1$	$c \alpha_1$	0	
0	0	0	1	l	0	0	0	1	0	0	0	1	0	0	0	1	

The roll, pitch and yaw approach has no translational vector, thus  $p_x$ ,  $p_y$  and  $p_z$  must equal zero. In any way, if we pre-multiply Equation 2.66 by  $Rot(y_W, \alpha_2)^{-1}$  we obtain

$$\operatorname{Rot}(\mathbf{y}_{\mathrm{W}}, \alpha_{2})^{-1} \operatorname{T}_{4}^{\mathrm{W}} = \operatorname{Rot}(\mathbf{z}_{\mathrm{W}}, \alpha_{3}) \operatorname{Rot}(\mathbf{x}_{\mathrm{W}}, \alpha_{1})$$
(2.67)

The left hand side is

$$LHS = \begin{bmatrix} c\alpha_{2}n_{x} - s\alpha_{2}n_{z} & c\alpha_{2}o_{x} - s\alpha_{2}o_{z} & c\alpha_{2}a_{x} - s\alpha_{2}a_{z} & c\alpha_{2}p_{x} - s\alpha_{2}p_{z} \\ n_{y} & o_{y} & a_{y} & p_{y} \\ s\alpha_{2}n_{x} + c\alpha_{2}n_{z} & s\alpha_{2}o_{x} + c\alpha_{2}o_{z} & s\alpha_{2}a_{x} + c\alpha_{2}a_{z} & s\alpha_{2}p_{x} + c\alpha_{2}p_{z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(2.68)

The right hand side is

$$RHS = \begin{bmatrix} c\alpha_{3} & -s\alpha_{3}c\alpha_{1} & s\alpha_{3}s\alpha_{1} & 0\\ s\alpha_{3} & c\alpha_{3}c\alpha_{1} & -c\alpha_{3}s\alpha_{1} & 0\\ 0 & s\alpha_{1} & c\alpha_{1} & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(2.69)

The third row, first column element on the right hand side is zero. Equating this to the element on the left hand at the same location we obtain

$$\mathbf{s}\alpha_2\mathbf{n}_{\mathbf{x}} + \mathbf{c}\alpha_2\mathbf{n}_{\mathbf{z}} = \mathbf{0} \tag{2.70}$$

then,

$$\alpha_2 = \operatorname{atan} 2(-n_z, n_x) \tag{2.71}$$

Equating the 1,1 and 1,2 elements from left and right hand sides we obtain

$$\mathbf{c}\alpha_2\mathbf{n}_{\mathbf{x}} - \mathbf{s}\alpha_2\mathbf{n}_{\mathbf{z}} = \mathbf{c}\alpha_3 \tag{2.72}$$

$$\mathbf{n}_{y} = \mathbf{s}\boldsymbol{\alpha}_{3} \tag{2.73}$$

then,

$$\alpha_{3} = \operatorname{atan} 2(n_{y}, c\alpha_{2}n_{x} - s\alpha_{2}n_{z})$$
(2.74)

Equating the 3,2 and 3,3 elements from left and right hand sides we obtain

$$\mathbf{c}\alpha_1 = \mathbf{s}\alpha_2\mathbf{a}_x + \mathbf{c}\alpha_2\mathbf{a}_z \tag{2.75}$$

$$\mathbf{s}\alpha_1 = \mathbf{s}\alpha_2\mathbf{o}_{\mathbf{x}} + \mathbf{c}\alpha_2\mathbf{o}_{\mathbf{z}} \tag{2.76}$$
then

$$\alpha_1 = \operatorname{atan} 2(\operatorname{s}\alpha_2 \operatorname{o}_x + \operatorname{c}\alpha_2 \operatorname{o}_z, \operatorname{s}\alpha_2 \operatorname{a}_x + \operatorname{c}\alpha_2 \operatorname{a}_z)$$
(2.77)

6 multiplies, 3 additions, and 3 transcendental function calls.

### 2.10.3. Surface geometries

This work takes the shape of surface geometry traversed by the rover into account; (flat surface, step surface, inclined surface, sinusoidal surface, random surface). The surface frame is an orientation axes with respect universal ground frame  $O_G$  setting up as follows:  $x_S$  axis is normal to surface;  $y_S$ axis is in tangent direction of contact surface; and  $z_S$  axis is directed perpendicularly to  $x_S$ - $y_S$  plane. See Figure 2.33



(a) Flat surface and inclined surface



(b) Sinusoidal surface.

Figure 2.33. Surface frame.

The homogenous matrix of surface frame with respect to universal ground frame, can be given by Roll, Pitch, and Yaw,

$$R_{S}^{G} = RPY(\beta_{2},\beta_{3},\beta_{1}) = Rot(y_{G},\beta_{2})Rot(z_{G},\beta_{3})Rot(x_{G},\beta_{1})$$

$$= \begin{bmatrix} C\beta_{2} & 0 & S\beta_{2} \\ 0 & 1 & 0 \\ -S\beta_{2} & 0 & S\beta_{2} \end{bmatrix} \cdot \begin{bmatrix} C\beta_{3} & -S\beta_{3} & 0 \\ S\beta_{3} & C\beta_{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\beta_{1} & -S\beta_{1} \\ 0 & S\beta_{1} & C\beta_{1} \end{bmatrix}$$

$$= \begin{bmatrix} c\beta_{2}c\beta_{3} & -c\beta_{2}s\beta_{3}c\beta_{1} + s\beta_{2}s\beta_{1} & c\beta_{2}s\beta_{3}s\beta_{1} + s\beta_{2}c\beta_{1} \\ s\beta_{3} & c\beta_{3}c\beta_{1} & -c\beta_{3}s\beta_{1} \\ -s\beta_{2}c\beta_{3} & s\beta_{2}s\beta_{3}c\beta_{1} + c\beta_{2}s\beta_{1} & s\beta_{2}s\beta_{3}c\beta_{1} + c\beta_{2}s\beta_{1} \end{bmatrix}$$
(2.78)



The touching point occurs between wheel and surface, and its position vector with respect universal ground frame is given by

$$\mathbf{r}_{\mathrm{S}}^{\mathrm{G}} = \begin{bmatrix} \left(\mathbf{r}_{4}^{\mathrm{G}}\right)_{\mathbf{x}_{\mathrm{U}}} \\ -\left(\mathbf{r}_{4}^{\mathrm{U}}\right)_{\mathbf{y}_{\mathrm{U}}} \\ -\left(\mathbf{r}_{4}^{\mathrm{U}}\right)_{\mathbf{z}_{\mathrm{U}}} \end{bmatrix}$$
(2.79)

Because the surface frame  $O_s$ , universal wheel frame  $O_w$ , and endeffector frame  $O_4$  are situated at the same point, those frames have the same position vector as obtained in equation (2.80)

$$\mathbf{r}_{\mathrm{S}}^{\mathrm{G}} = \mathbf{r}_{\mathrm{w}}^{\mathrm{G}} = \mathbf{r}_{\mathrm{4}}^{\mathrm{G}} \tag{2.80}$$

The homogeneous transformations starting from wheel end-effector frame  $O_4$  and surface frame  $O_8$ :

 $\mathbf{R}_{4}^{\mathrm{U}} \cdot \mathbf{R}_{5}^{\mathrm{s}} \cdot \mathbf{R}_{6}^{\mathrm{s}} = \mathbf{R}_{6}^{\mathrm{U}} = \mathbf{I}$  (2.81)

$$\mathbf{R}_{\mathrm{S}}^{4} \cdot \mathbf{R}_{\mathrm{G}}^{\mathrm{S}} = \mathbf{R}_{\mathrm{U}}^{4} \cdot \mathbf{I} \tag{2.82}$$

$$\mathbf{R}_{\mathrm{S}}^{4} = \mathbf{R}_{\mathrm{U}}^{4} \cdot \mathbf{I} \cdot \mathbf{R}_{\mathrm{S}}^{\mathrm{G}} \tag{2.83}$$

$$\mathbf{R}_{\mathrm{S}}^{4} = \mathbf{R}_{\mathrm{U}}^{4} \cdot \mathbf{R}_{\mathrm{S}}^{\mathrm{G}} \tag{2.84}$$

# **Chapter Three**

## 3. Computational Dynamics

The approach which thesis will recommend and base in dynamic calculations for computing the equation of motion is Walker and Orin [61, 62] application "recursive Newton-Euler formulation"; because of its explicit notations and less execution time. They formulated the equations of motion in explicit form in comparison with others; simply it will yield a set of recursive equations, which can be applied to the links sequentially to compute the generalized forces referenced in their own coordinates in a short period of time and in on-line control.

#### 3.1. Dynamic equations of motion

The second order nonlinear system equations of motion for the manipulator, with n joints and n+1 links, are generated generally from inertia, friction, Coriolis and Centrifugal, and gravity as shown in equation 3.1.

$$J(q)\ddot{q} + C\dot{q} + F(\dot{q}) + G(q) = Q$$
(3.1)

Where,

- q  $n \times 1$  vector of generalized joint coordinates,
- $\dot{q}$  n × 1 vector of joint velocities,
- $\ddot{q}$  n × 1 vector of joint accelerations,
- J(q) n × 4 symmetric joint space inertia matrix, or manipulator inertia tensor,
- C  $n \times 4$  viscous friction matrix,
- $F(\dot{q})$  n × 1 vector defining Coriolis and Centrifugal forces,
- G(q) n × 1 vector defining the gravity terms,
- $Q = n \times 1$  vector defining the input generalized forces.

The manipulator joint space inertia and gravitational force are dependent on the manipulator configurations,  $\mathbf{q}$ , So that they are considered as function of variable joints. The Coriolis and Centrifugal forces are considered as functions of joint velocity,  $\dot{\mathbf{q}}$  [49].

The input generalized forces  $Q_i$  are forces and moments on the joint i exerted by the actuator and by consequences of normal force, friction surface, and frictional moments exerted by surface on wheel end-effector.

#### 3.2. Output generalized coordinates

The environmental inputs acting on the manipulator system are represented in forces and torques exerted on an end-effectors. The outputs of the system are represented in link's positions, velocities, and accelerations. In other words, the forces and torques cause the accelerations and velocities, irrespective of linear or angular forms.

Dynamics conduct two problems: forward dynamics recursion and backward dynamics recursion. The forward dynamics studies the trajectory of end-effectors with regard to the forces and torques that intuitively cause the motion. The inverse dynamic computes the forces and torques required to cause motion. See Figure 3.1.



Figure 3.1. Dynamics propagations

Figure 3.2 shows link i is connected to its two adjacent links; i.e. link i-1 by joint i and also link i+1 by joint i+1. As well as, it shows the force and

moment ( $F_i$  and  $T_i$ ) which act directly on end-terminal of link i by link i-1; and force and moment ( $F_{i+1}$  and  $T_{i+1}$ ) which act directly on another end-terminal of link i+1 by link i. Furthermore, the inertia force and moment ( $f_i$ ,  $\tau_i$ ) act directly on the center of mass of link i.



Figure 3.2. Recursive Newton-Euler Formulation notations on the base of the standard of the DH convention.

Moreover, Figure 3.2 also shows that the rover motion is referred with respect to the universal frame  $O_U(x_U, y_U, z_U)$ . In other words, the coordinate frames and equations of motion of each link are expressed to the universal frame. The universal frame is chosen at the center of platform forming  $y_U-z_U$  horizontal plane, which is parallel to ground plane; and  $x_U$  axis is normal to the ground plane directed upward. The coordinates frames are assigned at joints by utilizing from DH convention as explained in Chapter 2.

#### 3.3. Newton-Euler Recursive Relations

The computations for determining the equations of motion will be complicated if the calculations are considered with respect to the fixed base frame [55], because the inertia matrix I<sub>i</sub> depends on the orientation of link i. The efficient solution is to consider the dynamic and kinematics of each link expressed to its own coordinates frame [49]. Therefore, the equation of motion of each link is expressed to its own coordinate frame instead of making it expressed to the base frame following the notations made by Walker and Orin [61], See Appendix D. The basic idea behind the Newton Euler recursive formulation is broken down into two steps, i.e. forward and backward recursion.

#### 3.3.1. Forward recursion

This approach transforms the output generalized velocities and accelerations from the universal frame to the end-effector frame, link by link in iterative techniques, using the relationships of moving coordinate systems [49].

The generalized coordinates (links positions, velocities, and accelerations) starting from universal frame  $O_U$  and ending at end-effector  $O_4$  frame can respectively and briefly be symbolized

$$q_{i+1} = \begin{cases} \begin{bmatrix} \psi & \phi & \theta \end{bmatrix}^{T} , i = U \\ \begin{bmatrix} 0 & 0 & \theta_{i+1} \end{bmatrix}^{T} , 0 \le i \le n-1 \end{cases}$$
(3.2)

$$\dot{\mathbf{q}}_{i+1} = \begin{cases} \begin{bmatrix} \dot{\psi} & \dot{\phi} & \dot{\theta} \end{bmatrix}^{\mathrm{T}}, i = \mathrm{U} \\ \begin{bmatrix} 0 & 0 & \dot{\theta}_{i+1} \end{bmatrix}^{\mathrm{T}}, 0 \le i \le n-1 \end{cases}$$
(3.3)

$$\ddot{\boldsymbol{q}}_{i+1} = \begin{cases} \begin{bmatrix} \ddot{\psi} & \ddot{\phi} & \ddot{\theta} \end{bmatrix}^{T}, i = \boldsymbol{U} \\ \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \ddot{\theta}_{i+1} \end{bmatrix}^{T}, \boldsymbol{0} \le i \le n-1 \end{cases}$$
(3.4)

For i = U,  $q_0$  describes the platform orientation with respect to universal frame. The platform of the rover is not bolted on any stationary point anymore, and its attitude is under the influence of the ground heights and the configurations of the four legged manipulators. As explained in chapter 1, the

attitude angles of the platform are evaluated with respect to the universal frame, i.e. roll ( $\phi$ ), pitch ( $\theta$ ), and yaw ( $\psi$ ) angles respectively rotate about y<sub>U</sub>, z<sub>U</sub>, and x<sub>U</sub> axes. These attitude angles are forming 3×1 vectors filled up with the generalized position, velocity, and acceleration coordinates of the first iterative step, respectively, as follow

$$\mathbf{q}_0 = \begin{bmatrix} \boldsymbol{\psi} & \boldsymbol{\phi} & \boldsymbol{\theta} \end{bmatrix}^{\mathrm{T}} \tag{3.5}$$

$$\dot{\mathbf{q}}_{0} = \begin{bmatrix} \dot{\psi} & \dot{\phi} & \dot{\theta} \end{bmatrix}^{\mathrm{T}}$$
(3.6)

and,

$$\ddot{\mathbf{q}}_{0} = \begin{bmatrix} \ddot{\boldsymbol{\psi}} & \ddot{\boldsymbol{\phi}} & \ddot{\boldsymbol{\theta}} \end{bmatrix}^{\mathrm{T}}$$
(3.7)

The homogeneous transformation matrix of base frame  $O_0$  with respect to universal frame  $O_U$  is

$$\mathbf{R}_{0}^{\mathrm{U}} = \begin{bmatrix} c\phi c\theta & -c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi + s\phi c\psi \\ s\theta & c\theta c\psi & -c\theta s\psi \\ -s\phi c\theta & s\phi s\theta c\psi + c\phi s\psi & -s\phi s\theta s\psi + c\phi c\psi \end{bmatrix}$$
(3.8)

And the homogeneous matrix of the universal frame  $O_U$  with respect to the base frame  $O_0$  is given as the inverse of the above matrix

$$\mathbf{R}_{\mathrm{U}}^{0} = \begin{bmatrix} \mathbf{c}\phi\mathbf{c}\theta & \mathbf{s}\theta & -\mathbf{s}\phi\mathbf{c}\theta \\ -\mathbf{c}\phi\mathbf{s}\theta\mathbf{c}\psi + \mathbf{s}\phi\mathbf{s}\psi & \mathbf{c}\theta\mathbf{c}\psi & \mathbf{s}\phi\mathbf{s}\theta\mathbf{c}\psi + \mathbf{c}\phi\mathbf{s}\psi \\ \mathbf{c}\phi\mathbf{s}\theta\mathbf{s}\psi + \mathbf{s}\phi\mathbf{c}\psi & -\mathbf{c}\theta\mathbf{s}\psi & -\mathbf{s}\phi\mathbf{s}\theta\mathbf{s}\psi + \mathbf{c}\phi\mathbf{c}\psi \end{bmatrix}$$
(3.9)

According to initial values of the system, the angular velocity and acceleration of universal frame  $O_U$  with respect to base frame  $O_0$  expressed in universal frame itself can be, respectively, given as

$$\boldsymbol{\omega}_{\mathrm{U}}^{\mathrm{U}} = \begin{bmatrix} \left(\boldsymbol{\omega}_{\mathrm{U}}^{\mathrm{U}}\right)_{\mathbf{x}_{\mathrm{U}}} & \left(\boldsymbol{\omega}_{\mathrm{U}}^{\mathrm{U}}\right)_{\mathbf{y}_{\mathrm{U}}} & \left(\boldsymbol{\omega}_{\mathrm{U}}^{\mathrm{U}}\right)_{\mathbf{z}_{\mathrm{U}}} \end{bmatrix}^{\mathrm{T}}$$
(3.10)

and,

$$\dot{\omega}_{U}^{U} = \left[ \left( \dot{\omega}_{U}^{U} \right)_{x_{U}} \quad \left( \dot{\omega}_{U}^{U} \right)_{y_{U}} \quad \left( \dot{\omega}_{U}^{U} \right)_{z_{U}} \right]^{T}$$
(3.11)

Moreover, the linear velocity and acceleration of universal frame  $O_U$  with respect to base frame  $O_0$  expressed in universal frame itself can be, respectively, given as

$$\mathbf{v}_{U}^{U} = \mathbf{R}_{0}^{U} \cdot \begin{bmatrix} 0 & \frac{\mathbf{a}_{4}\dot{\theta}_{4R} + \mathbf{a}_{4}\dot{\theta}_{4L}}{2} & 0 \end{bmatrix}^{\mathrm{T}}$$
(3.12)

and,

$$\dot{\mathbf{v}}_{U}^{U} = \begin{bmatrix} -g & 0 & 0 \end{bmatrix}^{T} + \mathbf{R}_{0}^{U} \cdot \begin{bmatrix} 0 & \frac{\mathbf{a}_{4}\ddot{\boldsymbol{\theta}}_{4R} + \mathbf{a}_{4}\ddot{\boldsymbol{\theta}}_{4L}}{2} & 0 \end{bmatrix}^{T}$$
(3.13)

g is a gravity acceleration pointing downward of  $x_U$ -axis, and its magnitude is equal to 9.81 and 3.63 m/s<sup>2</sup> according to sea level of the earth and Mars surface.  $\dot{v}_U^U$  and  $\dot{v}_U^U$  vectors are treated in projection onto inertial coordinate system referenced to the universal frame  $O_U$ .

The position vector from the universal frame  $O_U$  to base frame  $O_0$  expressed in base frame is

$$\mathbf{r}_0^0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \tag{3.14}$$

Following the computational algorithm as in Appendix C, The angular velocity propagation for the base link when i = U

$$\omega_{0}^{0} = \mathbf{R}_{U}^{0} \left( \omega_{U}^{U} + \dot{\mathbf{q}}_{0} \right)$$

$$= \begin{bmatrix} c\phi c\theta & s\theta & -s\phi c\theta \\ -c\phi s\theta c\psi + s\phi s\psi & c\theta c\psi & s\phi s\theta c\psi + c\phi s\psi \\ c\phi s\theta s\psi + s\phi c\psi & -c\theta s\psi & -s\phi s\theta s\psi + c\phi c\psi \end{bmatrix} \cdot \left[ \omega_{U}^{U} + \begin{vmatrix} \dot{\psi} \\ \dot{\phi} \\ \dot{\theta} \end{vmatrix} \right]$$

$$(3.15)$$

The angular acceleration propagation for the base frame when i = U is

$$\dot{\omega}_{0}^{0} = \mathbf{R}_{U}^{0} \left( \dot{\omega}_{U}^{U} + \ddot{\mathbf{q}}_{0} + \omega_{U}^{U} \times \dot{\mathbf{q}}_{0} \right)$$

$$= \begin{bmatrix} c\phi c\theta & s\theta & -s\phi c\theta \\ -c\phi s\theta c\psi + s\phi s\psi & c\theta c\psi & s\phi s\theta c\psi + c\phi s\psi \\ c\phi s\theta s\psi + s\phi c\psi & -c\theta s\psi & -s\phi s\theta s\psi + c\phi c\psi \end{bmatrix} \cdot \left( \dot{\omega}_{U}^{U} + \begin{bmatrix} \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\theta} \end{bmatrix} + \omega_{U}^{U} \times \begin{bmatrix} \dot{\psi} \\ \dot{\phi} \\ \dot{\theta} \end{bmatrix} \right)$$

$$(3.16)$$

The linear acceleration propagation for the base link when i = U is

$$\dot{\mathbf{v}}_{0}^{0} = \dot{\boldsymbol{\omega}}_{0}^{0} \times \mathbf{r}_{0}^{0} + \boldsymbol{\omega}_{0}^{0} \times \left(\boldsymbol{\omega}_{0}^{0} \times \mathbf{r}_{0}^{0}\right) + \mathbf{R}_{\mathrm{U}}^{0} \dot{\mathbf{v}}_{\mathrm{U}}^{\mathrm{U}}$$

$$= \dot{\boldsymbol{\omega}}_{0}^{0} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \boldsymbol{\omega}_{0}^{0} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} c\phi c\theta & s\theta & -s\phi c\theta \\ -c\phi s\theta c\psi + s\phi s\psi & c\theta c\psi & s\phi s\theta c\psi + c\phi s\psi \\ c\phi s\theta s\psi + s\phi c\psi & -c\theta s\psi & -s\phi s\theta s\psi + c\phi c\psi \end{bmatrix} \cdot \begin{bmatrix} -\mathbf{g} \\ \frac{\mathbf{r}\ddot{\theta}_{4\mathrm{R}} - \mathbf{r}\ddot{\theta}_{4\mathrm{L}}}{2} \\ \mathbf{0} \end{bmatrix} (3.17)$$

The velocity and acceleration of the platform center of mass, when i = 0, are computed respectively as follows:

$$\mathbf{v}_{c,0}^{0} = \omega_{0}^{0} \times \mathbf{r}_{c,0}^{0} + \mathbf{v}_{0}^{0}$$

$$= \omega_{0}^{0} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \mathbf{v}_{0}^{0}$$
(3.18)

and,

$$\dot{\mathbf{v}}_{c,0}^{0} = \dot{\boldsymbol{\omega}}_{0}^{0} \times \mathbf{r}_{c,0}^{0} + \boldsymbol{\omega}_{0}^{0} \times \left(\boldsymbol{\omega}_{0}^{0} \times \mathbf{r}_{c,0}^{0}\right) + \dot{\mathbf{v}}_{0}^{0}$$

$$= \dot{\boldsymbol{\omega}}_{0}^{0} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \boldsymbol{\omega}_{0}^{0} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \dot{\mathbf{v}}_{0}^{0} \qquad (3.19)$$

Once the velocities and accelerations of the platform center of mass are computed, the inertia force and moment acting on the platform center of mass can be computed. Assuming the viscous damping friction is negligible, the total external force acting on each link center of mass is given by the Newton's second law, and whilst the moment acting on each link center of mass is given by Euler's equation. Newton-Euler formulation for the platform center of mass can be presented as:

$$f_0^0 = m_0 \dot{v}_{c,0}^0 \tag{3.20}$$

and,

$$\boldsymbol{\tau}_{0}^{0} = \boldsymbol{I}_{0} \dot{\boldsymbol{\omega}}_{0}^{0} + \boldsymbol{\omega}_{0}^{0} \times \left( \boldsymbol{I}_{0} \boldsymbol{\omega}_{0}^{0} \right)$$
(3.21)

For  $0 \le i \le n-1$ ,  $q_{i+1}$  describes the motion of the manipulator starting from link 1 ending at link 4. **n** represents the number of joints of the manipulator. Our mobile robot employs four revolute joints for each manipulator; no any prismatic joint is employed. Thus, the notations of output generalized position coordinates will match the joint angles. Each entry inside  $q_{i+1}$  is composed of a 3×1 vector

$$q_{1} = \begin{bmatrix} 0 & 0 & \theta_{1} \end{bmatrix}^{T} 
q_{2} = \begin{bmatrix} 0 & 0 & \theta_{2} \end{bmatrix}^{T} 
q_{3} = \begin{bmatrix} 0 & 0 & \theta_{3} \end{bmatrix}^{T} 
q_{4} = \begin{bmatrix} 0 & 0 & \theta_{4} \end{bmatrix}^{T}$$
(3.22)

As well as, for  $0 \le i \le 3$ , the generalized joint velocities and joint accelerations are, respectively, as shown bellow:

$$\dot{\mathbf{q}}_{i+1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dot{\mathbf{\theta}}_{i+1} \end{bmatrix}^{\mathrm{T}}$$
(3.23)

$$\ddot{\mathbf{q}}_{i+1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \ddot{\mathbf{\theta}}_{i+1} \end{bmatrix}^{\mathrm{T}}$$
(3.24)

Now completing the algorithm as shown in Appendix C and D, The angular velocity propagation for link 1 when i = 0

$$\omega_{1}^{l} = \mathbf{R}_{0}^{1} \begin{pmatrix} \omega_{0}^{0} + \dot{\mathbf{q}}_{1} \end{pmatrix}$$

$$= \begin{bmatrix} \mathbf{C}_{1} & \mathbf{S}_{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -1 \\ -\mathbf{S}_{1} & \mathbf{C}_{1} & \mathbf{0} \end{bmatrix} \cdot \begin{pmatrix} \omega_{0}^{0} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{\theta}}_{1} \end{bmatrix} \end{pmatrix}$$
(3.25)

The angular velocity propagation for link 2 when i = 1

$$\omega_{2}^{2} = \mathbf{R}_{1}^{2} \begin{pmatrix} \omega_{1}^{1} + \dot{\mathbf{q}}_{2} \end{pmatrix}$$

$$= \begin{bmatrix} \mathbf{C}_{2} & \mathbf{S}_{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{S}_{2} & -\mathbf{C}_{2} & \mathbf{0} \end{bmatrix} \cdot \begin{pmatrix} \omega_{1}^{1} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{\theta}}_{2} \end{bmatrix} \end{pmatrix}$$
(3.26)

The angular velocity propagation for link 3 when i = 2

$$\omega_{3}^{3} = \mathbb{R}_{2}^{3} \left( \omega_{2}^{2} + \dot{q}_{3} \right)$$

$$= \begin{bmatrix} -\mathbb{C}_{3} & -\mathbb{S}_{3} & 0 \\ \mathbb{S}_{3} & -\mathbb{C}_{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left( \omega_{2}^{2} + \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{3} \end{bmatrix} \right)$$
(3.27)

The angular velocity propagation for link 4 when i = 3

Now, the angular accelerations propagation starting from link 1 and ending at end effector link can be computed using the formula

$$\dot{\omega}_{i+1}^{i+1} = \mathbf{R}_{i}^{i+1} \left( \dot{\omega}_{i}^{i} + \ddot{\mathbf{q}}_{i+1} + \omega_{i}^{i} \times \dot{\mathbf{q}}_{i+1} \right)$$
(3.29)

The angular acceleration propagation for link 1 when i = 0

$$\dot{\omega}_{1}^{1} = \mathbf{R}_{0}^{1} \left( \dot{\omega}_{0}^{0} + \ddot{\mathbf{q}}_{1} + \omega_{0}^{0} \times \dot{\mathbf{q}}_{1} \right)$$

$$= \begin{bmatrix} \mathbf{C}_{1} & \mathbf{S}_{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -1 \\ -\mathbf{S}_{1} & \mathbf{C}_{1} & \mathbf{0} \end{bmatrix} \left( \dot{\omega}_{0}^{0} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \ddot{\mathbf{\theta}}_{1} \end{bmatrix} + \omega_{0}^{0} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{\theta}}_{1} \end{bmatrix} \right)$$
(3.30)

The angular acceleration propagation for link 2 when i = 1

$$\dot{\omega}_{2}^{2} = \mathbf{R}_{1}^{2} \left( \dot{\omega}_{1}^{1} + \ddot{\mathbf{q}}_{2} + \omega_{1}^{1} \times \dot{\mathbf{q}}_{2} \right)$$

$$= \begin{bmatrix} \mathbf{C}_{2} & \mathbf{S}_{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{S}_{2} & -\mathbf{C}_{2} & \mathbf{0} \end{bmatrix} \left( \dot{\omega}_{1}^{1} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \ddot{\mathbf{\theta}}_{2} \end{bmatrix} + \omega_{1}^{1} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{\theta}}_{2} \end{bmatrix} \right)$$
(3.31)

The angular acceleration propagation for link 3 when i = 2

$$\dot{\omega}_{3}^{3} = \mathbf{R}_{2}^{3} \left( \dot{\omega}_{2}^{2} + \ddot{\mathbf{q}}_{3} + \omega_{2}^{2} \times \dot{\mathbf{q}}_{3} \right)$$

$$= \begin{bmatrix} -\mathbf{C}_{3} & -\mathbf{S}_{3} & \mathbf{0} \\ \mathbf{S}_{3} & -\mathbf{C}_{3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \left( \dot{\omega}_{2}^{2} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \ddot{\mathbf{\theta}}_{3} \end{bmatrix} + \omega_{2}^{2} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{\theta}}_{3} \end{bmatrix} \right)$$
(3.32)

The angular acceleration propagation for link 4 when i = 3

$$\dot{\omega}_{4}^{4} = \mathbf{R}_{3}^{4} \left( \dot{\omega}_{3}^{3} + \ddot{\mathbf{q}}_{4} + \omega_{3}^{3} \times \dot{\mathbf{q}}_{4} \right)$$

$$= \begin{bmatrix} \mathbf{C}_{4} & \mathbf{S}_{4} & \mathbf{0} \\ -\mathbf{S}_{4} & \mathbf{C}_{4} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \left( \dot{\omega}_{3}^{3} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \ddot{\mathbf{\theta}}_{4} \end{bmatrix} + \omega_{3}^{3} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \dot{\mathbf{\theta}}_{4} \end{bmatrix} \right)$$
(3.33)

And now, the linear acceleration propagations starting from link 1 to link 4 are computed by following this algorithm

$$\dot{\mathbf{v}}_{i+1}^{i+1} = \dot{\boldsymbol{\omega}}_{i+1}^{i+1} \times \mathbf{r}_{i+1}^{i+1} + \boldsymbol{\omega}_{i+1}^{i+1} \times \left(\boldsymbol{\omega}_{i+1}^{i+1} \times \mathbf{r}_{i+1}^{i+1}\right) + \mathbf{R}_{i}^{i+1} \dot{\mathbf{v}}_{i}^{i}$$
(3.34)

The linear acceleration propagation for link 1 when i = 0 is

$$\dot{\mathbf{v}}_{1}^{1} = \dot{\omega}_{1}^{1} \times \mathbf{r}_{1}^{1} + \omega_{1}^{1} \times \left(\omega_{1}^{1} \times \mathbf{r}_{1}^{1}\right) + \mathbf{R}_{0}^{1} \dot{\mathbf{v}}_{0}^{0}$$

$$= \dot{\omega}_{1}^{1} \times \begin{bmatrix} \mathbf{0} \\ -\mathbf{d}_{1} \\ \mathbf{0} \end{bmatrix} + \omega_{1}^{1} \times \begin{bmatrix} \mathbf{0} \\ -\mathbf{d}_{1} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{1} & \mathbf{S}_{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -1 \\ -\mathbf{S}_{1} & \mathbf{C}_{1} & \mathbf{0} \end{bmatrix} \cdot \dot{\mathbf{v}}_{0}^{0}$$
(3.35)

The linear acceleration propagation for link 2 when i = 1 is

107

$$\dot{\mathbf{v}}_{2}^{2} = \dot{\omega}_{2}^{2} \times \mathbf{r}_{2}^{2} + \omega_{2}^{2} \times \left(\omega_{2}^{2} \times \mathbf{r}_{2}^{2}\right) + \mathbf{R}_{1}^{2} \dot{\mathbf{v}}_{1}^{1}$$

$$= \dot{\omega}_{2}^{2} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \omega_{2}^{2} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{2} & \mathbf{S}_{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \mathbf{S}_{2} & -\mathbf{C}_{2} & \mathbf{0} \end{bmatrix} \cdot \dot{\mathbf{v}}_{1}^{1}$$
(3.36)

The linear acceleration propagation for link 3 when i = 2 is

$$\dot{\mathbf{v}}_{3}^{3} = \dot{\omega}_{3}^{3} \times \mathbf{r}_{3}^{3} + \omega_{3}^{3} \times \left(\omega_{3}^{3} \times \mathbf{r}_{3}^{3}\right) + \mathbf{R}_{2}^{3} \dot{\mathbf{v}}_{2}^{2}$$

$$= \dot{\omega}_{3}^{3} \times \begin{bmatrix} \mathbf{a}_{3} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \omega_{3}^{3} \times \begin{bmatrix} \mathbf{a}_{3} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{C}_{3} & -\mathbf{S}_{3} & \mathbf{0} \\ \mathbf{S}_{3} & -\mathbf{C}_{3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \dot{\mathbf{v}}_{2}^{2}$$
(3.37)

The linear acceleration propagation for link 4 when i = 3 is

$$\dot{\mathbf{v}}_{4}^{4} = \dot{\omega}_{4}^{4} \times \mathbf{r}_{4}^{4} + \omega_{4}^{4} \times \left(\omega_{4}^{4} \times \mathbf{r}_{4}^{4}\right) + \mathbf{R}_{3}^{4} \dot{\mathbf{v}}_{3}^{3}$$

$$= \dot{\omega}_{4}^{4} \times \begin{bmatrix} \mathbf{a}_{4} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \omega_{4}^{4} \times \begin{bmatrix} \mathbf{a}_{4} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{4} & \mathbf{S}_{4} & \mathbf{0} \\ -\mathbf{S}_{4} & \mathbf{C}_{4} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \dot{\mathbf{v}}_{3}^{3}$$
(3.38)

The velocity and acceleration of the center of mass of link i, starting from link 1 and ending at link 4, can be computed respectively as follows:

$$\dot{\mathbf{v}}_{c,i+1}^{i+1} = \dot{\omega}_{i+1}^{i+1} \times \mathbf{r}_{c,i+1}^{i+1} + \omega_{i+1}^{i+1} \times \left( \omega_{i+1}^{i+1} \times \mathbf{r}_{c,i+1}^{i+1} \right) + \dot{\mathbf{v}}_{i+1}^{i+1}$$
(3.39)

The linear acceleration of center of mass of link 1 when i = 0 is

$$\dot{\mathbf{v}}_{c,1}^{1} = \dot{\omega}_{1}^{1} \times \mathbf{r}_{c,1}^{1} + \omega_{1}^{1} \times \left(\omega_{1}^{1} \times \mathbf{r}_{c,1}^{1}\right) + \dot{\mathbf{v}}_{1}^{1}$$

$$= \dot{\omega}_{1}^{1} \times \begin{bmatrix} \mathbf{0} \\ \frac{\mathbf{d}_{1}}{2} \\ \mathbf{0} \end{bmatrix} + \omega_{1}^{1} \times \begin{bmatrix} \mathbf{0} \\ \frac{\mathbf{d}_{1}}{2} \\ \mathbf{0} \end{bmatrix} + \dot{\mathbf{v}}_{1}^{1} \qquad (3.40)$$

The linear acceleration of center of mass of link 2 when i = 1 is

$$\dot{\mathbf{v}}_{c,2}^{2} = \dot{\omega}_{2}^{2} \times \mathbf{r}_{c,2}^{2} + \omega_{2}^{2} \times \left(\omega_{2}^{2} \times \mathbf{r}_{c,2}^{2}\right) + \dot{\mathbf{v}}_{2}^{2}$$

$$= \dot{\omega}_{2}^{2} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \omega_{2}^{2} \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \dot{\mathbf{v}}_{2}^{2}$$
(3.41)

The linear acceleration of center of mass of link 3 when i = 2 is

$$\dot{\mathbf{v}}_{c,3}^{3} = \dot{\omega}_{3}^{3} \times \mathbf{r}_{c,3}^{3} + \omega_{3}^{3} \times \left(\omega_{3}^{3} \times \mathbf{r}_{c,3}^{3}\right) + \dot{\mathbf{v}}_{3}^{3}$$

$$= \dot{\omega}_{3}^{3} \times \begin{bmatrix} -\frac{\mathbf{a}_{3}}{2} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \omega_{3}^{3} \times \begin{bmatrix} -\frac{\mathbf{a}_{3}}{2} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \dot{\mathbf{v}}_{3}^{3} \qquad (3.42)$$

The linear acceleration of center of mass of link 4 when i = 3 is

$$\dot{\mathbf{v}}_{c,4}^{4} = \dot{\omega}_{4}^{4} \times \mathbf{r}_{c,4}^{4} + \omega_{4}^{4} \times \left(\omega_{4}^{4} \times \mathbf{r}_{c,4}^{4}\right) + \dot{\mathbf{v}}_{4}^{4}$$

$$= \dot{\omega}_{4}^{4} \times \begin{bmatrix} -\mathbf{a}_{4} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \omega_{4}^{4} \times \begin{bmatrix} -\mathbf{a}_{4} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \dot{\mathbf{v}}_{4}^{4} \qquad (3.43)$$

Starting from link 1 and ending at link 4, inertia force  $f_i^i$  acting on the center of mass of link i expressed in the frame  $O_i$  is given by

109

$$\mathbf{f}_{i+1}^{i+1} = \mathbf{m}_{i+1} \dot{\mathbf{v}}_{c,i+1}^{i+1} \tag{3.44}$$

Inertia force acting on the center of mass of link 1 when i = 0 is

$$f_{1}^{1} = m_{1} \dot{v}_{c,1}^{1}$$
(3.45)

Inertia force acting on the center of mass of link 2 when i = 1 is

$$\mathbf{f}_{2}^{2} = \mathbf{m}_{2} \dot{\mathbf{v}}_{c,2}^{2} \tag{3.46}$$

Inertia force acting on the center of mass of link 3 when i = 2 is

$$\mathbf{f}_{3}^{3} = \mathbf{m}_{3} \dot{\mathbf{v}}_{c,3}^{3} \tag{3.47}$$

Inertia force acting on the center of mass of link 4 when i = 3 is

$$f_4^4 = m_4 \dot{v}_{c,4}^4 \tag{3.48}$$

Starting from link 1 and ending at link 4, inertia torques acting on the center of masses of link i expressed in the frame  $O_i$  is given by are given by following this algorithm:

$$\boldsymbol{\tau}_{i+1}^{i+1} = \mathbf{I}_{i+1} \dot{\boldsymbol{\omega}}_{i+1}^{i+1} + \boldsymbol{\omega}_{i+1}^{i+1} \times \left( \mathbf{I}_{i+1} \boldsymbol{\omega}_{i+1}^{i+1} \right)$$
(3.49)

Inertia torque acting on the center of mass of link 1 when i = 0 is

$$\tau_{1}^{1} = I_{1}\dot{\omega}_{1}^{1} + \omega_{1}^{1} \times \left(I_{1}\omega_{1}^{1}\right)$$

$$= \frac{m_{1}(d_{1})^{2}}{12} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\omega}_{1}^{1} + \omega_{1}^{1} \times \left(\frac{m_{1}(d_{1})^{2}}{12} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \omega_{1}^{1}\right)$$
(3.50)

Inertia torque acting on the center of mass of link 2 when i = 1 is

$$\begin{aligned} \tau_{2}^{2} &= \mathbf{I}_{2}\dot{\omega}_{2}^{2} + \omega_{2}^{2} \times \left(\mathbf{I}_{2}\omega_{2}^{2}\right) \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \dot{\omega}_{2}^{2} + \omega_{2}^{2} \times \left( \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \omega_{2}^{2} \right) \end{aligned}$$
(3.51)

Inertia force acting on the center of mass of link 3 when i = 2 is

$$\tau_{3}^{3} = I_{3}\dot{\omega}_{3}^{3} + \omega_{3}^{3} \times \left(I_{3}\omega_{3}^{3}\right)$$

$$= \frac{m_{3}(a_{3})^{2}}{12} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \dot{\omega}_{3}^{3} + \omega_{3}^{3} \times \left(\frac{m_{3}(a_{3})^{2}}{12} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \omega_{3}^{3}\right)$$
(3.52)

Inertia force acting on the center of mass of link 4 when i = 3 is

$$\boldsymbol{\tau}_{4}^{4} = \mathbf{I}_{4} \dot{\boldsymbol{\omega}}_{4}^{4} + \boldsymbol{\omega}_{4}^{4} \times \left(\mathbf{I}_{4} \boldsymbol{\omega}_{4}^{4}\right) \tag{3.53}$$

#### 3.3.2. Backward recursion

Inverse dynamics approach computes the forces and torques recursively from link 4 to link 1. After computing the inertia forces and moments exerted on the center of masses of links, backward computational procedures can be followed by evaluating one a link at a time starting from the end-effector frame and ending at the base frame as shown in recursive form:

$$F_{i}^{i} = R_{i+1}^{i} F_{i+1}^{i+1} + f_{i}^{i}$$
(3.54)

$$T_{i}^{i} = R_{i+1}^{i} \left( T_{i+1}^{i+1} + \left( R_{i}^{i+1} r_{i}^{i} \right) \times F_{i+1}^{i+1} \right) + \left( R_{i}^{i} r_{i}^{i} + r_{c,i}^{i} \right) \times f_{i}^{i} + \tau_{i}^{i}$$
(3.55)

 $F_s^s$  and  $T_s^s$  are external force and moment exerted on the end-effector link in frame O<sub>4</sub> (x<sub>4</sub>, y<sub>4</sub>, z<sub>4</sub>). These can be defined in 3×1 vector as

$$\mathbf{F}_{\mathrm{S}}^{\mathrm{S}} = \begin{bmatrix} \mathbf{F}_{\mathrm{n}} & -\mathbf{F}_{\mathrm{f}} & \mathbf{0} \end{bmatrix}^{\mathrm{T}}$$
(3.56)

$$\mathbf{T}_{\mathrm{S}}^{\mathrm{S}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{\mathrm{T}} \tag{3.57}$$

The force exerted on link 4 by link 3 when i = 4 is

$$F_{4}^{4} = R_{S}^{4}F_{S}^{8} + f_{4}^{4}$$
$$= R_{U}^{4} \cdot R_{S}^{G} \times \begin{bmatrix} F_{n} \\ -F_{f} \\ 0 \end{bmatrix} + f_{4}^{4}$$
(3.58)

The force exerted on link 3 by link 2 is

$$F_{3}^{3} = R_{4}^{3}F_{4}^{4} + f_{3}^{3}$$

$$= \begin{bmatrix} C_{4} & -S_{4} & 0 \\ S_{4} & C_{4} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot F_{4}^{4} + f_{3}^{3}$$
(3.59)

The force exerted on link 2 by link 1 is  $% \left( \frac{1}{2} \right) = \left( \frac{1}{2} \right) \left( \frac{1}{2}$ 

$$F_{2}^{2} = R_{3}^{2}F_{3}^{3} + f_{2}^{2}$$

$$= \begin{bmatrix} -C_{3} & S_{3} & 0 \\ -S_{3} & -C_{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot F_{3}^{3} + f_{2}^{2}$$
(3.60)

The force exerted on link 1 by link 0 is

$$F_{1}^{1} = R_{2}^{1}F_{2}^{2} + f_{1}^{1}$$

$$= \begin{bmatrix} C_{2} & 0 & S_{2} \\ S_{2} & 0 & -C_{2} \\ 0 & 1 & 0 \end{bmatrix} \cdot F_{2}^{2} + f_{1}^{1}$$
(3.61)

The force exerted on link 0

The force exerted on link 0 can be transformed into universal frame  $\mathbf{O}_U$ 

$$F_{0}^{U} = \mathbf{R}_{0}^{U} F_{0}^{0}$$

$$= \begin{bmatrix} c\phi c\theta & -c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi + s\phi c\psi \\ s\theta & c\theta c\psi & -c\theta s\psi \\ -s\phi c\theta & s\phi s\theta c\psi + c\phi s\psi & -s\phi s\theta s\psi + c\phi c\psi \end{bmatrix} \cdot \mathbf{F}_{0}^{0}$$
(3.63)

The moments exerted on link i by link i-1 can be computed by following this algorithm

$$T_{i}^{i} = R_{i+1}^{i} \left( T_{i+1}^{i+1} + (R_{i}^{i+1}r_{i}^{i}) \times F_{i+1}^{i+1} \right) + (r_{i}^{i} + r_{c,i}^{i}) \times f_{i}^{i} + \tau_{i}^{i}$$
(3.64)

The moments exerted on link 4 by link 3 when i = 4 is

$$T_{4}^{4} = \mathbf{R}_{S}^{4} \left( T_{S}^{S} + (\mathbf{R}_{4}^{S} \mathbf{r}_{4}^{4}) \times \mathbf{F}_{S}^{S} \right) + (\mathbf{r}_{4}^{4} + \mathbf{r}_{c,4}^{4}) \times \mathbf{f}_{4}^{4} + \mathbf{\tau}_{4}^{4}$$

$$= \mathbf{R}_{S}^{4} \cdot \left( \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \left( \mathbf{R}_{S}^{4} \cdot \begin{bmatrix} \mathbf{a}_{4} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) \times \begin{bmatrix} -\mathbf{F}_{n} \\ \mathbf{F}_{f} \\ \mathbf{0} \end{bmatrix} \right) + \left( \begin{bmatrix} \mathbf{a}_{4} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{a}_{4} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) \times \mathbf{f}_{4}^{4} + \mathbf{\tau}_{4}^{4}$$
(3.65)

The moment exerted on link 3 by link 2 when i = 3 is

$$T_{3}^{3} = R_{4}^{3} \left( T_{4}^{4} + \left( R_{3}^{4} r_{3}^{3} \right) \times F_{4}^{4} \right) + \left( r_{3}^{3} + r_{c,3}^{3} \right) \times f_{3}^{3} + \tau_{3}^{3}$$

$$= \begin{bmatrix} C_{4} & -S_{4} & 0 \\ S_{4} & C_{4} & 0 \\ 0 & 0 & 1 \end{bmatrix} \left( T_{4}^{4} + \left( \begin{bmatrix} C_{4} & S_{4} & 0 \\ -S_{4} & C_{4} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{3} \\ 0 \\ 0 \end{bmatrix} \right) \times F_{4}^{4} \right) + \left( \begin{bmatrix} a_{3} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -0.5 a_{3} \\ 0 \\ 0 \end{bmatrix} \right) \times f_{3}^{3} + \tau_{3}^{3}$$

$$(3.66)$$

The moment exerted on link 2 by link 1 when i = 2 is

$$\begin{aligned} \mathbf{T}_{2}^{2} &= \mathbf{R}_{3}^{2} \left( \mathbf{T}_{3}^{3} + (\mathbf{R}_{2}^{3} \mathbf{r}_{2}^{2}) \times \mathbf{F}_{3}^{3} \right) + (\mathbf{r}_{2}^{2} + \mathbf{r}_{c,2}^{2}) \times \mathbf{f}_{2}^{2} + \mathbf{\tau}_{2}^{2} \\ &= \begin{bmatrix} -\mathbf{C}_{3} & \mathbf{S}_{3} & \mathbf{0} \\ -\mathbf{S}_{3} & -\mathbf{C}_{3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \left( \mathbf{T}_{3}^{3} + \left( \begin{bmatrix} -\mathbf{C}_{3} & -\mathbf{S}_{3} & \mathbf{0} \\ \mathbf{S}_{3} & -\mathbf{C}_{3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) \times \mathbf{F}_{3}^{3} \right) + \left( \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) \times \mathbf{f}_{2}^{2} + \mathbf{\tau}_{2}^{2} \end{aligned}$$
(3.67)

The moment exerted on link 1 by link 0 when i = 1 is

$$\begin{aligned} T_{1}^{1} &= R_{2}^{1} \left( T_{2}^{2} + (R_{1}^{2}r_{1}^{1}) \times F_{2}^{2} \right) + (r_{1}^{1} + r_{c,1}^{1}) \times f_{1}^{1} + \tau_{1}^{1} \\ &= \begin{bmatrix} C_{2} & 0 & S_{2} \\ S_{2} & 0 & -C_{2} \\ 0 & 1 & 0 \end{bmatrix} \left( T_{2}^{2} + \left( \begin{bmatrix} C_{2} & S_{2} & 0 \\ 0 & 0 & 1 \\ S_{2} & -C_{2} & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -d_{1} \\ 0 \end{bmatrix} \right) \times F_{2}^{2} \right) + \left( \begin{bmatrix} 0 \\ -d_{1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5d_{1} \\ 0 \end{bmatrix} \right) \times f_{1}^{1} + \tau_{1}^{1} \right) \end{aligned}$$
(3.68)

The moment exerted on link 0 when i = 0 is

$$\begin{split} \mathbf{T}_{0}^{0} &= \mathbf{R}_{1}^{0} \left( \mathbf{T}_{1}^{1} + (\mathbf{R}_{0}^{1} \mathbf{r}_{0}^{0}) \times \mathbf{F}_{1}^{1} \right) + (\mathbf{r}_{0}^{0} + \mathbf{r}_{c,0}^{0}) \times \mathbf{f}_{0}^{0} + \mathbf{\tau}_{0}^{0} \\ &= \begin{bmatrix} \mathbf{C}_{1} & \mathbf{0} & -\mathbf{S}_{1} \\ \mathbf{S}_{1} & \mathbf{0} & \mathbf{C}_{1} \\ \mathbf{0} & -\mathbf{1} & \mathbf{0} \end{bmatrix} \left( \mathbf{T}_{1}^{1} + \left( \begin{bmatrix} \mathbf{C}_{1} & \mathbf{S}_{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{1} \\ -\mathbf{S}_{1} & \mathbf{C}_{1} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) \times \mathbf{F}_{1}^{1} \right) + \left( \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right) \times \mathbf{f}_{0}^{0} + \mathbf{\tau}_{0}^{0} \end{split}$$
(3.69)

The moment exerted on link 0 can be transformed into universal frame  $O_U$ 

$$T_{0}^{U} = \mathbf{R}_{0}^{U} T_{0}^{0}$$

$$= \begin{bmatrix} c\phi c\theta & -c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi + s\phi c\psi \\ s\theta & c\theta c\psi & -c\theta s\psi \\ -s\phi c\theta & s\phi s\theta c\psi + c\phi s\psi & -s\phi s\theta s\psi + c\phi c\psi \end{bmatrix} \cdot T_{0}^{0}$$
(3.70)

The forces and torque exerted by the actuator at joint  $\dot{i}$  is

$$Q_{i} = \begin{cases} \left(F_{i}^{i}\right)^{T} \left(R_{i+1}^{i} Z_{U}\right); \text{ input force for prismatic link} \\ \left(T_{i}^{i}\right)^{T} \left(R_{i+1}^{i} Z_{U}\right); \text{ input torque for rotational link} \end{cases}$$
(3.71)

All joints used are revolute type, thus the input torque  $Q_i$  at each joint is the sum of the projection of  $T_i^i$  onto  $z_U(x_U, y_U, z_U)$  about the  $z_U$  axis

$$\mathbf{Q}_{i} = \left(\mathbf{T}_{i}^{i}\right)^{\mathrm{I}} \left(\mathbf{R}_{i+1}^{i} \mathbf{z}_{\mathrm{U}}\right) \tag{3.72}$$

The dynamic equation at joint 1 when i = 1 is given by

$$Q_{1} = (T_{1}^{1})^{T} (R_{1}^{0} z_{U})$$

$$= (T_{1}^{1})^{T} \left( \begin{bmatrix} C_{1} & 0 & -S_{1} \\ S_{1} & 0 & C_{1} \\ 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$$
(3.73)

The dynamic equation at joint 2 when i = 2 is given by

$$Q_{2} = (T_{2}^{2})^{T} (R_{2}^{1} Z_{U})$$

$$= (T_{2}^{2})^{T} \left( \begin{bmatrix} C_{2} & 0 & S_{2} \\ S_{2} & 0 & -C_{2} \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$$
(3.74)

The dynamic equation of joint 3 when i = 3 is given by

$$Q_{3} = (T_{3}^{3})^{T} (R_{3}^{2} z_{U})$$

$$= (T_{3}^{3})^{T} \left( \begin{bmatrix} -C_{3} & S_{3} & 0 \\ -S_{3} & -C_{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$$
(3.75)

Finally, The dynamic equation at joint 4 when i = 4 is given by

$$Q_{4} = (T_{4}^{4})^{T} (R_{4}^{3} z_{U})$$

$$= (T_{4}^{4})^{T} \left( \begin{bmatrix} C_{4} & -S_{4} & 0 \\ S_{4} & C_{4} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right)$$
(3.76)

# **Chapter Four**

## 4. System forces and moments

The system forces and moments are transformed to platform universal frame from outermost link till the innermost, link by link. The system forces are generated from the system weight of rover, inertial forces on the center of mass of links, and direct contacts between wheels and ground surface expressed as the normal force, and frictional force.

The longitudinal and lateral forces exerted on wheel are relatively small values and even avoided here in this work. Furthermore, the moments between wheel and surface are also negligible.

#### 4.1.1. System weight

Benefiting from Newton-Euler recursive method, the total weight of rover will be evaluated at the platform universal frame. This is resulted from transforming the gravity force of each link starting from wheel link to platform link, link by link. In other meaning, the system weight is defined as a gravitational force vector of a system mass constant times the acceleration of gravity that points downward vertically in  $x_U$  of the platform universal frame  $O_U$ .

System weight = 
$$\begin{bmatrix} -mg & 0 & 0 \end{bmatrix}^T$$
 (4.1)

Where, **m** is the system mass constant of rover links and is equal to 12kg, **g** is the gravitational acceleration produced in a body due to the Mars' gravitational attraction; Its SI unit is  $m/s^2$  and its values on the surfaces of the earth and Mars, respectively, are  $9.8m/s^2$  and  $3.63 m/s^2$ .

The vertical projection of system weight from center of mass will be distributed among the contact wheels on the base of joint configurations, ground geometries, and rover attitude. The amounts of distributed weights on wheels are simply defined as wheel pressures on ground contacts. In the case of symmetric configurations and flat surface, the projection of center of mass will be at the middle area of the support polygon, so the system weight will be distributed equally among these wheels. However, if the changing occurs in joint configurations and ground geometry during the travel; the position of vertical project of center of mass will change and the amounts of wheeled pressures on ground will already change.

Wheel universal frame  $O_w$  is assigned at each contact point in order to represent the part of distributed weight on that wheel.  $O_u$  and  $O_w$  are contingent.

$$\mathbf{F}_{\mathbf{W}}^{\mathbf{W}} = \begin{bmatrix} -\mathbf{F}_{\mathbf{W}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$
(4.2)

#### 4.1.2. Normal force

The normal force is inspired from Newton's third law which states that for each action force there is reaction force with the same magnitude and opposite direction. The contact always generates reaction force acting perpendicular to the contact surface expressed in surface frame  $O_s$ .

Whenever any wheel of the rover are in contact with ground, the gravitational or weight force acting on wheel will apply to the ground, so the ground will react on the wheel with normal force. The magnitude of the normal force is equal weight force component applied in  $x_s$  axis. The direction of the

normal force is instantaneously perpendicular to the surface in  $x_s$  axis. In the case of the flat surface, the normal force is coplanar to the positive axis  $x_w$  direction. However, if the wheels choose their footholds on inclined or smooth uneven surface, the normal force will make angle relatively to wheel universal frame  $O_w$ .

In static stabilizing condition, the number of supporting wheels on ground can vary between 3 and 4 for a quadruped robot. In the case of symmetric configurations and flat surface, the weight force acting by supporting wheel on the ground is equal the weight of system rover divided by the number of supporting wheels

$$F_n = \frac{mg}{nc}$$
(4.3)

Where,  $F_n$  is the static normal force acting from the ground on the supporting wheel. mg is the weight of the rover acing on ground and directed coplanar with respect universal frame. nc is the number of wheels which are in contact with the ground. However as shown in Figure 4.1, in the case of non-

symmetric rover and non-uniform surface geometries the specific equation 4.3 is totally not capable for evaluating the normal forces.



Figure 4.1. Normal forces acting on wheels perpendicular to surface.

This rover dealt with unknown reactions for four, three, and two legs; the system of four legs has three equations and four unknown so it is considered as indeterminate system of equations, while the system of three legs has three equations and three unknowns so it is considered determinate system of equation Furthermore, the two legs system has two variables and provided with two equations, thus this is considered determinate system of equation.

### 4.1.3. Frictional force

In general, the friction is resulted from the pressing two surfaces with each other, and generates deformation, heat, as well as frictional force in the opposite direction of motion. The types of frictions are rolling, sliding slipping frictions. In any way, in pure rolling motion there is no sliding or slipping; and rolling on solid surface yields no rolling friction at all. The direction of motion is always perpendicular to the normal force and tangent to surface. The rolling friction occurs between wheels and contact area surface. Whereas rolling frictional force is a function of normal force acting from ground on wheel and coefficient of rolling friction.

$$\mathbf{F}_{\mathrm{f}} = \boldsymbol{\mu} \mathbf{F}_{\mathrm{n}} \tag{4.4}$$

Where,  $F_f$  is the rolling frictional force occurred between the wheel and the soft terrain.  $\mu$  is rolling coefficient friction between two surfaces.  $F_n$  is the normal force exerted on the wheel. However, rolling friction occurred when the rover is moving on soft terrain. Thus, the rolling coefficient of this work is equal zero because we assume solid surface.

#### 4.1.4. Wheeled motor torque

The motor exerts required amount of torque that enables wheel to grip with surface and propel it in tangent direction of the surface. The motor torque is equal the cross product of traction force and wheel radius.

$$T_{\rm m} = \mathbf{r} \times \mathbf{F}_{\rm t} \tag{4.5}$$

Where,  $T_m$  is the motor torque, and  $F_t$  is the traction force in the direction of tangential line of surface. The motor torque rotates about  $z_3$  axis in the direction of wheel rotation.

Finally, the resultant of forces and moments exerted on wheel endeffector are computed with respect to frame  $O_S(x_S, y_S, z_S)$  as shown in Figure 4.2, and it can be obtained respectively in two 3 × 1 vectors.

$$\mathbf{F}_{\mathrm{S}}^{\mathrm{S}} = \begin{bmatrix} \mathbf{F}_{\mathrm{n}} & -\mathbf{F}_{\mathrm{f}} & \mathbf{0} \end{bmatrix}^{\mathrm{T}}$$
(4.6)

$$\mathbf{T}_{\mathrm{S}}^{\mathrm{S}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{\mathrm{T}} \tag{4.7}$$

Because the wheels are locomoted on solid surface as assumed, the rolling friction is negligible in this work.



Figure 4.2. External forces and moments exerted by ground on end-effector projected in frame O<sub>4</sub>.

Where,

 $F_{n} \quad$  normal force perpendicular to contact surface and in  $\boldsymbol{x}_{S}$  axis direction.

 $F_{\rm f}$  frictional force tangential of contact surface in opposite direction of wheel linear motion in the direction of -y<sub>s</sub> axis.

 $F_W \ \ \, weight$  acing on center of wheel and directed downward in  $\mbox{-} x_W$  axis of wheel universal frame  $O_W.$
$T_m$  motor moment in direction of wheel rotational motion about  $z_3$ -axis.

 $\beta$  Slope angle of inclined surface.

The summation of all moments (resulted fro normal forces, inertial forces, gravity forces exerted on center of mass of links, and torques exerted on link) about the contact wheels are equal zero. Thus, this is the definition of balanced equation.

Now for the four contact legs, the summation of all moments can be given as in equations 4.9, 4.15, 4.21, and 4.27 as shown respectively in:

$$\begin{split} \sum M_{4RF} &= \begin{bmatrix} 0\\ 0\\ 0\\ \end{bmatrix} \\ (r_{4RF}^{U} - r_{4RF}^{U}) \times \left( R_{SRR}^{U} F_{SRR}^{SRR} \right) + (r_{4LF}^{U} - r_{4RF}^{U}) \times \left( R_{SLF}^{U} F_{SLF}^{SLF} \right) + (r_{4LR}^{U} - r_{4RF}^{U}) \times \left( R_{SLR}^{U} F_{SLR}^{SLR} \right) + \\ (r_{c,4RF}^{U} - r_{4RF}^{U}) \times \left( R_{4RF}^{U} f_{4RF}^{4RF} \right) + (r_{c,3RF}^{U} - r_{4RF}^{U}) \times \left( R_{3RF}^{U} f_{3RF}^{3RF} \right) + (r_{c,4RR}^{U} - r_{4RF}^{U}) \times \left( R_{4RR}^{U} f_{4RR}^{4RR} \right) + \\ (r_{c,3RR}^{U} - r_{4RF}^{U}) \times \left( R_{3RR}^{U} f_{3RR}^{3RR} \right) + (r_{c,2R}^{U} - r_{4RF}^{U}) \times \left( R_{2R}^{U} f_{2R}^{2R} \right) + (r_{c,1R}^{U} - r_{4RF}^{U}) \times \left( R_{1R}^{U} f_{1R}^{1R} \right) + \\ (r_{c,3RF}^{U} - r_{4RF}^{U}) \times \left( R_{0R}^{U} f_{0R}^{0R} \right) + (r_{c,1L}^{U} - r_{4RF}^{U}) \times \left( R_{1L}^{U} f_{1L}^{1L} \right) + (r_{c,2L}^{U} - r_{4RF}^{U}) \times \left( R_{2L}^{U} f_{2L}^{2L} \right) + \\ (r_{c,3LF}^{U} - r_{4RF}^{U}) \times \left( R_{3LF}^{U} f_{3LF}^{3LF} \right) + (r_{c,4LF}^{U} - r_{4RF}^{U}) \times \left( R_{4LF}^{U} f_{4LF}^{4LF} \right) + \\ (r_{c,3LR}^{U} - r_{4RF}^{U}) \times \left( R_{3LR}^{U} f_{3LR}^{3LR} \right) + (r_{c,4LR}^{U} - r_{4RF}^{U}) \times \left( R_{4LR}^{U} f_{4LR}^{4LR} \right) + \\ R_{4RF}^{U} r_{4RF}^{4RF} + R_{3RF}^{U} r_{3RF}^{3RF} + R_{4RR}^{U} r_{4RR}^{4RR} + R_{3RR}^{U} r_{3RR}^{3RR} + \\ R_{4LF}^{U} r_{4LF}^{2R} + R_{3LF}^{U} r_{3LF}^{3LF} + R_{4LR}^{U} r_{4RR}^{4LR} + R_{3LR}^{U} r_{3RR}^{3RR} + \\ R_{4LF}^{U} r_{4LF}^{2R} + R_{1R}^{U} r_{1R}^{3LF} + R_{4LR}^{U} r_{4LR}^{4LR} + R_{3LR}^{U} r_{3RR}^{3RR} + \\ R_{4LF}^{U} r_{4LF}^{2R} + R_{1R}^{U} r_{1R}^{3LF} + R_{4LR}^{U} r_{4LR}^{2L} + R_{0R}^{U} r_{0R}^{3RR} = \begin{bmatrix} 0\\ 0\\ 0 \end{bmatrix}$$

$$(4.8)$$

The above equation can be abbreviated with notations and it provides for

definition of the equation of balanced,

$$F_{nSRR} \cdot B_1 + F_{nSLF} \cdot B_2 + F_{nSLR} \cdot B_3 + M_1 = 0$$

$$(4.9)$$

Where,

$$\mathbf{B}_{1} = (\mathbf{r}_{4RR}^{U} - \mathbf{r}_{4RF}^{U}) \times (\mathbf{R}_{SRR}^{U} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{\mathrm{T}})$$
(4.10)

$$\mathbf{B}_{2} = (\mathbf{r}_{4LF}^{U} - \mathbf{r}_{4RF}^{U}) \times (\mathbf{R}_{SLF}^{U} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{\mathrm{T}})$$
(4.11)

$$B_{3} = (r_{4LR}^{U} - r_{4RF}^{U}) \times (R_{SLR}^{U} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{T})$$
(4.12)

and

$$\mathbf{M}_{1} = \sum_{i} (\mathbf{r}_{c,i}^{U} - \mathbf{r}_{4RF}^{U}) \times \left(\mathbf{R}_{i}^{U} \mathbf{f}_{i}^{i}\right) + \sum_{i} \mathbf{R}_{i}^{U} \boldsymbol{\tau}_{i}^{i}$$
(4.13)

$$\begin{split} \sum M_{4RR} &= \begin{bmatrix} 0\\ 0\\ 0\\ \end{bmatrix} \\ (r_{4RF}^{U} - r_{4RR}^{U}) \times \left( R_{SRF}^{U} F_{SRF}^{SRF} \right) + (r_{4LF}^{U} - r_{4RR}^{U}) \times \left( R_{SLF}^{U} F_{SLF}^{SLF} \right) + (r_{4LR}^{U} - r_{4RR}^{U}) \times \left( R_{SLR}^{U} F_{SLR}^{SLR} \right) + \\ (r_{c,4RF}^{U} - r_{4RR}^{U}) \times \left( R_{4RF}^{U} f_{4RF}^{4RF} \right) + (r_{c,3RF}^{U} - r_{4RR}^{U}) \times \left( R_{3RF}^{U} f_{3RF}^{3RF} \right) + (r_{c,4RR}^{U} - r_{4RR}^{U}) \times \left( R_{4RR}^{U} f_{4RR}^{4RR} \right) + \\ (r_{c,3RR}^{U} - r_{4RR}^{U}) \times \left( R_{3RR}^{U} f_{3RR}^{3R} \right) + (r_{c,2R}^{U} - r_{4RR}^{U}) \times \left( R_{2R}^{U} f_{2R}^{2R} \right) + (r_{c,1R}^{U} - r_{4RR}^{U}) \times \left( R_{1R}^{U} f_{1R}^{1R} \right) + \\ (r_{c,3RF}^{U} - r_{4RR}^{U}) \times \left( R_{0R}^{U} f_{0R}^{0R} \right) + (r_{c,1L}^{U} - r_{4RR}^{U}) \times \left( R_{1L}^{U} f_{1L}^{1L} \right) + (r_{c,2L}^{U} - r_{4RR}^{U}) \times \left( R_{2L}^{U} f_{2L}^{2L} \right) + \\ (r_{c,3LF}^{U} - r_{4RR}^{U}) \times \left( R_{3LR}^{U} f_{3LF}^{3LF} \right) + (r_{c,4LF}^{U} - r_{4RR}^{U}) \times \left( R_{4LF}^{U} f_{4LF}^{4LF} \right) + \\ (r_{c,3LR}^{U} - r_{4RR}^{U}) \times \left( R_{3LR}^{U} f_{3LR}^{3LR} \right) + (r_{c,4R}^{U} - r_{4RR}^{U}) \times \left( R_{4LR}^{U} f_{4LR}^{4LR} \right) + \\ R_{4RF}^{U} \tau_{4RF}^{4RF} + R_{3RF}^{U} \tau_{3RF}^{3RF} + R_{4RR}^{U} \tau_{4RR}^{4RR} + R_{3RR}^{U} \tau_{3RR}^{3RR} + \\ R_{4LF}^{U} \tau_{4LF}^{4LF} + R_{3LF}^{U} \tau_{3LF}^{3LF} + R_{4LR}^{U} \tau_{4LR}^{4RR} + R_{3LR}^{U} \tau_{3LR}^{3RR} + \\ R_{4LF}^{U} \tau_{2R}^{2R} + R_{1R}^{U} \tau_{1R}^{1R} + R_{2L}^{U} \tau_{2L}^{2L} + R_{1L}^{U} \tau_{1L}^{1L} + R_{0R}^{U} \tau_{0R}^{0R} = \begin{bmatrix} 0\\ 0\\ 0\\ \end{bmatrix}$$

$$(4.14)$$

$$F_{nSRF} \cdot C_1 + F_{nSLF} \cdot C_2 + F_{nSLR} \cdot C_3 + M_2 = 0$$
(4.15)

Where,

$$\mathbf{C}_{1} = (\mathbf{r}_{4RF}^{U} - \mathbf{r}_{4RR}^{U}) \times (\mathbf{R}_{SRF}^{U} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{\mathrm{T}})$$
(4.16)

$$C_{2} = (r_{4LF}^{U} - r_{4RR}^{U}) \times (R_{SLF}^{U} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{T})$$
(4.17)

$$C_{3} = (r_{4LR}^{U} - r_{4RR}^{U}) \times (R_{SLR}^{U} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{T})$$
(4.18)

$$M_{2} = \sum_{i} (r_{c,i}^{U} - r_{4RR}^{U}) \times (R_{i}^{U} f_{i}^{i}) + \sum_{i} R_{i}^{U} \tau_{i}^{i}$$
(4.19)

$$\begin{split} \sum M_{4LF} &= \begin{bmatrix} 0\\ 0\\ 0\\ \end{bmatrix} \\ (r_{4RF}^{U} - r_{4LF}^{U}) \times \left( R_{SRF}^{U} F_{SRF}^{SRF} \right) + (r_{4RR}^{U} - r_{4LF}^{U}) \times \left( R_{SRR}^{U} F_{SRR}^{SRR} \right) + (r_{4LR}^{U} - r_{4LF}^{U}) \times \left( R_{SLR}^{U} F_{SLR}^{SLR} \right) + \\ (r_{c,4RF}^{U} - r_{4LF}^{U}) \times \left( R_{4RF}^{U} f_{4RF}^{4RF} \right) + (r_{c,3RF}^{U} - r_{4LF}^{U}) \times \left( R_{3RF}^{U} f_{3RF}^{3SF} \right) + (r_{c,4RR}^{U} - r_{4LF}^{U}) \times \left( R_{4RR}^{U} f_{4RR}^{4RR} \right) + \\ (r_{c,3RR}^{U} - r_{4LF}^{U}) \times \left( R_{3RR}^{U} f_{3RR}^{3R} \right) + (r_{c,2R}^{U} - r_{4LF}^{U}) \times \left( R_{2R}^{U} f_{2R}^{2R} \right) + (r_{c,1R}^{U} - r_{4LF}^{U}) \times \left( R_{1R}^{U} f_{1R}^{1R} \right) + \\ (r_{c,0R}^{U} - r_{4LF}^{U}) \times \left( R_{0R}^{U} f_{0R}^{0R} \right) + (r_{c,1L}^{U} - r_{4LF}^{U}) \times \left( R_{1L}^{U} f_{1L}^{1L} \right) + (r_{c,2L}^{U} - r_{4LF}^{U}) \times \left( R_{2L}^{U} f_{2L}^{2L} \right) + \\ (r_{c,3LF}^{U} - r_{4LF}^{U}) \times \left( R_{3LF}^{U} f_{3LF}^{3LF} \right) + (r_{c,4LF}^{U} - r_{4LF}^{U}) \times \left( R_{4LF}^{U} f_{4LF}^{4LF} \right) + \\ (r_{c,3LR}^{U} - r_{4LF}^{U}) \times \left( R_{3LR}^{U} f_{3LR}^{3LR} \right) + (r_{c,4R}^{U} - r_{4LF}^{U}) \times \left( R_{4LF}^{U} f_{4LR}^{4R} \right) + \\ R_{4RF}^{U} r_{4RF}^{4RF} + R_{3RF}^{U} r_{3RF}^{3RF} + R_{4RR}^{U} r_{4RR}^{4RR} + R_{3RR}^{U} r_{3RR}^{3RR} + \\ R_{4LF}^{U} r_{4LF}^{4LF} + R_{3LF}^{U} r_{3LF}^{3LF} + R_{4LR}^{U} r_{4LR}^{4RR} + R_{3LR}^{U} r_{3LR}^{3LR} + \\ R_{4LF}^{U} r_{4RF}^{2R} + R_{1R}^{U} r_{4RF}^{3LF} + R_{4LR}^{U} r_{4LR}^{2L} + R_{3LR}^{U} r_{3RR}^{3LR} + \\ R_{2R}^{U} r_{2R}^{2R} r_{2R}^{2R} + R_{1R}^{U} r_{1R}^{1R} + R_{2L}^{U} r_{2L}^{2L} + R_{1L}^{U} r_{1L}^{1L} + R_{0R}^{U} r_{0R}^{0R} = \begin{bmatrix} 0\\ 0\\ 0\\ \end{bmatrix} \right)$$

$$(4.20)$$

$$F_{nSRF} \cdot D_1 + F_{nSRR} \cdot D_2 + F_{nSLR} \cdot D_3 + M_3 = 0$$
(4.21)

Where,

$$\mathbf{D}_{1} = (\mathbf{r}_{4\mathrm{RF}}^{\mathrm{U}} - \mathbf{r}_{4\mathrm{LF}}^{\mathrm{U}}) \times (\mathbf{R}_{\mathrm{SRF}}^{\mathrm{U}} \begin{bmatrix} 1 & -\mu & \mathbf{0} \end{bmatrix}^{\mathrm{T}})$$
(4.22)

$$\mathbf{D}_{2} = (\mathbf{r}_{4\mathrm{RR}}^{\mathrm{U}} - \mathbf{r}_{4\mathrm{LF}}^{\mathrm{U}}) \times (\mathbf{R}_{\mathrm{SRR}}^{\mathrm{U}} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{\mathrm{T}})$$
(4.23)

$$\mathbf{D}_{3} = (\mathbf{r}_{4LR}^{U} - \mathbf{r}_{4LF}^{U}) \times (\mathbf{R}_{SLR}^{U} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{\mathrm{T}})$$
(4.24)

$$M_{3} = \sum_{i} (r_{c,i}^{U} - r_{4LF}^{U}) \times \left(R_{i}^{U} f_{i}^{i}\right) + \sum_{i} R_{i}^{U} \tau_{i}^{i}$$
(4.25)

$$\begin{split} \sum M_{4LR} &= \begin{bmatrix} 0\\ 0\\ 0\\ \end{bmatrix} \\ (r_{4RF}^{U} - r_{4LR}^{U}) \times \left( R_{SRF}^{U} F_{SRF}^{SRF} \right) + (r_{4RR}^{U} - r_{4LR}^{U}) \times \left( R_{SRR}^{U} F_{SRR}^{SRR} \right) + (r_{4LF}^{U} - r_{4LR}^{U}) \times \left( R_{SLF}^{U} F_{SLF}^{SLF} \right) + \\ (r_{c,4RF}^{U} - r_{4LR}^{U}) \times \left( R_{4RF}^{U} f_{4RF}^{4RF} \right) + (r_{c,3RF}^{U} - r_{4LR}^{U}) \times \left( R_{3RF}^{U} f_{3RF}^{3RF} \right) + (r_{c,4RR}^{U} - r_{4LR}^{U}) \times \left( R_{4RR}^{U} f_{4RR}^{4RR} \right) + \\ (r_{c,3RR}^{U} - r_{4LR}^{U}) \times \left( R_{3RR}^{U} f_{3RR}^{3R} \right) + (r_{c,2R}^{U} - r_{4LR}^{U}) \times \left( R_{2R}^{U} f_{2R}^{2R} \right) + (r_{c,1R}^{U} - r_{4LR}^{U}) \times \left( R_{1R}^{U} f_{1R}^{1R} \right) + \\ (r_{c,0R}^{U} - r_{4LR}^{U}) \times \left( R_{0R}^{U} f_{0R}^{0R} \right) + (r_{c,1L}^{U} - r_{4LR}^{U}) \times \left( R_{1L}^{U} f_{1L}^{1L} \right) + (r_{c,2L}^{U} - r_{4LR}^{U}) \times \left( R_{2L}^{U} f_{2L}^{2L} \right) + \\ (r_{c,3LF}^{U} - r_{4LR}^{U}) \times \left( R_{3LR}^{U} f_{3LF}^{3LF} \right) + (r_{c,4LF}^{U} - r_{4LR}^{U}) \times \left( R_{4LF}^{U} f_{4LF}^{4LR} \right) + \\ (r_{c,3LR}^{U} - r_{4LR}^{U}) \times \left( R_{3LR}^{U} f_{3LR}^{3LR} \right) + (r_{c,4LR}^{U} - r_{4LR}^{U}) \times \left( R_{4LF}^{U} f_{4LR}^{4LR} \right) + \\ R_{4RF}^{U} r_{4RF}^{4RF} + R_{3RF}^{U} r_{3RF}^{3RF} + R_{4RR}^{U} r_{4RR}^{4RR} + R_{3RR}^{U} r_{3RR}^{3RR} + \\ R_{4LF}^{U} r_{4LF}^{2R} + R_{3LF}^{U} r_{3LF}^{3LF} + R_{4LR}^{U} r_{4LR}^{4RR} + R_{3LR}^{U} r_{3RR}^{3LR} + \\ R_{4LF}^{U} r_{4LF}^{2R} + R_{1R}^{U} r_{1R}^{3LF} + R_{4LR}^{U} r_{4LR}^{2L} + R_{3LR}^{U} r_{3RR}^{3LR} + \\ R_{2R}^{U} r_{2R}^{2R} r_{2R}^{2R} + R_{1R}^{U} r_{1R}^{1R} + R_{2L}^{U} r_{2L}^{2L} + R_{1L}^{U} r_{1L}^{1L} + R_{0R}^{U} r_{0R}^{0R} = \begin{bmatrix} 0\\ 0\\ 0\\ \end{bmatrix} \right]$$

$$(4.26)$$

$$F_{nSRF} \cdot E_1 + F_{nSRR} \cdot E_2 + F_{nSLF} \cdot E_3 + M_4 = 0$$
(4.27)

Where,

$$E_{1} = (r_{4RF}^{U} - r_{4LR}^{U}) \times (R_{SRF}^{U} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{T})$$
(4.28)

$$\mathbf{E}_{2} = (\mathbf{r}_{4\mathrm{RR}}^{\mathrm{U}} - \mathbf{r}_{4\mathrm{LR}}^{\mathrm{U}}) \times (\mathbf{R}_{\mathrm{SRR}}^{\mathrm{U}} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{\mathrm{T}})$$
(4.29)

$$E_{3} = (r_{4LF}^{U} - r_{4LR}^{U}) \times (R_{SLF}^{U} \begin{bmatrix} 1 & -\mu & 0 \end{bmatrix}^{T})$$
(4.30)

$$M_{4} = \sum_{i} (\mathbf{r}_{c,i}^{U} - \mathbf{r}_{4LR}^{U}) \times \left( \mathbf{R}_{i}^{U} \mathbf{f}_{i}^{i} \right) + \sum_{i} \mathbf{R}_{i}^{U} \boldsymbol{\tau}_{i}^{i}$$
(4.31)

## Normal Force Algorithm:

% Right legs and left legs are in contact with ground,

if  $(RF \mapsto \bullet \&\& RR \mapsto \bullet) \&\& (LF \mapsto \bullet \&\& LR \mapsto \bullet)$ 

$$if (Roll \sim = 0) \& \& (Pitch == 0)$$

$$F_{nSRF} = \frac{-M_3(2)}{D_1(2) + D_2(2)}$$
(4.32)

$$F_{nSRR} = F_{nSRF} \tag{4.33}$$

$$F_{nSLF} = \frac{-M_1(2)}{B_2(2) + B_3(2)}$$
(4.34)

$$F_{nSLR} = F_{nSLF} \tag{4.35}$$

if 
$$(Roll == 0 \& \& Pitch \sim = 0) || (Roll == 0 \& \& Pitch == 0)$$

$$F_{nSRF} = \frac{-M_2(3)}{C_1(3) + C_2(3)}$$
(4.36)

$$F_{nSRR} = \frac{-M_1(3)}{B_1(3) + B_3(3)}$$
(4.37)

$$F_{nSLF} = F_{nSRF} \tag{4.38}$$

$$\mathbf{F}_{\mathrm{nSLR}} = \mathbf{F}_{\mathrm{nSRR}} \tag{4.39}$$

End

End

# $if (RF \mapsto \bullet \&\& RR \mapsto \bullet) \&\& (LF \mapsto \circ \&\& LR \mapsto \circ)$

% Right legs are in contact with ground and left legs are without,

$$\text{Coefficient} = \begin{bmatrix} 0 & B_1(3) \\ C_1(3) & 0 \end{bmatrix}$$
(4.40)

$$\mathbf{b} = \begin{bmatrix} -\mathbf{M}_1(3) & -\mathbf{M}_2(3) \end{bmatrix}$$
(4.41)

$$x = inv(Coefficient) * b$$
(4.42)

$$\mathbf{F}_{\mathrm{nSRF}} = \mathbf{x}(1) \tag{4.43}$$

$$F_{nSRR} = x(2) \tag{4.44}$$

$$F_{nSLF} = 0 \tag{4.45}$$

$$F_{nSLR} = 0 \tag{4.46}$$

elseif (RF  $\mapsto \circ$  & & RR  $\mapsto \circ$ ) & & (LF  $\mapsto \bullet$  & LR  $\mapsto \bullet$ )

%Left legs are in contact with ground and right legs without contact

$$\text{Coefficient} = \begin{bmatrix} 0 & D_3(3) \\ E_3(3) & 0 \end{bmatrix}$$
(4.47)

$$\mathbf{b} = \begin{bmatrix} -\mathbf{M}_3(3) & -\mathbf{M}_4(3) \end{bmatrix}$$
(4.48)

$$\mathbf{x} = \operatorname{inv}(\operatorname{Coefficient}) * \mathbf{b} \tag{4.49}$$

$$\mathbf{F}_{\mathbf{n}\mathbf{SRF}} = \mathbf{0} \tag{4.50}$$

$$F_{nSRR} = 0 \tag{4.51}$$

$$\mathbf{F}_{\mathrm{nSLF}} = \mathbf{x}(1) \tag{4.52}$$

$$\mathbf{F}_{\mathsf{nSLR}} = \mathbf{x}(2) \tag{4.53}$$

End

$$\mathrm{if}\,\big(\mathrm{RF}\mapsto\bullet\,\&\,\&\,\mathrm{RR}\mapsto\bullet\big)\&\,\&\big(\big(\mathrm{LF}\mapsto\bullet\&\,\&\,\mathrm{LR}\mapsto\circ\big)\|\,\big(\mathrm{LF}\mapsto\circ\&\,\&\,\mathrm{LR}\mapsto\bullet\big)\big)$$

% Right legs are in contact with ground and either left front or rear leg is without,

Coefficient = 
$$\begin{bmatrix} 0 & B_1(2) & B_2(2) & B_3(2) \\ 0 & B_1(3) & B_2(3) & B_3(3) \\ C_1(3) & 0 & C_2(3) & C_3(3) \\ H_1(1) & H_2(1) & H_3(1) & H_4(1) \end{bmatrix}$$
(4.54)

$$\mathbf{b} = \begin{bmatrix} -M_1(2) & -M_1(3) & -M_2(3) & -\text{System}\_\text{Force}\_\text{U}(1) \end{bmatrix}^{\text{T}}$$
(4.55)

$$\mathbf{x} = \text{inv}(\text{Coefficient}) * \mathbf{b} \tag{4.56}$$

$$\mathbf{F}_{\mathrm{nSRF}} = \mathbf{x}(1) \tag{4.57}$$

$$\mathbf{F}_{\mathrm{nSRR}} = \mathbf{x}(2) \tag{4.58}$$

$$F_{nSLF} = x(3) \tag{4.59}$$

$$\mathbf{F}_{\mathrm{nSLR}} = \mathbf{x}(4) \tag{4.60}$$

 $elseif((RF \mapsto \bullet \& \& RR \mapsto \circ) || (RF \mapsto \circ \& \& RR \mapsto \bullet)) \& \& (LF \mapsto \bullet \& \& LR \mapsto \bullet)$ 

%Either right font or rear leg is without contact and left legs are in contact,

Coefficient = 
$$\begin{bmatrix} D_{1}(2) & D_{2}(2) & 0 & D_{3}(2) \\ D_{1}(3) & D_{2}(3) & 0 & D_{3}(3) \\ E_{1}(3) & E_{2}(3) & E_{3}(3) & 0 \\ H_{1}(1) & H_{2}(1) & H_{3}(1) & H_{4}(1) \end{bmatrix}$$
(4.61)

$$\mathbf{b} = \begin{bmatrix} -M_3(2) & -M_3(3) & -M_4(3) & -\text{System}\_\text{Force}\_\text{U}(1) \end{bmatrix}^{\text{T}}$$
(4.62)

x = inv(Coefficient) * b	(4.63)
$F_{nSRF} = x(1)$	(4.64)
$F_{nSRR} = x(2)$	(4.65)
$F_{nSLF} = x(3)$	(4.66)
$F_{nSLR} = x(4)$	(4.67)

134

End

#### 4.1.5. Constraints

The normal force is positive, if the wheel remains in contact with ground. Otherwise, it is equal zero.

 $F_n > 0$ if leg  $\mapsto \bullet$ ; means that the wheel is in contact with ground. $F_n = 0$ if leg  $\mapsto \circ$ ; means that the wheel is not in contact with ground.

During motion of rigid wheels on rigid surface, if motor exerts high torque, then wheel will slip and provide low speed. Thus the traction force must be less or equal the frictional force to make rigid wheels capable for gripping with rigid surface.

$$\mathbf{F}_{\mathrm{m}} \le \boldsymbol{\mu} \mathbf{F}_{\mathrm{n}} \tag{4.68}$$

### **Chapter Five**

#### 5. Results and discussion

This work concludes that the rover will be dynamically stable if it meets this condition: "The universal moment at platform resulted from gravity force, inertial forces and torques exerted on center of mass of each link, and normal forces exerted on end-effectors must not equal the critical moments".

The platform can be represented as a collection of effects of system normal forces, system weights, and system inertial, since the backward dynamic system propagates those forces and moments from outermost to innermost link by link starting from end-effector till the platform link.

The critical moment is the required moment to rotate the rover and lose one side's connections with ground in order to rotate the rover about the opposite sides. The four critical moments about edges of contact points are threshold limits evaluated by substituting in universal moment with zero normal forces for the opposite side as shown in Figure 5.1,



Figure 5.1. Four critical moments

The left legs are substituted zero normal forces in balance equations 4.14 and 4.8, respectively, in order to find the normals on the right critical contact line,

$$F_{nSRF} = \frac{-M_2(3)}{C_1(3)}$$
(5.1)

$$F_{nRR} = \frac{-M_1(3)}{B_1(3)}$$
(5.2)

The critical moment required to turn the rover over the right side takes place when the left legs are uncontact with ground and the equations 5.1 and 5.2 are substituted in equation F.8, we obtain

$$\begin{aligned} \tau_{C,Right} &= r_{4RF}^{U} \times \left(R_{SRF}^{U} F_{SRF}^{SRF}\right) + r_{4RR}^{U} \times \left(R_{SRR}^{U} F_{SRR}^{SRR}\right) + \\ & r_{c,4RF}^{U} \times \left(R_{4RF}^{U} f_{4RF}^{4RF}\right) + r_{c,3RF}^{U} \times \left(R_{3RF}^{U} f_{3RF}^{3RF}\right) + r_{c,4RR}^{U} \times \left(R_{4RR}^{U} f_{4RR}^{4RR}\right) + r_{c,3RR}^{U} \times \left(R_{3RR}^{U} f_{3RR}^{3RR}\right) + \\ & r_{c,2R}^{U} \times \left(R_{2R}^{U} f_{2R}^{2R}\right) + r_{c,1R}^{U} \times \left(R_{1R}^{U} f_{1R}^{1R}\right) + r_{c,0R}^{U} \times \left(R_{0R}^{U} f_{0R}^{0R}\right) + r_{c,1L}^{U} \times \left(R_{1L}^{U} f_{1L}^{1L}\right) + r_{c,2L}^{U} \times \left(R_{2L}^{U} f_{2L}^{2L}\right) + \\ & r_{c,3LF}^{U} \times \left(R_{3LF}^{U} f_{3LF}^{3LF}\right) + r_{c,4LF}^{U} \times \left(R_{4LF}^{U} f_{4LF}^{4LF}\right) + r_{c,3LR}^{U} \times \left(R_{3LR}^{U} f_{3LR}^{3LR}\right) + r_{c,4LR}^{U} \times \left(R_{4LR}^{U} f_{4LR}^{4LR}\right) + \\ & R_{4RF}^{U} \tau_{4RF}^{4RF} + R_{3RF}^{U} \tau_{3RF}^{3RF} + R_{4RR}^{U} \tau_{4RR}^{4RR} + R_{3RR}^{U} \tau_{3RR}^{3RR} + R_{4LF}^{U} \tau_{4LF}^{4LF} + R_{3LF}^{U} \tau_{3LF}^{3LF} + R_{4LR}^{U} \tau_{4LR}^{4LR} + R_{3LR}^{U} \tau_{3LR}^{3LR} + \\ & R_{2R}^{U} \tau_{2R}^{2R} + R_{1R}^{U} \tau_{1R}^{1R} + R_{2L}^{U} \tau_{2L}^{2L} + R_{1L}^{U} \tau_{1L}^{1L} + R_{0R}^{U} \tau_{0R}^{3RR} \\ \end{aligned}$$

The rear legs are substituted zero normal forces in balance equations 4.20 and 4.8, respectively, in order to find the normals on the front critical contact line,

$$F_{nSRF} = \frac{-M_3(2)}{D_1(2)}$$
(5.4)

$$F_{nSLF} = \frac{-M_1(2)}{B_2(2)}$$
(5.5)

The critical moment required to turn the rover over the front side takes place when the rear legs are uncontact with ground and the equations 5.4 and 5.5 are substituted in equation F.8, we obtain

$$\begin{aligned} \tau_{C,Front} &= r_{4RF}^{U} \times \left(R_{SRF}^{U} F_{SRF}^{SRF}\right) + r_{4LF}^{U} \times \left(R_{SLF}^{U} F_{SLF}^{SLF}\right) + \\ & r_{c,4RF}^{U} \times \left(R_{4RF}^{U} f_{4RF}^{4RF}\right) + r_{c,3RF}^{U} \times \left(R_{3RF}^{U} f_{3RF}^{3RF}\right) + r_{c,4RR}^{U} \times \left(R_{4RR}^{U} f_{4RR}^{4RR}\right) + r_{c,3RR}^{U} \times \left(R_{3RR}^{U} f_{3RR}^{3RR}\right) + \\ & r_{c,2R}^{U} \times \left(R_{2R}^{U} f_{2R}^{2R}\right) + r_{c,1R}^{U} \times \left(R_{1R}^{U} f_{1R}^{1R}\right) + r_{c,0R}^{U} \times \left(R_{0R}^{U} f_{0R}^{0R}\right) + r_{c,1L}^{U} \times \left(R_{1L}^{U} f_{1L}^{1L}\right) + r_{c,2L}^{U} \times \left(R_{2L}^{U} f_{2L}^{2L}\right) + \\ & r_{c,3LF}^{U} \times \left(R_{3LF}^{3LF} f_{3LF}^{3LF}\right) + r_{c,4LF}^{U} \times \left(R_{4LF}^{U} f_{4LF}^{4LF}\right) + r_{c,3RR}^{U} \times \left(R_{3LR}^{U} f_{3LR}^{3LR}\right) + r_{c,4LR}^{U} \times \left(R_{4LR}^{U} f_{4LR}^{4LR}\right) + \\ & R_{4RF}^{U} \tau_{4RF}^{4RF} + R_{3RF}^{U} \tau_{3RF}^{3RF} + R_{4RR}^{U} \tau_{4RR}^{4RR} + R_{3RR}^{U} \tau_{3RR}^{3RR} + R_{4LF}^{U} \tau_{4LF}^{4LF} + R_{3LF}^{U} \tau_{3LF}^{3LF} + R_{4LR}^{U} \tau_{4LR}^{4LR} + R_{3LR}^{U} \tau_{3LR}^{3LR} + \\ & R_{2R}^{U} \tau_{2R}^{2R} + R_{1R}^{U} \tau_{1R}^{1R} + R_{2L}^{U} \tau_{2L}^{2L} + R_{1L}^{U} \tau_{1L}^{1L} + R_{0R}^{U} \tau_{0R}^{0R} \end{aligned}$$

The right legs are substituted zero normal forces in balance equations 4.26 and 4.20, respectively, in order to find the normals on the left critical contact line,

$$F_{nSLF} = \frac{-M_4(3)}{E_3(3)}$$
(5.7)

$$F_{nSLR} = \frac{-M_3(3)}{D_3(3)}$$
(5.8)

The critical moment required to turn the rover over the left side takes place when the right legs are uncontact with ground and the equations 5.7 and 5.8 are substituted in equation F.8, we obtain

$$\begin{split} \tau_{C,Leff} &= r_{4LF}^{U} \times \left(R_{SLF}^{U} F_{SLF}^{SLF}\right) + r_{4LR}^{U} \times \left(R_{SLR}^{U} F_{SLR}^{SLR}\right) + \\ &\quad r_{c,4RF}^{U} \times \left(R_{4RF}^{U} f_{4RF}^{4RF}\right) + r_{c,3RF}^{U} \times \left(R_{3RF}^{U} f_{3RF}^{3RF}\right) + r_{c,4RR}^{U} \times \left(R_{4RR}^{U} f_{4RR}^{4RR}\right) + r_{c,3RR}^{U} \times \left(R_{3RR}^{U} f_{3RR}^{3RR}\right) + \\ &\quad r_{c,2R}^{U} \times \left(R_{2R}^{U} f_{2R}^{2R}\right) + r_{c,1R}^{U} \times \left(R_{1R}^{U} f_{1R}^{1R}\right) + r_{c,0R}^{U} \times \left(R_{0R}^{U} f_{0R}^{0R}\right) + r_{c,1L}^{U} \times \left(R_{1L}^{U} f_{1L}^{1L}\right) + r_{c,2L}^{U} \times \left(R_{2L}^{U} f_{2L}^{2L}\right) + \\ &\quad r_{c,3LF}^{U} \times \left(R_{3LF}^{U} f_{3LF}^{3LF}\right) + r_{c,4LF}^{U} \times \left(R_{4LF}^{U} f_{4LF}^{4LF}\right) + r_{c,3LR}^{U} \times \left(R_{3LR}^{U} f_{3RR}^{3LR}\right) + r_{c,4LR}^{U} \times \left(R_{4LR}^{U} f_{4LR}^{4LR}\right) + \\ &\quad R_{4RF}^{U} \tau_{4RF}^{4RF} + R_{3RF}^{U} \tau_{3RF}^{3RF} + R_{4RR}^{U} \tau_{4RR}^{4R} + R_{3RR}^{U} \tau_{3RR}^{3RR} + R_{4LF}^{U} \tau_{4LF}^{4LF} + R_{3LF}^{U} \tau_{3LF}^{3LF} + R_{4LR}^{U} \tau_{4LR}^{4LR} + R_{3LR}^{U} \tau_{3LR}^{3LR} + \\ &\quad R_{2R}^{U} \tau_{2R}^{2R} + R_{1R}^{U} \tau_{1R}^{1R} + R_{2L}^{U} \tau_{2L}^{2L} + R_{1L}^{U} \tau_{1L}^{1L} + R_{0R}^{0R} \tau_{0R}^{0R} \end{split}$$

The front legs are substituted zero normal forces in balance equations 4.26 and 4.14, respectively, in order to find the normals on the rear critical contact line,

$$F_{nSRR} = \frac{-M_4(2)}{E_2(2)}$$
(5.10)

$$F_{nSLR} = \frac{-M_2(2)}{C_3(2)}$$
(5.11)

The critical moment required to turn the rover over the rear side takes place when the front legs are uncontact with ground and the equations 5.10 and 5.11 are substituted in equation F.8, we obtain

$$\begin{split} \tau_{C,Rear} &= r_{4RR}^{U} \times \left(R_{SRR}^{U} F_{SRR}^{SRR}\right) + r_{4LR}^{U} \times \left(R_{SLR}^{U} F_{SLR}^{SLR}\right) + \\ &\quad r_{c,4RF}^{U} \times \left(R_{4RF}^{U} f_{4RF}^{4RF}\right) + r_{c,3RF}^{U} \times \left(R_{3RF}^{U} f_{3RF}^{3RF}\right) + r_{c,4RR}^{U} \times \left(R_{4RR}^{U} f_{4RR}^{4RR}\right) + r_{c,3RR}^{U} \times \left(R_{3RR}^{U} f_{3RR}^{3RR}\right) + \\ &\quad r_{c,2R}^{U} \times \left(R_{2R}^{U} f_{2R}^{2R}\right) + r_{c,1R}^{U} \times \left(R_{1R}^{U} f_{1R}^{1R}\right) + r_{c,0R}^{U} \times \left(R_{0R}^{U} f_{0R}^{0R}\right) + r_{c,1L}^{U} \times \left(R_{1L}^{U} f_{1L}^{1L}\right) + r_{c,2L}^{U} \times \left(R_{2L}^{U} f_{2L}^{2L}\right) + \\ &\quad r_{c,3LF}^{U} \times \left(R_{3LF}^{U} f_{3LF}^{3LF}\right) + r_{c,4LF}^{U} \times \left(R_{4LF}^{U} f_{4LF}^{4LF}\right) + r_{c,3LR}^{U} \times \left(R_{3LR}^{U} f_{3LR}^{3LR}\right) + r_{c,4LR}^{U} \times \left(R_{4LR}^{U} f_{4LR}^{4RF}\right) + \\ &\quad R_{4RF}^{U} \tau_{4RF}^{4RF} + R_{3RF}^{U} \tau_{3RF}^{3RF} + R_{4RR}^{U} \tau_{4RR}^{4RR} + R_{3RR}^{U} \tau_{3RR}^{3RR} + R_{4LF}^{U} \tau_{4LF}^{4LF} + R_{3LF}^{U} \tau_{3LF}^{3LF} + R_{4LR}^{U} \tau_{4LR}^{4LR} + R_{3LR}^{U} \tau_{3LR}^{3LR} + \\ &\quad R_{2R}^{U} \tau_{2R}^{2R} + R_{1R}^{U} \tau_{1R}^{1R} + R_{2L}^{U} \tau_{2L}^{2L} + R_{1L}^{U} \tau_{1L}^{1L} + R_{0R}^{U} \tau_{0R}^{0R} \end{split}$$

The on-line executions of set of manipulations and locomotions are presented here over various types of surface geometries; this chapter studies and analyzes the normal forces, platform attitude, inertial effects, gravity forces, and dynamic stability margin for different surface geometries, and variable inertial accelerations, movable rover configurations under the considerations of being symmetric or non-symmetric form. In the case of symmetric configuration, the rover attitude (roll, pitch, an yaw) harmonizes the surface geometries, otherwise the joint configurations importantly contribute in attitude calculations. This chapter covers important examples provided with tests required to integrate all factors with each others in algorithmic and computational manner to deeply study their influence on dynamic stability. 1. Wheels, RCJ, LCJ, RDJ, and LDJ motions on flat surface.

This example studies the effect of acceleration of wheels and variable configuration of joints. The rover locomoted forward on a flat surface and subjected to three tests done in wheels accelerations 2, 4, and  $5m/s^2$  as represented in black, green, and blue curves, respectively. In addition, the rover configurations of four manipulators are manipulated in symmetric manner as shown in table 5.1:

**Table 5.1.**  $\operatorname{conf}_1 \rightarrow \operatorname{conf}_2$ 

	$\theta_{1R}$	$\theta_{2R}$	$\theta_{3RF}$	$\theta_{3RR}$	$\theta_{\rm 1L}$	$\theta_{_{2L}}$	$\theta_{\rm 3LF}$	$\theta_{3LR}$
conf_1	0	$\frac{\pi}{3}$	$-\frac{\pi}{3}$	$\frac{\pi}{3}$	0	$\frac{\pi}{3}$	$\frac{\pi}{3}$	$-\frac{\pi}{3}$
conf_2	0	0	$-\frac{\pi}{4}$	$\frac{\pi}{4}$	0	0	$\frac{\pi}{4}$	$-\frac{\pi}{4}$

Moreover, the platform attitude (roll, pitch, and yaw) will be congruent with the flat surface as shown in Figure 5.2



Figure 5.2. Platform attitude.

Furthermore as being symmetric and forward locomotion, the normal forces exerted on the wheels are distributed in equal manner; the front legs share the same value; and rear legs as well, as shown in Figure 5.3. The effects of normal forces were significantly propagated from outermost link (wheels) into innermost links (platform).



Figure 5.3 shows the front legs having the same vales, which increases in direct proportional to wheeled accelerations and shoulders' angle, while the rear legs were decreasing with respect the mentioned factors, wheel accelerations  $4m/s^2$  made the rear legs with 0.5294 Newton as normal forces at time 200s,

and  $5m/s^2$  made the rear legs without contact with ground at times 123 second. The loss of rear legs' connections with ground endangers the situation and threatens the rover's stability; because the rover will undergo to lateral tumbling about the front legs while the acceleration value was high.

In Figure 5.4, shows six dash curves which are critical moments where the universal moment of platform (solid curves) must not touch the critical curves in order to keep the system stable; otherwise the rover will tumble losing its stability. The upper three dash curves indicate for critical moments required to tumble about the rear legs, and the lower three dash curves indicate for critical moments required to tumble the rover about the front legs.

The solid curves were firstly relatively far from the rear critical curves when shoulders were open with 120° angle and conjunctional joints were manipulated with 60°, but they were coming approach to the front critical curves when shoulders joined with angle 90° and 0° conjunctional joints; solid black curve was far the dash black curve during the travel time, this indicates for the dynamic stable system and the four legs are in contact with ground. While solid green curve was trying to touch the dash green curve during time interval [160-200s]; this indicates for critical dynamical stability where the rear preserved small pressure on surface. Finally, the solid blue curve touched the dash blue curve at time 123 second; this indicates for dynamic instability where the rear legs lost the contacts with ground.



Figure 5.4. Universal moments and critical moments about  $z_U$  axis.

In Figure 5.5, the universal moment about the yu-axis is fixed and zero for three acceleration values as shown in solid curves, but the right and left critical curves were coming approach when the shoulders were coming approach to each other, and the RCJ and LCJ were approaching to zero angle.



Figure 5.5. Universal moments and critical moments about  $y_U$  axis.

Since the rover was manipulated in symmetric manner in this case, the effects of torques exerted on RCJ, LCJ, RDJ, and LDJ are cancelled as a result for being moving in the same magnitudes and in opposite rotations with respect to universal frame. The only effects of joint torques are those exerted on wheels which propagated in serial form from outermost link (wheels) to innermost link (platform); link by link. See Table 5.2 and Figure 5.6.

Accelerations in m/s <sup>2</sup>	$ au_{4\mathrm{RF}}^{4\mathrm{RF}}$ (N.m)	$ au_{4RR}^{4RR}$ (N.m)	$ au_{4LF}^{4LF}$ (N.m)	$ au_{4LR}^{4LR}$ (N.m)
2	0.0338	0.0338	-0.0338	-0.0338
4	0.0677	0.0677	-0.0677	-0.0677
5	0.0846	0.0846	-0.0846	-0.0846

Table 5.2. Torques exerted on wheel.



**Figure 5.6.** Propagated torques about  $(x_U, y_U, z_U)$  axes.

The gravity forces of link center of masses have no moment effect on the platform, because the rover attitude is congruent with the flat surface. See Figure 5.7.



Figure 5.7. Propagated moment of gravity forces about  $(x_U, y_U, z_U)$  axes.

However, the propagated moments of the inertial forces of wheels have significant effects on platform as shown in Figure 5.8.



Figure 5.8. Propagated moment of inertial forces about  $(x_U, y_U, z_U)$  axes.

Finally, the normal forces exerted on wheels create moments about the universal frames as shown in Figure 5.9



Figure 5.9. Propagated moment of normal forces about  $(x_U, y_U, z_U)$  axes.

2. Wheels, RFDJ, and RRDJ motions on step flat-inclined surface.

This example studies the effect of rover configurations on step flatinclined surface as shown in Figure 5.10; the right wheels locomoted on flat surface and left wheels locomoted on inclined surface with angle 22.5°. The rover locomoted forward and subjected to three tests done in RFDJ and RRDJ as represented in black, green, and blue curves with wheel acceleration 2 m/s<sup>2</sup>, respectively, and as shown in Table 5.3:



Figure 5.10. Rover posture on step flat-inclined surface.

	$\theta_{1R}$	$\theta_{2R}$	$\theta_{3RF}$	$\theta_{3RR}$	$\theta_{1L}$	$\theta_{_{\rm 2L}}$	$\theta_{\rm 3LF}$	$\theta_{3LR}$
conf_0	0	0	$-\frac{\pi}{4}$	$\frac{\pi}{4}$	0	0	$\frac{\pi}{4}$	$-\frac{\pi}{4}$
conf_1	0	0	$-\frac{\pi}{8}$	$\frac{\pi}{8}$	0	0	$\frac{\pi}{4}$	$-\frac{\pi}{4}$
conf_2	0	0	$-\frac{\pi}{18}$	$\frac{\pi}{18}$	0	0	$\frac{\pi}{4}$	$-\frac{\pi}{4}$

**Table 5.3.** conf\_0  $\rightarrow$  conf\_0, conf\_0  $\rightarrow$  conf\_1, conf\_0  $\rightarrow$  conf\_2

In this example, the first test is fix symmetric configuration as shown in above table, but the configurations of four manipulators are manipulated in non-symmetric manner in test 2 and test 3 as a result of rotation of right shoulders (RDJ). The non-symmetric manner and the variance of elevations on right and left sides significantly influence in platform attitude where the whole rover undergoes under roll rotations as expressed in  $y_U$  axis as shown in Figure 5.11:





The calculations take the three contact legs into account as being nonsymmetric configurations. Therefore, this example assumes right rear legs without contact with surface.



Figure 5.12. Normal forces.

Above Figure shows the blue curve of right front leg becoming without contact at times 118 second. It means that when the right shoulders were closing to each other making the right front leg without contact with surface, the rover rotated about single line delimited by the left legs. This process threatens the dynamic rover stability as shown in Figure 5.13; the solid blue curve touched the dash line at 118 second, thus the adopting of conf\_2 (test 3) will lead to dynamic unstable system. The solid green curve was trying to approach from the lower dash green curve, thus it is about to reached to critical

dynamic stable system. Finally the black curves are relatively far from each other and this indicates for fully dynamically stable system.



Figure 5.13. Universal moments and critical moments about  $y_U$  axis.

The locomotion of wheels and the manipulation of RDJ yielded torques propagated into universal platform frame as shown in Figure 5.14. As well as the constant black curves indicate for constant wheels torques and symmetric manner. However, the interior manipulations in shoulders disturbed the symmetric form and add manipulation effects as shown in figure bellow.



Figure 5.14. Propagated torques about  $(x_U,y_U,z_U)$  axes.

The gravity forces of link center of masses yielded moment effect on the platform, because the rover attitude is rotating about  $y_U$ -axis as seen in Figure 5.15.



Figure 5.15. Propagated moment of gravity forces about  $(x_U, y_U, z_U)$  axes.

The moment of inertial force is resulted from the locomotive wheels and manipulation of RDJ, Figure 5.16 shows that the black curves are constant values because of symmetric manner and fixed manipulations, while the rest curves are variable with respect to RDJ manipulations.



Figure 5.16. Propagated moment of inertial forces about  $(x_U, y_U, z_U)$  axes.

Finally, the effect of normal forces is simulated in Figure 5.17 which shows the black curves with constant values because of constant normal forces, continuous connection with surface during the travel, and fixed manipulation. However, the blue curves suddenly and significantly changed at time 118s as a result of discontinuity occurred between the right front leg and surface; In fact as explained previously, this time the rover lost its dynamic stability.



Figure 5.17. Propagated moment of normal forces about  $(x_U, y_U, z_U)$  axes.

#### 3. Wheels, RDJ, and LDJ Motions on Inclined Surface.

This example studies the effect of rover elevation on inclined surface with angle 20.7° as shown in Figure 5.18; the rover locomoted forward and subjected to three tests done in RDJ and LDJ as represented in black, green, and blue curves with wheel acceleration  $2.5 \text{ m/s}^2$  as shown in Table 5.4:



Figure 5.18. Rover's shoulders closing on inclined surface.

	$\theta_{1R}$	$\theta_{2R}$	$\theta_{3RF}$	$\theta_{3RR}$	$\theta_{1L}$	$\theta_{_{2L}}$	$\theta_{3LF}$	$\theta_{3LR}$
conf_0	0	0	$-\frac{\pi}{4}$	$\frac{\pi}{4}$	0	0	$\frac{\pi}{4}$	$-\frac{\pi}{4}$
conf_1	0	0	$-\frac{\pi}{9}$	$\frac{\pi}{9}$	0	0	$\frac{\pi}{9}$	$-\frac{\pi}{9}$
conf_2	0	0	$-\frac{\pi}{18}$	$\frac{\pi}{18}$	0	0	$\frac{\pi}{18}$	$-\frac{\pi}{18}$

**Table 5.4.** conf\_0  $\rightarrow$  conf\_0, conf\_0  $\rightarrow$  conf\_1, conf\_0  $\rightarrow$  conf\_2

For being symmetric configurations in three tests, the platform attitude was congruent to the inclination of surface irrespective to the shoulders opening or joining, as shown in Figure 5.19



Figure 5.19. Platform attitude.

Figure 5.20 shows that the front legs share the same normal forces and higher values in comparison to the rear legs, as a result of wheel accelerations and shoulders joining. On other words, high acceleration and lower shoulder angles yield pressure on the single line delimited by contact points of front legs. The following picture shows constant normal forces regarding to constant manipulations (black curve), and shows variable normal forces with respect to acceleration (green and blue curves). However, the front normal forces represented in blue curve shows constant normal forces after disconnection occurred between the rear legs and surface at time 137.



Figure 5.20. Normal forces.

Above Figure shows the blue curves denoting for rear legs were becoming without contact at times 137 second. It means that when the shoulders on both sides were closing to each other, the rover elevation with respect to inclined surface got higher and the pressure exerted on front legs got increase with taking into consideration the significant wheeled accelerations, and then the rover rotated about single line delimited by the front legs making the rear legs without contact with surface. This process threatens the dynamic rover stability as shown in Figure 5.21; the solid blue curve touched the dash line at 137 second, thus the adopting of third test will lead to dynamic unstable system. The solid green curve is trying nearly approaching from the dash green curve, thus it reached to critical dynamic stable system. Finally the black curves are relatively far from each other and this indicates for fully dynamically stable system.



Figure 5.21. Universal moments and critical moments about  $z_U$  axis.

The effects of propagated moments resulted from gravity forces, inertial forces exerted on the center of mass of links, normal forces exerted on wheels are simulated in Figure 2.22, 2.23, and 2.24 respectively.



Figure 5.22. Propagated moment of gravity forces about  $(x_U,y_U,z_U)$  axes.



Figure 5.23. Propagated moment of inertial forces about  $(x_U,y_U,z_U)$  axes.

Figure 5.24 shows significant and sudden change occurred in blue curve about  $z_U$  axis, as a result of losing the connection between rear legs and surface; and this simply simulates the unstable situation.



Figure 5.24. Propagated moment of normal forces about  $(x_U,y_U,z_U)$  axes.
4. Wheels motion on flat and inclined surface.

This example can be briefed in three periods; firstly, a flat period when the rover moved over flat surface and the platform attitude were congruent with surface. Secondly, a transition period when the rover suddenly faced an inclined surface with inclined angle 30° at time 100 second, the front wheels started to move on inclined surface while rear legs were still on flat surface and the platform attitude was under rotation. Finally, an inclined period when the rear legs traversed the flat surface and the platform attitude became congruent with inclined surface.

The configurations are in fixed symmetric forms with different open shoulders 90°, 45°, 20°, respectively, as shown in Table 5.5.

	$\theta_{1R}$	$\theta_{2R}$	$\theta_{3RF}$	$\theta_{3RR}$	$\theta_{\rm 1L}$	$\theta_{_{2L}}$	$\theta_{3LF}$	$\theta_{3LR}$
conf_0	0	0	$-\frac{\pi}{4}$	$\frac{\pi}{4}$	0	0	$\frac{\pi}{4}$	$-\frac{\pi}{4}$
conf_1	0	0	$-\frac{\pi}{8}$	$\frac{\pi}{8}$	0	0	$\frac{\pi}{8}$	$-\frac{\pi}{8}$
conf_2	0	0	$-\frac{\pi}{18}$	$\frac{\pi}{18}$	0	0	$\frac{\pi}{18}$	$-\frac{\pi}{18}$

**Table 5.5.**  $\operatorname{conf}_0 \rightarrow \operatorname{conf}_0$ ,  $\operatorname{conf}_1 \rightarrow \operatorname{conf}_1$ ,  $\operatorname{conf}_2 \rightarrow \operatorname{conf}_2$ 

The time delay between the front wheel and rear leg can be computed as fellow

$$t_{delay\_R} = \sqrt{\frac{-a_3 \sin\theta_{3RF} + a_3 \sin\theta_{3RR}}{vv \times 0.5}}$$
$$t_{delay\_L} = \sqrt{\frac{a_3 \sin\theta_{3LF} - a_3 \sin\theta_{3LR}}{vv \times 0.5}}$$

Where, vv is wheel acceleration value, in this example  $vv = 0.5 \text{ m/s}^2$  is chosen small in order to make the transition period longer and to study the comparisons clearly and precisely. The three tests have 1.5042, 1.2649, and 0.7454 second as time delays between front and rear legs. During the transition period, the rover undergoes to pitch rotation. The time of rotation is a function of shoulder angle and wheel acceleration. In Figure 5.25, the black curve stands for first test and it takes longer rotation time; and blue curve accomplishes its rotation faster. Then after the rear legs traversed the flat surface, the steady pitch attitude takes place and the whole rover becomes congruent with the surface inclination.



As being symmetric form and moving on flat surface, the normal forces for front legs were equally greater than the rear legs as a result of inertial effects. However at the time of contacts with incline surface, the normal forces of front legs were gradually decreasing during the transition period as shown in Figure 5.26. The front legs represented in blue curve shows it becoming zero during the transition period exactly, whereas the rear legs became fully responsible for the rover heaviness see Figure 5.27.



Figure 5.26. Normal forces.



Figure 5.27. Rear legs lost the contact with ground.

Figure 5.28 shows the simulation of dynamic stability for three tests; in first test represented in solid black curve which is far from the critical curves; while green curves were somehow close trying to reach the critical situation; and finally the dynamic instability occurred in third test represented in solid blue curve touching the rear critical curves represented in dash blue curve during the time of transition period; where it shows the universal moment at platform equal the rear critical moment. Thus, the open shoulders with 20° is not capable for moving from flat to inclined surface with an angle 30°. The zooming for transition period keeps a small distance that separates the universal moments and the critical moments for all boarders, else rear critical moments regarding to third test.



Figure 5.28. Universal moments and critical moments about  $z_U$  axis.

5. Wheels motion on sinusoidal surface.

This example studies the effect of wheel acceleration locomoted on sinusoidal surface subjected to three tests done in wheels accelerations 1, 2, and  $3 \text{ m/s}^2$  as represented in black, green, and blue curves, respectively. In addition, the rover configurations of four manipulators were manipulated in fixed symmetric manner as shown in Table 5.6:

**Table 5.6.**  $\operatorname{conf}_0 \rightarrow \operatorname{conf}_0$ 

	$\theta_{1R}$	$\theta_{2R}$	$\theta_{3RF}$	$\theta_{3RR}$	$\theta_{1L}$	$\theta_{\rm 2L}$	$\theta_{3LF}$	$\theta_{3LR}$
conf_0	0	0	$-\frac{\pi}{6}$	$\frac{\pi}{6}$	0	0	$\frac{\pi}{6}$	$-\frac{\pi}{6}$

There is fixed 60° angle between open shoulders on both sides, thus the delay time between the front and rear legs are 0.8944, 0.6325, and 0.5164 second. The rear legs share the elevations of front legs after the elapse of those delay times. The pitch orientations for three speeds were simulated in Figure 5.29; it shows rover was ascending the sinusoidal surface with negative angle and descending with positive angle with same amplitude for three speeds and different time delay between those speeds. Zero amplitude of pitch angle denotes the top of concave and bottom of convex.



Figure 5.29. Pitch angle.

The normal forces for front legs were equal and greater than equal rear legs in ascending and descending locomotion as a result of high acceleration effect of wheels as shown in Figure 5.30. The amounts of wheel acceleration are chosen large enough to overcome the gravity force which decelerates the rover at ascending motion, and in order to study their effects on dynamic stability. The normal forces of rear legs with highest speed  $(3 \text{ m/s}^2)$  represented in blue curve were zero during the travel else in the bottom of convex when the rover moves half ascending travel where the pitch angle -11.3°; while regarding to second test  $(2 \text{ m/s}^2)$  represented in green, the normal forces of

rear legs became zero for shorter time in comparison with blue and green curves as a result of less acceleration.



Figure 5.30. Normal forces.

Figure 5.31 shows that the universal moment far from the critical moments in the case of black curves  $(1 \text{ m/s}^2)$ , so that this test is considered dynamically stable throughout the travel. In the case of 2 m/s<sup>2</sup> the solid green curve was touch the lower dash curve some part of travel time, thus the second test is considered dynamic instable. However, in the case 3 m/s<sup>2</sup> the universal moment represented in solid blue curve touched the dash blue curve

most of the time and this concludes the dynamic unstable system, else the periodic time interval shown in Figure 5.32.



Figure 5.31. Universal moments and critical moments about  $z_U$  axis



Figure 5.32. Zooming for universal moments and critical moments about  $z_U$  axis

The torques exerted on wheels are simulated for three tests as shown in Figure 5.33, those values are in direct proportional to wheel acceleration.



**Figure 5.33.** Propagated torques about  $(x_U, y_U, z_U)$  axes.

The effect of moment of inertial forces are simulated in constant curves as a result of constant configurations, and it in direct proportional to wheel acceleration as shown in Figure 5.34



Figure 5.34. Propagated moment of inertial forces about  $(x_U,y_U,z_U)$  axes.

The effect of moment of gravity forces are simulated in sinusoidal curves as a result of sinusoidal attitude, and it not in direct proportional to wheel acceleration, but platform attitude angles as shown in Figure 5.35



Figure 5.35. Propagated torques about  $(x_U, y_U, z_U)$  axes.

The effect of moment resulted from normal forces is simulated in Figure 5.36, and it is appeared in sinusoidal curves. The third test yielded the highest moment about  $z_U$  axis as a result of highest acceleration. The top of concave is non-uniform due to the normal force constraint.



Figure 5.36. Propagated moment of normal forces about  $(x_U,y_U,z_U)$  axes.

6. Wheels, RDJ, LDJ motions on random surface.

This example studies the effect of wheel acceleration locomoted on random surface subjected to two tests done in wheels accelerations 0.05 and 1  $m/s^2$  as represented in black and blue curves, respectively. In addition, the rover configurations of four manipulators are manipulated in symmetric manner as shown in Table 5.7:

**Table 5.7.**  $conf_0 \rightarrow conf_1$ 

	$\theta_{1R}$	$\theta_{2R}$	$\theta_{3RF}$	$\theta_{3RR}$	$\theta_{\rm 1L}$	$\theta_{_{2L}}$	$\theta_{3LF}$	$\theta_{3LR}$
conf_0	0	0	$-\frac{\pi}{4}$	$\frac{\pi}{4}$	0	0	$\frac{\pi}{4}$	$-\frac{\pi}{4}$
conf_1	0	0	$-\frac{\pi}{6}$	$\frac{\pi}{6}$	0	0	$\frac{\pi}{6}$	$-\frac{\pi}{6}$

This example considered symmetric configurations where the right and left shoulders were joined closely from 90° till 60°. Figure 5.37 simulated the platform orientation with respect to universal frame; it reflects the geometry of random elevations surface; the rover moved on flat surface, and ascended and descended non-uniform surface; the rear legs moved on the front elevations after delay time 1 second; In addition, it shows the platform subjected to three kinds of rotations: 1) Zero attitude, where the platform frame is contingent with universal frame during the interval [0-16s] and [111-134s] and [179-200s].

Clockwise rotations during the interval [17-110s], maximum angle 55.2° at time 100s.

3) Counter-clockwise rotations during the interval [135-200s], maximum positive angle reached to 48.26° at time 178s.



Figure 5.37. Platform pitch angle.

Figure 5.38 simulated the normal forces exerted on four legs for two tests represented in black and blue curve. The front legs represented in black curves were without contact during the interval [86-100s] and rear legs were without contacts during the time interval [169-178s]. While the front legs represented in blue curve were in contact with surface during travel times but the rear legs were without contacts during the time interval [164-178s].



Figure 5.38. Normal forces

Figure 5.39 simulated the universal moments exerted on platform represented in solid curves, and front and rear critical moments represented on dash curves. It shows the solid black curve  $(0.05 \text{ m/s}^2)$  touched the rear critical moment during time interval [86-100s] In other words, the front legs were without contact with surface and the whole rover rotated about the rear legs, thus the rover was dynamically unstable system during this interval. It also shows the solid black curve touched the front critical moment during the time interval [169-178s], and analytically it means that the rover rotated about the

front legs making the it dynamically unstable system. Moreover, it shows the solid blue curve  $(1 \text{ m/s}^2)$  touched the front critical moment during time interval [164-178s], therefore, the rover is dynamically unstable during this interval.



Figure 5.39. Universal moments and critical moments about  $z_U$  axis

These imply the advantages and disadvantages of inertial accelerations of wheels and the rover configurations. The higher acceleration (1 m/s2) positively sustained the stability during interval [86-100s], but it negatively speeded the process of instability during interval [164-178s], whereas less acceleration  $(0.05 \text{ m/s}^2)$  delayed the instability for 5 second.

According to rover configurations, it was used the same configurations for the two acceleration tests, and it shows that open shoulders with 90° as shown in time 0s is much safer and secure for keeping a significant distance between the universal moments and critical moments curves of both sides, while open shoulders with 60° endangered the system for keeping small distance between the universal moments and critical moments of both sides. Therefore, imposing control on configurations and accelerations evades the danger of tumbling.

### **Chapter Six**

### 6. Conclusion

This thesis exhibits a new mechanical design for a quadruped mobile robot. The four identical wheeled legs were gaining high level coordinations in various aspects. This feature contributed in increasing the rover speed stably and smoothly on uneven terrain. Besides, this work inherited the advantages and eliminated the drawbacks of both legged and wheeled locomotion in computational manner, for being equipped with wheels and legs simultaneously. Thus, the platform a base link undergoes under a smooth and soft locomotion in relative to four wheeled-legged manipulators and surface geometries.

The platform attitudes were evaluated with respect to platform universal frame. The changes occurred on joint configurations and different ground elevations disturb the symmetric posture, and rotate the platform smoothly leaving the universal axis by roll, pitch, and yaw angle. Newton-Euler Recursive method was employed, and it provided an online monitoring system for the sources of dynamic forces and moments exerted on each link of the four manipulators. The decomposition of universal forces and moments made the point clearer throughout studying the source of each force and moment exerted on the universal frame. The universal moment, which acts about platform link of the rover, is resulted from the normal forces acted at wheels, gravity forces, inertial forces and torques exerted on the center of mass of each link. When rover faced random surface during motion, a change has been occurred in dynamic disturbances at the wheels generating considerable moments about the platform link expressed in universal frame.

Because four legs are considered indeterminate system, in this thesis the normal forces were evaluated for three contact legs in the case the nonsymmetric rover. However, in the case of symmetric configurations the normal forces are distributed equally between the sides which sharing the same the inertial forces, ground geometries, and platform attitude. Thus regarding to symmetric four legs, normal forces were evaluated by considering two legs sharing the same value. The results simulated the effect of high acceleration on the connectivity between wheels and surface. A new dynamic stability criterion was presented for rover operating arbitrary on various shapes of surfaces, and variable rover configurations. In addition, this criterion provided on-line calculations for the effect of rover configurations, various surface geometry, platform attitudes, kinematic values, dynamic effects, and variable ground normal forces.

The gravity force is static feature, and it is not influenced with acceleration at all, but its moment significantly effects on the dynamic stability in the presence of changing in platform attitude. While inertial force is dynamic feature, and it is not influenced with ground geometry and platform attitude, and it significantly effects on the dynamic stability.

The simulation model was presented for a various examples exploiting MatLab which provided on-line calculations for predicting the behavior of a physical system under a variety of surface geometries and rover configurations.

In future work, inverse kinematics can be exploited for determining the generalized coordinates (angles of joints), and then evaluating the required rover configurations to enable the uncontact leg to select its foothold on surface. Furthermore, the platform attitudes can be evaluated as function of rover configurations, surface geometries, and dynamic forces and moments. In addition, normal forces exerted on four legs should be evaluated in the case of non-symmetric manner in future work.

#### References

- 1. Wettergreen, D., Thorpe, C., Whittaker, R., "*Exploring Mount Erebus by walking robot*", Robotics and Autonomous Systems, vol. 11, pp. 171-85, December 1993.
- Peterson, K. and Ward, C., "An Autonomous Mobile robot to Perform Waste Drum Inspections", Robotics and manufacturing – Recent Trends in Research, Education, and Applications, ASME Press, vol 5, pp. 637-641, 1994.
- 3. Gonzalez de Santos, P., Garcia, E., Estremera, J. and Armada, M. A., "SILO6: Design and configuration of a legged robot for humanitarian demining", IARP Workshop on Robots for Humanitarian Demining, Vienna, Austria, 3-5 November 2002.
- 4. Papadopoulos, E. and Sarkar, S., "On the Dynamic Modeling of an Articulated Electrohydraulic Forestry Machine", Proc. of the 1996 AIAA Forum on Advanced Developments in Space Robotics, Madison, WI, August 1-2, 1996.
- 5. Iagnemma, K., Shibly, H., and Dubowsky, S., "On-line Terrain Parameter Estimation for Rovers", IEEE International Conference on Robotics and Automation, Washington D.C, USA, 2002.
- James, J. Z., David, B. M., Jeffrey M. W., June F. Z., Fred B. O., Kelly J. M., Gerald, M. H., Phillip A., Dennis J. E., and Thomas, W. G., *"Exploration Rover Concepts and Development Challenges"*, NASA TM-2005-213555, AIAA-2005-2525, Orlando, Florida, March 2005.
- 7. Calafiore, G., Indri, M., Bona, B., "*Robot Dynamic Calibration: Optimal Excitation Trajectories and Experimental Parameter Estimation*", Journal of Robotic Systems, Torino, Italy, vol. 18, no. 2, pp. 55-68, 2001.

- 8. Sujan, A. V., and Dubowsky, S., "An Optimal Information Method for Mobile Manipulator Dynamic Parameter Identification", IEEE/ASME Transactions in Mechatronics, vol. 2, no. 2, June 2003.
- Fukuoka, Y., Kimura, H., and Cohen, A., "Adaptive Dynamic Walking of a Quadruped Robot on Irregular Terrain Based on Biological Concepts", The International Journal of Robotics Research, vol. 22, no. 3-4, pp. 187-202, March 2003.
- 10.Goswami, A., "Postural stability of biped robots and the foot rotation indicator (FRI) point", International Journal of Robotics Research, vol. 18, no. 2, pp. 523-533, 1999.
- 11. Papadopoulos, E., Sarkar, S., "*The Dynamic of an Articulated Forestry Machine and its Applications*", Proceedings of the IEEE International Conference on Robotics and Automotion, pp. 323-328, Albuquerque, NM, April 1997.
- 12. Matijevic J., "Mars Pathfinder Microrover Implementing a Low Cost Planetary Mission Experiment", Proceeding of the Second IAA International Conference on Low-Cost Planetary Missions, John Hopkins Applied Physics Laboratory, Maryland, USA, pp IAA-L-0510, April 1996.
- 13.Carr, M., "Workshop on Mobility", Report. Ames Research Center, July 2, 1995.
- 14. Hayati, S., Volpe, R., Backes, P., Balaram, J., and Welch, W., "*Microrover Research for Exploration of Mars*", AIAA Forum on Advanced Developments in Space Robotics, 1996.
- 15.Mars Pathfinder: http://mars.jpl.nasa.gov.
- 16.Jeffrey S. Norris, Mark W. Powell, Marsette A. Vona, Paul G. Backes, Justin V. Wick., "Mars Exploration Rover Operations with the Science Activity Planner", Proceedings of the 2005 IEEE, International

Conference on Robotics and Automation, pp.4629-4634, Barcelona, Spain, April 2005.

- 17. Mishkin, A., Morrison, J., Nguyen, T., Stone, H., Cooper, B., and Wilcox, B., "*Experiences with operations and autonomy of the Mars Pathfinder Microrover*", IEEE Aerospace Conference, pp.337-51, 1998.
- 18.Stroupe, Ashley W., Singh, S., Simmons, R., Smith, T., Tompkins, P., Verma, V., Vitti-Lyons, R., Wagner, M., "TECHNOLOGY FOR AUTONOMOUS SPACE SYSTEMS", The Robotics Institute Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, CMU-RI-TR-00-02, September 2002. See page 76.
- 19.http://www.windows.ucar.edu/tour/link=/space\_missions/mars/mars\_expl oration rover/mer vehicles.html, May, 2004.
- 20.Iagnemma, K., Shibly, H., Rzepniewski, A., and Dubowsky, S., "Planning and Control Algorithms for Enhanced Rough-Terrain Rover Mobility", Proceedings of the Sixth International Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS, 2001.
- 21.Iagnemma, K., Rzepniewski, A., Dubowsky, S., Huntsberger, T., Schenker, P., "Mobile Robot Kinematic Reconfigurability for Rough-Terrain", Proceedings of the SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems III, Boston, September 2000.
- 22.Farritor, S., Zhang, J., "A Modular Robotic Infrastructure to Support Planetary Surface Operations", The American Institute Of Aeronautics and Astronautics, Department of Mechanical Engineering, University of Nebraska, Lincoln, USA, January 9, 2000.
- 23. Yoneda, K. and Hirose, S. "*Three-dimensional stability criterion of integrated locomotion and manipulation*". Journal of Robotics and Mechatronics, vol. 9, no. 4, pp. 267-274, 1997.

- 24.Lamon, P. and Siegwart, R., "Wheel Torque Control in Rough Terrain -Modeling and Simulation", In Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, 2005.
- 25.Meghdari, A., Mahboobi, S. H., and Gaskarimahalle, A. L., "Dynamic Modeling of 'CEDRA' Rescue Robot on Uneven Terrains", Proceedings of IMECE2004-59239, ASME International Mechanical Engineering Congress, Anaheim, California, 13-19 November 2004.
- 26.Hardarson, F., "Stability analysis and synthesis of statically balanced walking for quadruped robots". PhD thesis, The Royal Inst. of Technology, 100 44 Stockholm, Sweden, June 2002.
- 27.Michel, O., "Webots<sup>TM</sup>: Professional Mobile Robot Simulation", International Journal of Advanced Robotic Systems, vol. 1, no. 1, pp. 39-42, ISSN 1729-8806, 2004.
- 28.Lee, Y., Hirose, S., "Three-Legged Walking for Fault Tolerant Locomotion of a Quadruped Robot with Demining Mission", Proceedings of the IEEE/RSJ, International Conference on Intelligent Robots and Systems, pp. 973-978, 2000.
- 29. Hayati, S., Volpe, R., Backes, P., Balaram, J., Welch, R., Ivlev, R., Tharp, G., Peters, S., Ohm, T., Petras, R., Laubach, S., "*The Rocky 7 Rover: A Mars Sciencecraft Prototype*", IEEE International Conference on Robotics and Automation, pp. 2458–2464, 1997.
- 30.Garcia, E., Estremera, J., and Gonzalez-de-Santos, P., "A Classification of Stability Margins for Walking Robots", Industrial Automation Institute, Madrid, Spain, 2002.
- 31.Garcia, E., Estremera, J. and Gonzalez de Santos, P., "A comparative study on stability margins for walking machines". Robotica, vol. 20, pp. 595-606, 2002.

- 32.McGhee, R. B. and Frank, A. A., "On the stability of quadruped creeping gaits", Mathematical Biosciences, University of southern California, Los Angeles, California, vol. 3, pp. 331-351, October, 1968.
- 33. Messuri, D. and Klein, c., "Automatic body Regulation for Maintaining Stability of a Legged Vehicle During Rough-Terrain Locomotion", IEEE Journal of Robotics and Automation, vol. RA1, no. 3, pp.132-141, September, 1985.
- 34.Nagy, P. V., Desa, S., Whittaker, W. L., "Energy-Based Stability Measures for Reliable Locomotion of Statically Stable Walkers: Theory and Application", The International Journal of Robotics Research, vol. 13, no.3, pp. 272-287, June 1994.
- 35. Hirose, S. Tsukagoshi, H. and Yoneda, K., "Normalized energy stability margin: generalized stability criterion for walking vehicles", Proceedings of Int. Conf. On Climbing and Walking Robots, Brussels, pp. 71-76, November, 1998.
- 36.Hirose, S. Tsukagoshi, H. and Yoneda, K., "Normalized Energy Stability Margin and its Contour of Walking Vehicle on Rough Terrain", Proceedings of the IEEE, International Conference on Robotics and Automation (ICRA), Seoul, Korea, pp. 181-186 May 2001.
- 37.Gonzalez, P., Santos, de., Jimenez, M., A. and Armada, M.A., "Dynamic Effects in Statically Stable Walking Machines", Journal of Intelligent and Robotic Systems, vol. 23, no 1, pp. 71- 85, 1998.
- 38.Orin, D. E., McGhee, R. B., and Jaswa, V. C., "Interactive Computer-Control of a Six-Legged Robot Vehicle with Optimization of Stability, Terrain Adaptability, and Energy," Proc. of 1976 IEEE Conference on Decision and Control, Clearwater Beach, Florida, pp. 382-391, December, 1976.
- 39. Vukobratovic, M., Frank, A.A. and Juricic, D. "On the stability of byped locomotion", IEEE Transactions on Biomedical Engineering, vol. 17, no. 1, pp. 25-36. 1970.

- 40.Kang. D.O., Lee, Y.J., Lee, S.H, Hong, Y.S. and Bien, Z. "A study on an adaptive gait for a quadruped walking robot under external forces". Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, pp. 2777-2782, 1997.
- 41.Lin, B. S. and Song, S. M. "Dynamic modeling, stability and energy efficiency of a quadrupedal walking machine", IEEE Conference on Robotics and Automation, pp. 367-373, Atlanta, Georgia, 1993.
- 42. Yoneda, K. and Hirose, S., "Tumble Stability Criterion of Integrated Locomotion and Manipulation", Proceedings IROS 96, pp. 870-876, 1996.
- 43.Zhou, D., Low, K.H. and Zielinska, T. "A stability analysis of walking robots based on leg-end supporting moments". Proceedings of IEEE International Conference on Robotics and Automation. San Francisco, CA. pp. 2834-2839, April 2000.
- 44.Papadopoulos, E.G., Rey, D.A., "A New Measure of Tipover Stability Margin for Mobile Manipulators", In Proceedings of the IEEE Internal Conference on Robotics and Automation. Conf. on Robotics and Automation, pp. 3111-3116, 1996.
- 45. Papadopoulos, E. and Rey, D., "*The Force-Angle Measure of Tipover Stability Margin for Mobile Manipulators*," Vehicle System Dynamics, vol. 33, no 1, pp. 29-48, 2000.
- 46.Garcia, E. and González de Santos, P. "An improved energy stability margin for walking machines subject to dynamic effects", Robotica, vol 00, pp. 1-8, April, 2004.
- 47.Garcia, E. and Gonzalez de Santos, P. "A New Dynamic Energy Stability Margin for Walking Machines", International Conference on Advanced Robotics ICAR'03, Coimbra, Portugal, 30 June-3 July, 2003.

- 48.Ghasempoor A. and Sepehri N., "A Measure of Machine Stability for Moving Base Manipulators", IEEE International Conference on Robotics and Automation, Nagoya, Japan, pp. 2249-2254, 1995.
- 49.Luh, J. Y. S., Walker, M. W., Paul, R. P. C., "On-line computational scheme for mechanical manipulators", ASME Journal of Dynamic Systems, Measurement, and control, vol. 102, pp. 69-76, June 1980.
- 50.Paul, R., "Modeling, trajectory calculation and servoing of a computer controlled arm," AIM-177, Stanford University, Artificial Intelligence Laboratory, 1972.
- 51.Luh, J., Walker, M., and Paul, R., "*Resolved-Acceleration Control of Mechanical Manipulators*," IEEE Trans. Automation. Control, AC-25, no. 3, pp.468–474. 1980.
- 52.Hemami, H., Jaswa, V. C., and McGhee, R. B., "Some Alternative Formulations of Manipulator Dynamics for Computer Simulation Studies," Proc of 13th Allerton Conference on Circuit and System Theory, University of Illinois, October, 1975.
- 53.Beer, P., Johnston, R., "Vector Mechanics for Engineers: Statics and Dynamics", McGraw-Hill Companies, Sixth Editions, pp. 155, 1996.
- 54. Uicker "On the dynamic analysis of spatial linkages using 4x4 matrices", Ph.D Dissertation, Northwestern University, 1965.
- 55. Stepanenko, Y., and Vukobratovic, M., "Dynamics of Articulated Open Chain Active Mechanisms", Mathematical Biosciences, vol.28, pp.137-170, 1976.
- 56. Armstrong. W., "*Recursive solution to the equations of motions of an n-link manipulator*", in Proc. 5th World Congress on Theory of Machines and Mechanisms, Montreal, pp. 1343-1346, July 1979.
- 57.Kahn, M.E., "The Near Minimum Time Control of Open Loop Articulated Kinematic Chains", Stanford AI Lab, AI Memo 106, 1969.

- 58.Hollerbach, J., "A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," IEEE Trans. Syst. Man Cybern., vol. SMC-10, pp. 730–736, Nov. 1980.
- 59.Silver, W.M., "On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators,", Int. J. Robotics Research. vol. 1, no. 2, pp. 118-128, 1982.
- 60.Ploen, S.R., "Geometric Algorithms for the Dynamics and Control of Multibody Systems", PhD Dissertation, University of California IRVINE. pp. 34-36, 1997.
- 61. Walker, M. W., and Orin, D. E., "*Efficient dynamic computer simulation of robotic mechanisms*", ASME Journal of Dynamic Systems, Measurement and Control, vol. 104, pp. 205–211, 1982.
- 62.Corke, P.I., "A Robotics Toolbox for MATLAB", IEEE Robotics and Automation Magazine, 3(1):24-32, March 1996.
- 63.http://en.wikibooks.org/wiki/Trigonometry:Solving\_Trigonometric\_Equa tions, May, 2005.
- 64.Denavit, J. and Hartenberg, R. S., "A kinematic notation for lower-pair mechanisms based on matrices", ASME Journal of Applied Mechanics, vol 77, pp. 215-221, June 1955.
- 65. Shibly, H., "Performance Evaluation and Efficient Control of Trajectory Following Robots with Friction and Backslash", Ph.D. Thesis, Carnegie-Mellon University, Pittsburgh, USA, 1988.
- 66.Richard P. Paul., "*Robot manipulators: Mathematics, programming, and control*", the MIT Press, Cambridge, Massachusetts and London, England, the Massachusetts Institute of Technology, 1981.
- 67.Alonzo Kelly., "*Essential Kinematics for Autonomous Vehicles*", Carnegie Mellon University, The Robotics institute, CMU-RI-TR-94-14, May 1994. available at

http://www.frc.ri.cmu.edu/~alonzo/pubs/reports/pdf\_files/kinematics.pdf

# Appendices

#### Appendix A: Denavit-Hartenburg Convention

In 1955, Denavit and Hartenburg [64,62] constructed a novel technique for setting up orthonormal coordinate frames to a pair of adjacent links in an open kinematics chain. DH describes the kinematics of the robot by describing the position and orientation of each link with respect to the previous link. In a simple manner, each pair of successive joints is characterized by a distance between joint axes **a**, a twist between joint axes  $\alpha$ , an offset **d**, and a joint angle  $\theta$ .

Each joint axis [65] should be firstly labeled in each manipulator with a coordinate frame number. Starting from  $O_0$  as the base frame to  $O_n$  as the end-effector. The next step is to set up the three dimensional coordinate system. The  $z_i$  axis represents the motion of link i+1, so that it is assigned along the axis of rotation for revolute joint or in the direction of translation for prismatic joint. For parallel joint axis,  $z_i x z_{i-1}=0$ , the  $x_{i-1}$  axis is directed from frame  $O_i$  to  $O_{i-1}$ , and for intersecting joint axes, the  $x_{i-1}$  is directed to be perpendicular to

the plane or parallel to the vector cross product  $z_{i-1} \times z_i$ . The y-axis is defined in the direction needed to complete a right-handed orthonormal coordinate frame. The system ( $x_0$ ,  $y_0$ ,  $z_0$ ) is assigned at link 0, the platform. For the endeffector, instead of attaching coordinate system ( $x_4$ ,  $y_4$ ,  $z_4$ ) to link 4, the system (n, o, a) is defined with  $x_4$  replaced by the unit normal vector n,  $y_4$  by the unit orientation vector o, and  $z_4$  by the unit approach vector a. the system (n, o, a) specifies the orientation of the wheel. The DH parameters,  $\theta_i$ ,  $d_i$ ,  $a_i$ , and  $\alpha_i$ , are defined for each joint pair according to the criteria as given bellow.

DH parameters	Notations	Description	
Joint angle	$\theta_{i}$	rotating angle between the $x_{i-1}$ and $x_i$ axes about $z_{i-1}$ axis.	
Link offset	$d_i$	translating distance from $x_{i-1}$ and $x_i$ along $z_{i-1}$ .	
Link length	a <sub>i</sub>	translating distance from $z_{i-1}$ and $z_i$ along the $x_i$ .	
Twisted angle	$\alpha_{i}$	rotating angle between $z_{i-1}$ and $z_i$ axis about $x_i$ axis.	

Table A.1. DH explanation.



Figure A.1. Two adjacent links [65].

So the homogeneous transformation matrix  $A_i^{i-1}$  that represents the position and orientation of the coordinate system i relative to i-1 is:

$$A_i^{i-1} = \operatorname{Rot}(z_{i-1}, \theta_i) . \operatorname{Tran}(0, 0, d_i) . \operatorname{Tran}(a_i, 0, 0) . \operatorname{Rot}(x_i, \alpha_i)$$

$$A_{i}^{i-1} = \begin{bmatrix} \cos\theta_{i} & -\sin\theta_{i} & 0 & 0\\ \sin\theta_{i} & \cos\theta_{i} & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_{i}\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & d_{i}\\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0\\ 0 & \cos\alpha_{i} & -\sin\alpha_{i} & 0\\ 0 & \sin\alpha_{i} & \cos\alpha_{i} & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_{i} & -\sin\theta_{i}\cos\alpha_{i} & \sin\theta_{i}\sin\alpha_{i} & a_{i}\cdot\cos\theta_{i} \\ \sin\theta_{i} & \cos\theta_{i}\cos\alpha_{i} & -\cos\theta_{i}\sin\alpha_{i} & a_{i}\cdot\sin\theta_{i} \\ 0 & \sin\alpha_{i} & \cos\alpha_{i} & d_{i} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(A.1)

194

In a robot manipulator, there are two types of joints; revolute and prismatic. For revolute joint,  $\theta_i$  varies by allowing for rotation between two links about an axis and is called the joint angle where the link offset  $d_i$  is constant; and for a prismatic joint, the link offset  $d_i$  varies by allowing for translation (sliding) motion along an axis and is called the joint displacement where the joint angle  $\theta_i$  is constant and the link length also  $a_i = 0$ . The generalized coordinates,  $q_i$ , represent the formulations of these two types as follows:

$$q_{i} = \begin{cases} \theta_{i} & \text{for a revolute joint} \\ d_{i} & \text{for a prismatic joint} \end{cases}$$
(A.2)



Figure A.2. Types of joints

# Appendix B: Inverse kinematics

Until now, we know the target in which the manipulator reached by solving the forward kinematic equations for the Rover, and we have completed the system transform graph and also defined the homogeneous transformation between frames of the platform universal, ground universal, and contact point. However, we are now concerned to know the joints angles in order to make the required joints' moves  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$  and  $\theta_4$  in term of the given numerical values of the orientation and position.

Equating the generalized matrix  $B_4^0$  to the forward kinematics  $A_4^0$ , we obtain matrix equation:

$$\mathbf{B}_4^0 = \mathbf{A}_4^0$$

$$\begin{bmatrix} n_{x} & o_{x} & a_{x} & p_{x} \\ n_{y} & o_{y} & a_{y} & p_{y} \\ n_{z} & o_{z} & a_{z} & p_{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -C_{1}C_{2}C_{34} + S_{1}S_{34} & C_{1}C_{2}S_{34} + S_{1}C_{34} & C_{1}S_{2} & -C_{1}C_{2}(a_{4}C_{34} + a_{3}C_{3}) + S_{1}(a_{4}S_{34} + a_{3}S_{3}) \\ -S_{1}C_{2}C_{34} - C_{1}S_{34} & S_{1}C_{2}S_{34} - C_{1}C_{34} & S_{1}S_{2} & -S_{1}C_{2}(a_{4}C_{34} + a_{3}C_{3}) - C_{1}(a_{4}S_{34} + a_{3}S_{3}) \\ S_{2}C_{34} & -S_{2}S_{34} & C_{2} & S_{2}(a_{4}C_{34} + a_{3}C_{3}) + d_{1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(B.1)

Where, the matrix equality implies 12 element-by-element equality forming 12 non-trivial equations

$$\mathbf{n}_{x} = -\mathbf{C}_{1}\mathbf{C}_{2}\mathbf{C}_{34} + \mathbf{S}_{1}\mathbf{S}_{34} \tag{B.2}$$

$n_{y} = -S_{1}C_{2}C_{34} - C_{1}S_{34}$ $n_{z} = S_{2}C_{34}$	(B.3) (B.4)
$0_x = C_1 C_2 S_{34} + S_1 C_{34}$	(B.5)
$o_{\rm v} = S_1 C_2 S_{34} - C_1 C_{34}$	(B.6)
$o_z = -S_2 S_{34}$	(B.7)
$a_x = C_1 S_2$	(B.8)
$a_v = S_1 S_2$	(B.9)
$a_z = C_2$	(B.10)
$p_x = -C_1C_2(a_4C_{34}+a_3C_3)+S_1(a_4S_{34}+a_3S_3)$	(B.11)
$p_{y} = -S_{1}C_{2}(a_{4}C_{34}+a_{3}C_{3})-C_{1}(a_{4}S_{34}+a_{3}S_{3})$	(B.12)

$$p_z = S_2(a_4C_{34} + a_3C_3) + d_1$$
(B.12)  
(B.13)

The solutions for  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$  and  $\theta_4$  through using the arc cosine or sine function are inaccurate, since the sign of angle will not be taken into consideration and the division by  $\sin\theta_i$  will make it undefined whenever  $\theta_i$  is close to 0 or  $\pm 180$  [66]. Therefore, the arc tangent function will mostly be taken into our computation providing two arguments,  $\{x, y\}$ , within the interval of  $-\pi \le \theta_i < \pi$  in order to check the sign of y and x and examine when either x or y is zero. x represent the adjacent side, and y represent the opposite side. This procedure will provide the correct and accurate results.

However, one of the most difficult forms of trigonometric equations is presented here that solved by squaring and adding [67]. Moreover, the arc

197
cosine function will, in this case, be taken into computation providing two arguments,  $\{x, y\}$ , within the interval of  $-\pi \le \theta_i < \pi$ 



Figure B.1. The atan2(y, x) function

The angle variables are evaluated in a sequential manner; each variable is isolated by pre-multiplying the matrix equation successively by the inverse transforms starting at base frame  $(A_1^0)^{-1}$  and working forward

$$B_4^0 = A_4^0$$
(B.14)

$$\left(A_{1}^{0}\right)^{-1}B_{4}^{0} = A_{4}^{1}$$
(B.15)

$$\left(\mathbf{A}_{2}^{1}\right)^{-1}\left(\mathbf{A}_{1}^{0}\right)^{-1}\mathbf{B}_{4}^{0} = \mathbf{A}_{4}^{2}$$
(B.16)

$$\left(\mathbf{A}_{3}^{2}\right)^{-1}\left(\mathbf{A}_{2}^{1}\right)^{-1}\left(\mathbf{A}_{1}^{0}\right)^{-1}\mathbf{B}_{4}^{0}=\mathbf{A}_{4}^{3}$$
(B.17)

The matrices elements on the left hand sides of the above matrix equations are functions of the (i-1)<sup>th</sup> joint variables and the numerical values

transform  $B_4^0$ . The matrix elements on the right hand sides are products of **A** matrices, and these are either zero, constant, or functions of the 1<sup>th</sup> to 4<sup>th</sup> joint variables. The products of **A** matrices, defined on the right hand side, are evaluated starting at link four  $A_4^3$  and working back towards the base frame as follows:

$$\mathbf{A}_{4}^{3} = \begin{bmatrix} \mathbf{C}_{4} & -\mathbf{S}_{4} & 0 & \mathbf{a}_{4}\mathbf{C}_{4} \\ \mathbf{S}_{4} & \mathbf{C}_{4} & 0 & \mathbf{a}_{4}\mathbf{S}_{4} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(B.18)

$$\mathbf{A}_{4}^{2} = \begin{bmatrix} -\mathbf{C}_{34} & \mathbf{S}_{34} & \mathbf{0} - \mathbf{a}_{4}\mathbf{C}_{34} - \mathbf{a}_{3}\mathbf{C}_{3} \\ -\mathbf{S}_{34} & -\mathbf{C}_{34} & \mathbf{0} - \mathbf{a}_{4}\mathbf{S}_{34} - \mathbf{a}_{3}\mathbf{S}_{3} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$
(B.19)

$$A_{4}^{1} = \begin{bmatrix} -C_{2}C_{34} & C_{2}S_{34} & S_{2} & -C_{2}(a_{4}C_{34} + a_{3}C_{3}) \\ -S_{2}C_{34} & S_{2}S_{34} & -C_{2} & -S_{2}(a_{4}C_{34} + a_{3}C_{3}) \\ -S_{34} & -C_{34} & 0 & -a_{4}S_{34} - a_{3}S_{3} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(B.20)

1. Differential joint angle;  $\theta_1$ :

If we pre-multiply equation B.14 by  $(A_1^0)^{-1}$ , we obtain equation B.15

$$\begin{pmatrix} A_{1}^{0} \end{pmatrix}^{-1} B_{4}^{0} = A_{4}^{1}$$

$$\begin{bmatrix} C_{1} & S_{1} & 0 & 0 \\ 0 & 0 & -1 & d_{1} \\ -S_{1} & C_{1} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} n_{x} & 0_{x} & a_{x} & p_{x} \\ n_{y} & 0_{y} & a_{y} & p_{y} \\ n_{z} & 0_{z} & a_{z} & p_{z} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -C_{2}C_{34} & C_{2}S_{34} & S_{2} & -C_{2}(a_{4}C_{34} + a_{3}C_{3}) \\ -S_{2}C_{34} & S_{2}S_{34} & -C_{2} & -S_{2}(a_{4}C_{34} + a_{3}C_{3}) \\ -S_{34} & -C_{34} & 0 & -a_{4}S_{34} - a_{3}S_{3} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(B.21)$$

The left hand side of above equation is a function of the given numerical values of generalized transform  $B_4^0$  pre-multiplied by a function of  $\theta_1$  inside the inverse transform of  $A_1^0$ . The right hand side is a function of  $\theta_2$ ,  $\theta_3$ , and  $\theta_4$ . After rearranging the above equation, we obtain

$$\begin{bmatrix} C_{1}n_{x}+S_{1}n_{y} & C_{1}o_{x}+S_{1}o_{y} & C_{1}a_{x}+S_{1}a_{y} & C_{1}p_{x}+S_{1}p_{y} \\ -n_{z} & -o_{z} & -a_{z} & -p_{z}+d_{1} \\ -S_{1}n_{x}+C_{1}n_{y} & -S_{1}o_{x}+C_{1}o_{y} & -S_{1}a_{x}+C_{1}a_{y} & -S_{1}p_{x}+C_{1}p_{y} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -C_{2}C_{34} & C_{2}S_{34} & S_{2} & -C_{2}(a_{4}C_{34}+a_{3}C_{3}) \\ -S_{2}C_{34} & S_{2}S_{34} & -C_{2} & -S_{2}(a_{4}C_{34}+a_{3}C_{3}) \\ -S_{34} & -C_{34} & 0 & -a_{4}S_{34}-a_{3}S_{3} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(B.22)

The third row, third column element on the right hand side of equation (B.22) is zero. Equating this to the element on the left hand at the same location we obtain

$$-S_1a_x + C_1a_y = 0 \tag{B.23}$$

201

This form denotes for a point symmetric redundancy, because it generates two solutions that are symmetric about the origin as shown in Figure B.2.



Figure B.2. Point symmetric redundancy

The first solution can obtained by Adding  $S_1a_x$  to both sides and dividing by  $C_1a_x$ , we get

$$\tan\theta_1 = \frac{\sin\theta_1}{\cos\theta_1} = \frac{a_y}{a_x} \tag{B.24}$$

$$\theta_1 = tg^{-1} \left( \frac{a_y}{a_x} \right) \tag{B.25}$$

The angle  $\theta_1$  is obtained from the computer in term of atan2 function as

$$\theta_1 = \operatorname{atan} 2(a_y, a_x) \tag{B.26}$$

the second solution for  $\theta_1$  can be obtained by adding to both sides  $-C_1a_y$ , and dividing by  $C_1a_x$ , and canceling  $-a_x$  on the left hand side and  $C_1$  on the right side hand.

$$\theta_{1} = \operatorname{atan} 2\left(-a_{y}, -a_{x}\right) \tag{B.27}$$

After determining the value of  $\theta_1$ , all elements inside the left hand side are totally known. We check the right hand side for other functions of single variables,  $\theta_1$  can be found.

## 2. Conjunctional joint angle; $\theta_2$ :

Examining the right hand side for further unknown individual joint coordinate, we can equate the 1,4 and 2,4 elements from left and right hand sides of equation B.22.

$$C_{1}p_{x} + S_{1}p_{y} = -C_{2}(a_{4}C_{34} + a_{3}C_{3})$$
(B.28)

$$-p_{z}+d_{1}=-S_{2}(a_{4}C_{34}+a_{3}C_{3})$$
(B.29)

then,

$$\theta_{2} = \operatorname{atan} 2\left(-\left(-p_{z}+d_{1}\right), -\left(C_{1}p_{x}+S_{1}p_{y}\right)\right)$$
(B.30)

The angle  $\theta_2$  here is always unique and there is no degeneracy as in the case of the previous angle  $\theta_1$ .

We check the right hand side for further functions of single variables. Finding none, we need for new pre-multiplication technique for obtaining new information.

3. Wheel frame;  $\theta_4$ :

As mentioned before, the wheel is equipped for locomotive and manipulative mechanism, meanwhile the inverse kinematics deals only with manipulations apart from wheel rotation. If we pre-multiply equation B.15 by  $(A_2^1)^{-1} \cdot (A_1^0)^{-1} \cdot B_4^0 = A_4^2$  $\begin{bmatrix} C_1C_2 & S_1C_2 & -S_2 & S_2d_1 \\ -S_1 & C_1 & 0 & 0 \\ C_1S_2 & S_1S_2 & C_2 & -C_2d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & 0_x & a_x & p_x \\ n_y & 0_y & a_y & p_y \\ n_z & 0_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -C_{34} & S_{34} & 0 & -a_4C_{34} - a_3C_3 \\ -S_{34} & -C_{34} & 0 & -a_4S_{34} - a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_1C_2n_x + S_1C_2n_y - S_2n_z & C_1C_20_x + S_1C_2n_y - S_2n_z & C_1C_2n_z + S_1C_2n_y - S_2n_z + S_1C_2n_y - S_2n_z + C_1n_y & -S_1n_z + C_1n_y &$ 

$$= \begin{bmatrix} -C_{34} & S_{34} & 0 & -a_4C_{34} - a_3C_3 \\ -S_{34} & -C_{34} & 0 & -a_4S_{34} - a_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(B.31)

Examining the right hand side, we can equate the 1,4 and 2,4 elements from left and right hand sides

$$C_1 C_2 p_x + S_1 C_2 p_y - S_2 p_z + S_2 d_1 = -a_4 C_{34} - a_3 C_3$$
(B.32)

$$-S_{1}p_{x} + C_{1}p_{y} = -a_{4}S_{34} - a_{3}S_{3}$$
(B.33)

The angle  $\theta_{\scriptscriptstyle 4}$  can be solved by squaring and adding techniques. Let,

$$K_1 = -a_4 C_{34} - a_3 C_3 \tag{B.34}$$

$$K_2 = -a_4 S_{34} - a_3 S_3 \tag{B.35}$$

These can be squared and added to give us one trigonometric equation as

$$(K_1)^2 + (K_2)^2 = (a_4)^2 + (a_3)^2 + 2a_3a_4(C_3C_{34} + S_3S_{34})$$
  
=  $(a_4)^2 + (a_3)^2 + 2a_3a_4C_4$  (B.36)

204

The result implies that there are two solutions for angle  $\theta_4$  which are symmetric about zero; (+) sign assigns for RF leg and LR leg; and (-) sign assigns for RR leg and LF leg.

$$\theta_{4} = \pm \operatorname{acos}\left(\left(\left(K_{1}\right)^{2} + \left(K_{2}\right)^{2} - \left(a_{4}\right)^{2} + \left(a_{3}\right)^{2}\right)/2a_{3}a_{4}\right)$$
(B.37)

Substituting  $K_1$  and  $K_2$ , we get

$$\theta_{4} = \pm \operatorname{acos}\left(\left(\left(C_{1}C_{2}p_{x} + S_{1}C_{2}p_{y} - S_{2}p_{z} + S_{2}d_{1}\right)^{2} + \left(-S_{1}p_{x} + C_{1}p_{y}\right)^{2} - \left(a_{4}\right)^{2} + \left(a_{3}\right)^{2}\right) / 2a_{3}a_{4}\right)$$
(B.38)

#### 4. Disjunctional joint angle; $\theta_3$ :

The angle  $\theta_3$  can be solved by a recursion technique in inverse kinematics problem; rearranging the equations B.34 and B.35, respectively, in the forms

$$K_{1} = -C_{3}(a_{4}C_{4} + a_{3}) + S_{3}(a_{4}S_{4})$$
(B.39)

$$K_{2} = -S_{3}(a_{4}C_{4} + a_{3}) - C_{3}(a_{4}S_{4})$$
(B.40)

And then equating  $K_3$  and  $K_4$  respectively to

$$K_3 = a_4 C_4 + a_3 \tag{B.41}$$

$$\mathbf{K}_4 = \mathbf{a}_4 \mathbf{S}_4 \tag{B.42}$$

We obtain,

$$K_1 = -C_3 K_3 + S_3 K_4 \tag{B.43}$$

$$K_2 = -S_3 K_3 - C_3 K_4 \tag{B.44}$$

Applying mutual multiplications of equations B.43 and B.44, we obtain

$$-S_{3}K_{1}K_{3} - C_{3}K_{1}K_{4} = -C_{3}K_{2}K_{3} + S_{3}K_{2}K_{4}$$
(B.45)

Rearranging the above equation to

$$S_{3}(-K_{1}K_{3}-K_{2}K_{4}) = C_{3}(K_{1}K_{4}-K_{2}K_{3})$$
(B.46)

Finally,

$$\theta_{3} = \operatorname{atan} 2 \left( K_{1} K_{4} - K_{2} K_{3}, -K_{1} K_{3} - K_{2} K_{4} \right)$$
(B.47)

Substituting  $K_1$ ,  $K_2$ ,  $K_3$ , and  $K_4$  we obtain

$$\theta_{3} = \operatorname{atan} 2((C_{1}C_{2}p_{x} + S_{1}C_{2}p_{y} - S_{2}p_{z} + S_{2}d_{1})(a_{4}S_{4}) - (-S_{1}p_{x} + C_{1}p_{y})(a_{4}C_{4} + a_{3}), - (C_{1}C_{2}p_{x} + S_{1}C_{2}p_{y} - S_{2}p_{z} + S_{2}d_{1})(a_{4}C_{4} + a_{3}) - (-S_{1}p_{x} + C_{1}p_{y})(a_{4}S_{4}))$$
(B.48)

206

# Appendix C: Kinematic and dynamic parameters

## **Rotational matrix**

Rotation matrix transformation from universal frame to base frame is given by

$$\mathbf{R}_{0}^{\mathrm{U}} = \begin{bmatrix} \mathbf{c}\phi\mathbf{c}\theta & -\mathbf{c}\phi\mathbf{s}\theta\mathbf{c}\psi + \mathbf{s}\phi\mathbf{s}\psi & \mathbf{c}\phi\mathbf{s}\theta\mathbf{s}\psi + \mathbf{s}\phi\mathbf{c}\psi \\ \mathbf{s}\theta & \mathbf{c}\mathbf{o}\mathbf{s}\theta\mathbf{c}\mathbf{o}\mathbf{s}\psi & -\mathbf{c}\theta\mathbf{s}\psi \\ -\mathbf{s}\phi\mathbf{c}\theta & \mathbf{s}\phi\mathbf{s}\theta\mathbf{c}\psi + \mathbf{c}\phi\mathbf{s}\psi & -\mathbf{s}\phi\sin\theta\mathbf{s}\psi + \mathbf{c}\phi\mathbf{c}\psi \end{bmatrix}$$
(C.1)

Rotation matrices of joints are given by

$$R_{1}^{0} = \begin{bmatrix} C_{1} & 0 & -S_{1} \\ S_{1} & 0 & C_{1} \\ 0 & -1 & 0 \end{bmatrix}$$
(C.2)  
$$R_{2}^{1} = \begin{bmatrix} C_{2} & 0 & S_{2} \\ S_{2} & 0 & -C_{2} \\ 0 & 1 & 0 \end{bmatrix}$$
(C.3)  
$$R_{3}^{2} = \begin{bmatrix} -C_{3} & S_{3} & 0 \\ -S_{3} & -C_{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
(C.4)  
$$R_{4}^{3} = \begin{bmatrix} C_{4} & -S_{4} & 0 \\ S_{4} & C_{4} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
(C.5)

## **Position vectors**

The position vector from the origin of frame i to i+1 with respect to frame i+1 is:

$$\mathbf{r}_{i+1}^{i+1} = \mathbf{R}_{i}^{i+1} \mathbf{r}_{i+1}^{i} = \begin{bmatrix} \mathbf{a}_{i+1} & \mathbf{d}_{i+1} \sin \alpha_{i+1} & \mathbf{d}_{i+1} \cos \alpha_{i+1} \end{bmatrix}^{\mathrm{T}}$$
(C.6)

Applying on the above relation starting from base to end-effector, we obtain

$$\mathbf{r}_0^0 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{\mathrm{T}} \tag{C.7}$$

$$\mathbf{r}_1^1 = \begin{bmatrix} \mathbf{0} & -\mathbf{d}_1 & \mathbf{0} \end{bmatrix}^{\mathrm{T}} \tag{C.8}$$

$$\mathbf{r}_2^2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}^{\mathrm{T}} \tag{C.9}$$

$$\mathbf{r}_3^3 = \begin{bmatrix} \mathbf{a}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix}^{\mathrm{T}} \tag{C.10}$$

$$\mathbf{r}_4^4 = \begin{bmatrix} \mathbf{a}_4 & \mathbf{0} & \mathbf{0} \end{bmatrix}^{\mathrm{T}} \tag{C.11}$$

Position vector of center of mass of link i with respect to frame  $O_i \mbox{ is }$ 

$$\mathbf{r}_{c,1}^{1} = \begin{bmatrix} 0 & 0.5 \, \mathbf{d}_{1} & 0 \end{bmatrix}^{\mathrm{T}} \tag{C.12}$$

$$\mathbf{r}_{c,2}^{2} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$$
(C.13)

$$\mathbf{r}_{c,3}^{3} = \begin{bmatrix} -0.5a_{3} & 0 & 0 \end{bmatrix}^{\mathrm{T}}$$
 (C.14)

$$\mathbf{r}_{c,4}^{4} = \begin{bmatrix} -a_{4} & 0 & 0 \end{bmatrix}^{\mathrm{T}}$$
(C.15)

#### Masses

The mass of the robot creates weight and inertia; weight is a force that points down vertically in the universal coordinate system. Inertia on the other hand creates resistance to acceleration caused by force. The distribution of masses among the four legs and platform plays a major role in Specifying the location of center of mass for the robot in the case of the rotations in legs.

#### **Rover's Center of mass**

Each part of the rover is considered as a rigid body, while the rover mass is represented in single concentrated point, called Center of Mass. In other meaning, the weight of the entire robot mass is focused only at the center of mass.

$$\begin{aligned} \mathbf{r}_{cm}^{U} &= \frac{\sum_{i=0}^{4} \mathbf{r}_{i}^{U} \mathbf{m}_{i}}{\sum_{i=0}^{4} \mathbf{m}_{i}} \\ &= \frac{\left( \mathbf{r}_{c,0R}^{U} \mathbf{m}_{0} + \mathbf{r}_{c,1R}^{U} \mathbf{m}_{1} + \mathbf{r}_{c,1R}^{U} \mathbf{m}_{1} + \mathbf{r}_{c,2R}^{U} \mathbf{m}_{2} + \mathbf{r}_{c,2L}^{U} \mathbf{m}_{2} + \mathbf{r}_{c,3RF}^{U} \mathbf{m}_{3} + \mathbf{r}_{c,3RR}^{U} \mathbf{m}_{3} + \frac{\mathbf{r}_{c,3RR}^{U} \mathbf{m}_{3} + \mathbf{r}_{c,3RR}^{U} \mathbf{m}_{3} + \mathbf{r}_{c,4RR}^{U} \mathbf{m}_{4} + \mathbf{r}_{c,4RR}^{U} \mathbf{m}_{4} + \mathbf{r}_{c,4LF}^{U} \mathbf{m}_{4} + \mathbf{r}_{c,4LR}^{U} \mathbf{m}_{4} + \mathbf{r}_{c,4LR}^{U} \mathbf{m}_{4} + \mathbf{r}_{c,4LR}^{U} \mathbf{m}_{4} + \mathbf{r}_{c,4RR}^{U} \mathbf{m}_{4} +$$

where,

 $\mathbf{r}_{i}^{U}$  position vector of frame  $\mathbf{O}_{i}$  with respect to universal frame  $\mathbf{O}_{U}$ 

 $m_i$  mass of the link i starting from platform link and ending at end-effector link.

#### Inertia

Inertia creates a resistance against the change in velocity or acceleration caused by external force. On other words, it is the propensity of the link at rest to stay at rest and propensity of the link in motion to stay in motion. Therefore, the link with high inertia will be in need for a sufficient amount of torque to accelerate or decelerate the object itself. Inertia is considered as mass in the case of linear motion and as moment of inertia in the case of rotational motion. The mass moment of inertia is directly proportional to the mass distribution and the shape of the link.

Inertia matrix for each rigid link is an identical matrix, and it includes moments of inertia and products of inertia conforming six unique elements. The moments of inertia are three diagonal elements, i.e. I<sub>xx</sub>, I<sub>yy</sub>, I<sub>zz</sub>. The products of inertia are off-diagonal elements, i.e. I<sub>xy</sub>, I<sub>yz</sub>.

$$\mathbf{I} = \begin{bmatrix} \mathbf{I}_{xx} & \mathbf{I}_{xy} & \mathbf{I}_{xz} \\ \mathbf{I}_{xy} & \mathbf{I}_{yy} & \mathbf{I}_{yz} \\ \mathbf{I}_{xz} & \mathbf{I}_{yz} & \mathbf{I}_{zz} \end{bmatrix}$$
(C.17)

The symmetry of link is used to recognize the principal axes. The offdiagonal elements are equal zero due to symmetry. The principal mass moments of inertia can be found without solving the corresponding eigenvalue problem. The moments of inertia can be transformed between coordinate systems.

$$\mathbf{I} = \begin{bmatrix} \mathbf{I}_{x} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{z} \end{bmatrix}$$
(C.18)

The SI unit for mass moment of inertia is kg m<sup>2</sup>

For each leg, the links used are one rectangular prism, two slender rods, and one thin disk as shown in Figure C.1



Figure C.1. Rover's DH and dynamic parameters



Figure C.2. Link's DH and Dynamic parameters.

Link 0 is a rectangular prism, and its inertia matrix can be obtain as

$$I_{0} = \frac{m_{0}}{3} \begin{bmatrix} a^{2} + d_{1}^{2} & 0 & 0\\ 0 & b^{2} + d_{1}^{2} & 0\\ 0 & 0 & a^{2} + b^{2} \end{bmatrix}$$
(B.19)

Link 1 is a slender rod, and its inertia matrix can be obtain as

$$I_{1} = \frac{m_{1}(d_{1})^{2}}{12} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
(C.20)

Inertia matrix for link 2

$$\mathbf{I}_{2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
(C.21)

Link 3 is a slender rod, and its inertia matrix can be obtain as

$$I_{3} = \frac{m_{3}(a_{3})^{2}}{12} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
(C.22)

Link 4 is a thin disk, and its inertia matrix can be obtain as

$$I_{4} = (I_{4})_{ex} - (I_{4})_{in}$$

$$= (m_{4ex}a_{4ex}^{2} - m_{4in}a_{4in}^{2}) \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$$
(C.23)

Now, the table of dynamic parameters can be filled up as follows

Link	m	r <sub>x</sub>	ry	rz	I <sub>x</sub>	Iy	Iz
0	m <sub>0</sub>	0	0	0	$\frac{m_0\left(a^2+d_1^2\right)}{3}$	$\frac{m_0\left(b^2+d_1^2\right)}{3}$	$\frac{m_0(a^2+b^2)}{3}$
1	$m_1$	0	$\frac{d_1}{2}$	0	$\frac{m_1(d_1)^2}{12}$	0	$\frac{m_1(d_1)^2}{12}$
2	m <sub>2</sub>	0	0	0	0	0	0
3	m3	$-\frac{a_3}{2}$	0	0	0	$\frac{\mathrm{m}_{3}\mathrm{(a_{3})}^{2}}{12}$	$\frac{\mathrm{m}_{3}\mathrm{(a_{3})}^{2}}{12}$
4	m4	-a4	0	0	$\frac{m_{4ex}a_{4ex}^2 - m_{4in}a_{4in}^2}{4}$	$\frac{m_{4ex}a_{4ex}^2 - m_{4in}a_{4in}^2}{4}$	$\frac{m_{4ex}a_{4ex}^2 - m_{4in}a_{4in}^2}{2}$

Table C.1. Dynamic parameters table.

#### Appendix D: Newton-Euler Recursive Formulation

The dynamic equations of the links are expressed here using the relationships of moving coordinate systems. The numerical algorithm for Newton-Euler Recursive method can be broken into two forward and backward recursions.

#### - The forward recursion

For rotational link i+1

$$\omega_{i+1}^{i} = \mathbf{Z}_{i} \dot{\mathbf{q}}_{i+1} \tag{D.1}$$

$$\dot{\omega}_{i+1}^{i} = \mathbf{Z}_{i} \ddot{\mathbf{q}}_{i+1} \tag{D.2}$$

$$\boldsymbol{\omega}_{i+1}^{0} = \boldsymbol{\omega}_{i+1}^{0} \cdot \boldsymbol{\omega}_{i+1} \tag{D.3}$$

$$\dot{\omega}_{i+1}^{0} = \dot{\omega}_{i}^{0} + Z_{i}\ddot{q}_{i+1} + \omega_{i}^{0} \times (Z_{i}\dot{q}_{i+1})$$
(D.3)
$$\dot{\omega}_{i+1}^{0} = \dot{\omega}_{i}^{0} + Z_{i}\ddot{q}_{i+1} + \omega_{i}^{0} \times (Z_{i}\dot{q}_{i+1})$$
(D.4)

$$\mathbf{v}_{i+1}^{0} = \boldsymbol{\omega}_{i+1}^{0} \times \mathbf{r}_{i+1}^{i} + \mathbf{v}_{i}^{0}$$
(D.5)

$$\mathbf{a}_{i+1}^{o} = \dot{\omega}_{i+1}^{o} \times \mathbf{r}_{i+1}^{i} + \omega_{i+1}^{o} \times \left(\omega_{i+1}^{o} \times \mathbf{r}_{i+1}^{i}\right) + \mathbf{v}_{i}^{o}$$
(D.6)

The velocity and acceleration of center of mass of link i are computed respectively as follows:

$$\mathbf{V}_{c,i}^{0} = \boldsymbol{\omega}_{i}^{0} \times \mathbf{r}_{c,i}^{0} + \mathbf{V}_{i}^{0}$$
(D.7)

$$\mathbf{a}_{c,i}^{0} = \dot{\mathbf{\omega}}_{i}^{0} \times \mathbf{r}_{c,i}^{i} + \mathbf{\omega}_{i}^{0} \times \left(\mathbf{\omega}_{i}^{0} \times \mathbf{r}_{c,i}^{0}\right) + \dot{\mathbf{v}}_{i}^{0} \tag{D.8}$$

Once the velocities and accelerations of the center of mass of links are computed, the inertia forces and moments can be computed for each mass link. Assuming the viscous damping friction is negligible, the total external force is given by the Newton's second law, and whilst the moment is given by Euler's equation. Newton-Euler's methods first described with regard to the fixed base coordinate system [55]

$$\begin{aligned} \mathbf{f}_{i}^{0} &= \mathbf{m}_{i} \dot{\mathbf{v}}_{c,i}^{0} \qquad (D.9) \\ \boldsymbol{\tau}_{i}^{0} &= \mathbf{I}_{i} \dot{\boldsymbol{\omega}}_{i}^{0} + \boldsymbol{\omega}_{i}^{0} \times \left( \mathbf{I}_{i} \boldsymbol{\omega}_{i}^{0} \right) \qquad (D.10) \end{aligned}$$

#### - The backward recursion

This approach transforms the generalized forces back from the endeffector  $O_{n+1}$  to the base frame  $O_0$ . The total force and moment exerted on center of mass of link i are equal the forces and moments, respectively, exerted on link i by link i-1 and i+1:

$$\mathbf{f}_{i}^{0} = \mathbf{F}_{i}^{0} - \mathbf{F}_{i+1}^{0} \tag{D.11}$$

$$\boldsymbol{\tau}_{i}^{0} = T_{i}^{0} - T_{i+1}^{0} - \mathbf{r}_{i+1}^{i} \times F_{i+1}^{0} - (\mathbf{r}_{i-1}^{i} + \mathbf{r}_{c,i}^{0}) \times \mathbf{f}_{i}^{0}$$
(D.12)

Arranging the above equations in recursive form, we obtain

$$F_i^0 = F_{i+1}^0 + f_i^0$$
(D.13)

$$T_{i}^{0} = T_{i+1}^{0} + r_{i-1}^{i} \times F_{i+1}^{0} + (r_{i-1}^{i} + r_{c,i}^{0}) \times f_{i}^{0} + \tau_{i}^{0}$$
(D.14)

#### Computational approach

Multiply  $R_0^{_{i+1}}$  with  $\omega_{_{i+1}}^{_{i+1}}$  [65], we obtain

$$\omega_{i+1}^{0} = R_{i+1}^{0} \omega_{i+1}^{i+1}$$
(D.15)

Multiplying the above equation by  $\mathbf{R}_0^{i+1}$ , we obtain

$$\omega_{i+1}^{i+1} = R_0^{i+1} \omega_{i+1}^0 \tag{D.16}$$

The lower sub script indicates for the reference coordinate frame. In such a way,  $\omega_{i+1}^{i+1}$  should be read as an angular velocity vector from frame  $O_i$  to frame  $O_{i+1}$  expressed in its own coordinate frame  $O_{i+1}$ .

The rotation matrix of homogeneous transformation of frame  $O_{i+1}$  with respect  $O_i$  is

$$\mathbf{R}_{i+1}^{i} = \begin{bmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1}\cos\alpha_{i+1} & \sin\theta_{i+1}\sin\alpha_{i+1} \\ \sin\theta_{i+1} & \cos\theta_{i+1}\cos\alpha_{i+1} & -\cos\theta_{i+1}\sin\alpha_{i+1} \\ 0 & \sin\alpha_{i+1} & \cos\alpha_{i+1} \end{bmatrix}$$
(D.17)

The rotation matrix of homogeneous transformation of frame  $O_i$  with respect  $O_{i+!}$  is equal to the transpose or inverse of  $R^i_{i+l}$ 

$$\mathbf{R}_{i}^{i+1} = \left(\mathbf{R}_{i+1}^{i}\right)^{-1} = \left(\mathbf{R}_{i+1}^{i}\right)^{\mathrm{T}}$$
(D.18)

The position vector from frame  $O_i$  to frame  $O_{i+1}$  expressed in frame  $O_{i+1}$  is:

$$\begin{aligned} \mathbf{r}_{i+1}^{i+1} &= \mathbf{R}_{i}^{i+1} \mathbf{r}_{i+1}^{i} \\ &= \begin{bmatrix} \mathbf{cos} \theta_{i+1} & \mathbf{sin} \theta_{i+1} & \mathbf{0} \\ -\mathbf{sin} \theta_{i+1} \mathbf{cos} \alpha_{i+1} & \mathbf{cos} \theta_{i+1} \mathbf{cos} \alpha_{i+1} & \mathbf{sin} \alpha_{i+1} \\ \mathbf{sin} \theta_{i+1} \mathbf{sin} \alpha_{i+1} & -\mathbf{cos} \theta_{i+1} \mathbf{sin} \alpha_{i+1} & \mathbf{cos} \alpha_{i+1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a}_{i+1} \mathbf{cos} \theta_{i+1} \\ \mathbf{a}_{i+1} \mathbf{sin} \theta_{i+1} \\ \mathbf{d}_{i+1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{a}_{i+1} \\ \mathbf{d}_{i+1} \mathbf{sin} \alpha_{i+1} \\ \mathbf{d}_{i+1} \mathbf{cos} \alpha_{i+1} \end{bmatrix} \end{aligned}$$
(D.19)

Rewriting the recursive relations to be expressed to local reference frame, we get:

### - Forward recursion: $0 \le i \le n-1$

For rotational joint axis i+1:

$$\boldsymbol{\omega}_{i+1}^{i+1} = \mathbf{R}_i^{i+1} \left( \boldsymbol{\omega}_i^i + \dot{\mathbf{q}}_{i+1} \right)$$

$$(D.20)$$

$$\dot{\boldsymbol{\omega}}_{i+1}^{i+1} = \mathbf{R}_i^{i+1} \left( \dot{\boldsymbol{\omega}}_i^i + \ddot{\mathbf{q}}_{i+1} \right)$$

$$(D.21)$$

$$\dot{\omega}_{i+1}^{i+1} = \mathbf{R}_{i}^{i+1} \left( \dot{\omega}_{i}^{i} + \ddot{\mathbf{q}}_{i+1} + \omega_{i}^{i} \times \dot{\mathbf{q}}_{i+1} \right)$$
(D.21)  
$$\mathbf{v}_{i+1}^{i+1} = \omega_{i+1}^{i+1} \times \mathbf{r}_{i+1}^{i+1} + \mathbf{R}_{i+1}^{i+1} \mathbf{v}_{i}^{i}$$
(D.22)

$$\begin{split} \mathbf{v}_{i+1}^{i+1} &= \omega_{i+1}^{i+1} \times \mathbf{r}_{i+1}^{i+1} + \mathbf{R}_{i}^{i+1} \mathbf{v}_{i}^{i} \\ \dot{\mathbf{v}}_{i+1}^{i+1} &= \dot{\omega}_{i+1}^{i+1} \times \mathbf{r}_{i+1}^{i+1} + \omega_{i+1}^{i+1} \times \left( \omega_{i+1}^{i+1} \times \mathbf{r}_{i+1}^{i+1} \right) + \mathbf{R}_{i}^{i+1} \dot{\mathbf{v}}_{i}^{i} \end{split} \tag{D.22}$$

$$\mathbf{V}_{c,i+1}^{i+1} = \boldsymbol{\omega}_{i+1}^{i+1} \times \mathbf{r}_{c,i+1}^{i+1} + \mathbf{V}_{i+1}^{i+1}$$
(D.24)

$$\dot{\mathbf{v}}_{c,i+1}^{i+1} = \dot{\boldsymbol{\omega}}_{i+1}^{i+1} \times \mathbf{r}_{c,i+1}^{i+1} + \boldsymbol{\omega}_{i+1}^{i+1} \times \left(\boldsymbol{\omega}_{i+1}^{i+1} \times \mathbf{r}_{c,i+1}^{i+1}\right) + \dot{\mathbf{v}}_{i+1}^{i+1}$$
(D.25)

$$\mathbf{f}_{i+1}^{i+1} = \mathbf{m}_{i+1} \dot{\mathbf{v}}_{c,i+1}^{i+1} \tag{D.26}$$

$$\tau_{i+1}^{i+1} = \mathbf{I}_{c,i+1}^{i+1} \dot{\omega}_{i+1}^{i+1} + \omega_{i+1}^{i+1} \times \left(\mathbf{I}_{c,i+1}^{i+1} \omega_{i+1}^{i+1}\right)$$
(D.27)

## - Backward recursion: $n \le i \le 0$

After computing the inertial forces and moments for each link, backward computational procedures can be followed by evaluating one a link at a time starting from the end-effector frame and ending at the base frame:

$$F_{i}^{i} = R_{i+1}^{i}F_{i+1}^{i+1} + f_{i}^{i}$$
(D.28)

$$T_{i}^{i} = R_{i+1}^{i} \left( T_{i+1}^{i+1} + \left( R_{i}^{i+1} r_{i}^{i} \right) \times F_{i+1}^{i+1} \right) + \left( r_{i}^{i} + r_{c,i}^{i} \right) \times f_{i}^{i} + \tau_{i}^{i}$$
(D.29)

# Appendix E: Free-Body Diagram for four manipulators



**Figure E.1.** Transform graphs for four legged manipulators starting from universal to end-effectors.

Where, L: Link,



Figure E.2. Forces and moments exerted on link 4RF, 4RR, 4LF, and 4LR.

$$\mathbf{F}_{4\mathrm{RF}}^{4\mathrm{RF}} = \mathbf{R}_{\mathrm{SRF}}^{4\mathrm{RF}} \mathbf{F}_{\mathrm{SRF}}^{\mathrm{SRF}} + \mathbf{f}_{4\mathrm{RF}}^{4\mathrm{RF}} \tag{E.1}$$

$$F_{4RR}^{4RR} = R_{SRR}^{4RR} F_{SRR}^{SRR} + f_{4RR}^{4RR}$$
(E.2)

$$T_{4RF}^{4RF} = R_{SRF}^{4RF} \left( T_{SRF}^{SRF} + \left( R_{4RF}^{SRF} r_{4RF}^{4RF} \right) \times F_{SRF}^{SRF} \right) + \left( r_{4RF}^{4RF} + r_{c,4RF}^{4RF} \right) \times f_{4RF}^{4RF} + \tau_{4RF}^{4RF}$$
(E.3)

$$T_{4RR}^{4RR} = R_{SRR}^{4RR} \left( T_{SRR}^{SRR} + \left( R_{4RR}^{SRR} r_{4RR}^{4RR} \right) \times F_{SRR}^{SRR} \right) + \left( r_{4RR}^{4RR} + r_{c,4RR}^{4RR} \right) \times f_{4RR}^{4RR} + \tau_{4RR}^{4RR}$$
(E.4)

And,

$$\mathbf{F}_{4\mathrm{LF}}^{4\mathrm{LF}} = \mathbf{R}_{\mathrm{SLF}}^{4\mathrm{LF}} \mathbf{F}_{\mathrm{SLF}}^{\mathrm{SLF}} + \mathbf{f}_{4\mathrm{LF}}^{4\mathrm{LF}} \tag{E.5}$$

$$\mathbf{F}_{4LR}^{4LR} = \mathbf{R}_{SLR}^{4LR} \mathbf{F}_{SLR}^{SLR} + \mathbf{f}_{4LR}^{4LR} \tag{E.6}$$

$$T_{4LF}^{4LF} = R_{SLF}^{4LF} \left( T_{SLF}^{SLF} + \left( R_{4LF}^{SLF} r_{4LF}^{4LF} \right) \times F_{SLF}^{SLF} \right) + \left( r_{4LF}^{4LF} + r_{c,4LF}^{4LF} \right) \times f_{4LF}^{4LF} + \tau_{4LF}^{4LF}$$
(E.7)

$$T_{4LR}^{4LR} = R_{SLR}^{4LR} \left( T_{SLR}^{SLR} + \left( R_{4LR}^{SLR} r_{4LR}^{4LR} \right) \times F_{SLR}^{SLR} \right) + \left( r_{4LR}^{4LR} + r_{c,4LR}^{4LR} \right) \times f_{4LR}^{4LR} + \tau_{4LR}^{4LR}$$
(E.8)

## Link 3RF, 3RR, 3LF, and 3LR



Figure E.3. Forces and moments exerted on link 3RF, 3RR, 3LF, and 3LR.

$$F_{3RF}^{3RF} = R_{4RF}^{3RF} F_{4RF}^{4RF} + f_{3RF}^{3RF}$$
(E.9)

$$F_{3RR}^{3RR} = R_{4RR}^{3RR} F_{4RR}^{4R} + f_{3RR}^{3RR}$$
(E.10)

$$T_{3RF}^{3RF} = R_{4RF}^{3RF} \left( T_{4RF}^{4RF} + \left( R_{3RF}^{4RF} r_{3RF}^{3RF} \right) \times F_{4RF}^{4RF} \right) + \left( r_{3RF}^{3RF} + r_{c,3RF}^{3RF} \right) \times f_{3RF}^{3RF} + \tau_{3RF}^{3RF}$$
(E.11)

$$T_{3RR}^{3RR} = R_{4RR}^{3RR} \left( T_{4RR}^{4RR} + \left( R_{3RR}^{4RR} r_{3RR}^{3RR} \right) \times F_{4RR}^{4RR} \right) + \left( r_{3RR}^{3RR} + r_{c,3RR}^{3RR} \right) \times f_{3RR}^{3RR} + \tau_{3RR}^{3RR}$$
(E.12)

and,

$$F_{3LF}^{3LF} = R_{4LF}^{3LF} F_{4LF}^{4LF} + f_{3LF}^{3LF}$$
(E.13)

$$\mathbf{F}_{3LR}^{3LR} = \mathbf{R}_{4LR}^{3LR} \mathbf{F}_{4LR}^{4LR} + \mathbf{f}_{3LR}^{3LR}$$
(E.14)

$$T_{3LF}^{3LF} = R_{4LF}^{3LF} \left( T_{4LF}^{4LF} + \left( R_{3LF}^{4LF} r_{3LF}^{3LF} \right) \times F_{4LF}^{4LF} \right) + \left( r_{3LF}^{3LF} + r_{c,3LF}^{3LF} \right) \times f_{3LF}^{3LF} + \tau_{3LF}^{3LF}$$
(E.15)

$$T_{3LR}^{3LR} = R_{4LR}^{3LR} \left( T_{4LR}^{4LR} + \left( R_{3LR}^{4LR} r_{3LR}^{3LR} \right) \times F_{4LR}^{4LR} \right) + \left( r_{3LR}^{3LR} + r_{c,3LR}^{3LR} \right) \times f_{3LR}^{3LR} + \tau_{3LR}^{3LR}$$
(E.16)

## link 2R and link 2L



Figure E.4. Forces and moments exerted on link 2R, 2L.

$$F_{2R}^{2R} = R_{3RF}^{2R} F_{3RF}^{3RF} + R_{3RR}^{2R} F_{3RR}^{3RR} + f_{2R}^{2R}$$
(E.17)

$$T_{2R}^{2R} = R_{3RF}^{2R} \left( T_{3RF}^{3RF} + \left( R_{2R}^{3RF} r_{2R}^{2R} \right) \times F_{3RF}^{3RF} \right) + R_{3RR}^{2R} \left( T_{3RR}^{3RR} + \left( R_{2R}^{3RR} r_{2R}^{2R} \right) \times F_{3RR}^{3RR} \right) + \left( r_{2R}^{2R} + r_{c,2R}^{2R} \right) \times F_{2R}^{2R} + \tau_{2R}^{2R}$$
(E.18)

and,

$$F_{2L}^{2L} = R_{3LF}^{2L} F_{3LF}^{3LF} + R_{3LR}^{2L} F_{3LR}^{3LR} + f_{2L}^{2L}$$

$$(E.19)$$

$$T_{2L}^{2L} = R_{3LF}^{2R} \left( T_{3LF}^{3LF} + \left( R_{2L}^{3LF} r_{2L}^{2L} \right) \times F_{3LF}^{3LF} \right) + R_{3LR}^{2L} \left( T_{3LR}^{3LR} + \left( R_{2L}^{3LR} r_{2L}^{2L} \right) \times F_{3LR}^{3LR} \right) +$$

$$\frac{2L}{(r_{2L}^{2L} + r_{c,2L}^{2L})} \times f_{2L}^{2L} + \tau_{2L}^{2L}}$$
(E.20)

## link 1R, link 1L



Figure E.5. Forces and moments exerted on link 1R and link 1L.

$$F_{1R}^{1R} = R_{2R}^{1R} F_{2R}^{2R} + f_{1R}^{1R}$$
(E.21)

$$T_{1R}^{1R} = R_{2R}^{1R} \left( T_{2R}^{2R} + \left( R_{1R}^{2R} r_{1R}^{1R} \right) \times F_{2R}^{2R} \right) + \left( r_{1R}^{1R} + r_{c,1R}^{1R} \right) \times f_{1R}^{1R} + \tau_{1R}^{1R}$$
(E.22)

And,

$$F_{1L}^{1L} = R_{2L}^{1L} F_{2L}^{2L} + f_{1L}^{1L}$$
(E.23)

$$T_{1L}^{1L} = R_{2L}^{1L} \left( T_{2L}^{2L} + \left( R_{1L}^{2L} r_{1L}^{1L} \right) \times F_{2L}^{2L} \right) + \left( r_{1L}^{1L} + r_{c,1L}^{1L} \right) \times f_{1L}^{1L} + \tau_{1L}^{1L}$$
(E.24)





Figure E.6. Forces and moments exerted on platform.

$$\begin{split} F_{0R}^{0R} &= R_{1R}^{0R} F_{1R}^{1R} + R_{0L}^{0R} R_{1L}^{0L} F_{1L}^{1L} + f_{0R}^{0R} \\ F_{0R}^{0R} &= R_{1R}^{0R} \left( T_{1R}^{1R} + \left( R_{0R}^{1R} r_{0R}^{0R} \right) \times F_{1R}^{1R} \right) + R_{0L}^{0R} R_{1L}^{0L} \left( T_{1L}^{1L} + \left( R_{0L}^{1L} r_{0L}^{0L} \right) \times F_{1L}^{1L} \right) + \\ \left( r_{0R}^{0R} + r_{c,0R}^{0R} \right) \times f_{0R}^{0R} + \tau_{0R}^{0R} \end{split}$$
(E.25) (E.25)

And finally, forces and moments exerted on platform expressed in universal frame can be given by,

$$F_{0R}^{U} = R_{0R}^{U} F_{0R}^{0R}$$
(E.27)

$$T_{0R}^{U} = R_{0R}^{U} T_{0R}^{0R}$$
(E.28)

#### Appendix F: Universal forces and moments

Newton-Euler Recursive method provides a monitoring system for the sources of dynamic forces and moments exerted on each link of the four manipulators. The decomposition of universal forces and moments can make the point clearer throughout studying the source of each force and moment exerted on the universal frame.

- 1. The source of universal forces: The universal forces are vector summations for:
  - Forces exerted on center of link masses resulted from gravity and inertial linear accelerations; R<sup>U</sup><sub>i</sub>f<sup>i</sup><sub>i</sub>, where i represents center of masses of the links starting from platform link and ending at the wheel links, link by link.
  - External normal forces exerted on the end-effectors;  $A_S^U F_S^S$ , exerted on the touching wheel with surface.

Substituting equation E.25 in equation E.27, we obtain

$$F_{0R}^{U} = R_{0R}^{U} F_{0R}^{0R}$$
  
=  $R_{0R}^{U} \left[ R_{1R}^{0R} F_{1R}^{1R} + R_{0L}^{0R} R_{1L}^{0L} F_{1L}^{1L} + f_{0R}^{0R} \right]$   
=  $R_{1R}^{U} F_{1R}^{1R} + R_{1L}^{U} F_{1L}^{1L} + R_{0R}^{U} f_{0R}^{0R}$  (F.1)

Substituting equations E.21 and E.23 in equation F.1, we obtain

$$F_{0R}^{U} = R_{1R}^{U} \left[ R_{2R}^{1R} F_{2R}^{2R} + f_{1R}^{1R} \right] + R_{1L}^{U} \left[ R_{2L}^{1L} F_{2L}^{2L} + f_{1L}^{1L} \right] + R_{0R}^{U} f_{0R}^{0R}$$
  
=  $R_{2R}^{U} F_{2R}^{2R} + R_{2L}^{U} F_{2L}^{2L} + R_{1R}^{U} f_{1R}^{1R} + R_{1L}^{U} f_{1L}^{1L} + R_{0R}^{U} f_{0R}^{0R}$  (F.2)

Substituting equations E.17 and E.19 in equation F.2, we obtain  $F_{0R}^{U} = R_{2R}^{U} \left[ R_{3RF}^{2R} F_{3RF}^{3RF} + R_{3RR}^{2R} F_{3RR}^{3RR} + f_{2R}^{2R} \right] + R_{2L}^{U} \left[ R_{3LF}^{2L} F_{3LF}^{3LF} + R_{3LR}^{2L} F_{3LR}^{3LR} + f_{2L}^{2L} \right] + R_{1R}^{U} f_{1R}^{1R} + R_{1L}^{U} f_{1L}^{1L} + R_{0R}^{U} f_{0R}^{0R}$   $= R_{3RF}^{U} F_{3RF}^{3RF} + R_{3RR}^{U} F_{3RF}^{3RF} + R_{3LF}^{U} F_{3LF}^{3LF} + R_{2R}^{U} f_{2R}^{3LR} + R_{2L}^{U} f_{2R}^{2L} + R_{2L}^{U} f_{1R}^{2L} + R_{1L}^{U} f_{1L}^{1L} + R_{0R}^{U} f_{0R}^{0R}$ (F.3)

Substituting equations E.9, E.10, E.13, and E.14 in equation F.3, we obtain  

$$F_{0R}^{U} = R_{3RF}^{U} \left[ R_{4RF}^{3RF} F_{4RF}^{4RF} + f_{3RF}^{3RF} \right] + R_{3RR}^{U} \left[ R_{4RR}^{3RR} F_{4RR}^{4RR} + f_{3RR}^{3RR} \right] + R_{3LF}^{U} \left[ R_{4LF}^{3LF} F_{4LF}^{4LF} + f_{3LF}^{3LF} \right] + R_{3LR}^{U} \left[ R_{4LR}^{3LR} F_{4LR}^{4LR} + f_{3LR}^{3LR} \right] + R_{2R}^{U} f_{2R}^{2R} + R_{2L}^{U} f_{2L}^{2L} + R_{1R}^{U} f_{1R}^{1R} + R_{1L}^{U} f_{1L}^{1L} + R_{0R}^{U} f_{0R}^{0R}$$

$$= R_{4RF}^{U} F_{4RF}^{4RF} + R_{4RR}^{U} F_{4RR}^{4RR} + R_{4LF}^{U} F_{4LF}^{4LF} + R_{4LR}^{U} F_{4LR}^{4LR} + R_{3RF}^{U} f_{3RF}^{3RF} + R_{3RR}^{U} f_{3RR}^{3RR} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{2R}^{U} f_{2R}^{2R} + R_{2L}^{U} f_{2L}^{2L} + R_{1R}^{U} f_{1R}^{1L} + R_{0R}^{U} f_{0R}^{3RF} + R_{3RF}^{U} f_{3RF}^{3RF} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{2R}^{U} f_{2R}^{2R} + R_{2L}^{U} f_{2L}^{2L} + R_{1R}^{U} f_{1R}^{1R} + R_{1L}^{U} f_{1L}^{1L} + R_{0R}^{U} f_{0R}^{3RF} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{2L}^{U} f_{2L}^{2L} + R_{1R}^{U} f_{1R}^{1R} + R_{1L}^{U} f_{1L}^{1L} + R_{0R}^{U} f_{0R}^{3RF} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{2L}^{U} f_{2R}^{2L} + R_{2L}^{U} f_{2L}^{2L} + R_{1L}^{U} f_{1L}^{1L} + R_{0R}^{U} f_{0R}^{0R} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{2L}^{U} f_{2L}^{2L} + R_{2L}^{U} f_{2L}^{2L} + R_{2L}^{U} f_{2L}^{2L} + R_{2L}^{U} f_{2L}^{2L} + R_{2L}^{U} f_{2L}^{3L} + R_{2L}^{U} f_{$$

$$\begin{aligned} & \text{Finally, substituting equations E.1, E.2, E.5 and E.6 in equation F.4, we obtain} \\ & \text{F}_{0\text{R}}^{\text{U}} = \text{R}_{4\text{RF}}^{\text{U}} \left[ \text{R}_{\text{SRF}}^{4\text{RF}} \text{F}_{\text{SRF}}^{\text{SRF}} + f_{4\text{RF}}^{4\text{RF}} \right] + \text{R}_{4\text{RR}}^{\text{U}} \left[ \text{R}_{\text{SRR}}^{4\text{RR}} \text{F}_{\text{SRR}}^{\text{SRR}} + f_{4\text{RR}}^{4\text{RR}} \right] + \text{R}_{4\text{LF}}^{\text{U}} \left[ \text{R}_{\text{SLF}}^{4\text{LF}} \text{F}_{\text{SLF}}^{\text{SLF}} + f_{4\text{LF}}^{4\text{LF}} \right] + \\ & \text{R}_{4\text{LR}}^{\text{U}} \left[ \text{R}_{\text{SLR}}^{4\text{LR}} \text{F}_{\text{SLR}}^{\text{SLR}} + f_{4\text{LR}}^{4\text{LR}} \right] + \text{R}_{3\text{RF}}^{0\text{R}} \text{f}_{3\text{RF}}^{3\text{RF}} + \text{R}_{3\text{RR}}^{0\text{U}} \text{f}_{3\text{RF}}^{3\text{LF}} + \text{R}_{3\text{LF}}^{0\text{U}} \text{f}_{3\text{LF}}^{3\text{LF}} + \text{R}_{3\text{LR}}^{0\text{U}} \text{f}_{3\text{LR}}^{3\text{LR}} + \text{R}_{2\text{R}}^{0\text{L}} \text{f}_{3\text{LR}}^{2\text{R}} + \\ & \text{R}_{2\text{L}}^{0\text{U}} \text{f}_{2\text{L}}^{2\text{L}} + \text{R}_{1\text{R}}^{0\text{H}} \text{f}_{1\text{R}}^{1\text{L}} + \text{R}_{0\text{U}}^{0\text{R}} \text{f}_{0\text{R}}^{0\text{R}} \end{aligned} \\ & = \text{R}_{\text{SRF}}^{0\text{U}} \text{F}_{\text{SRF}}^{\text{SRF}} + \text{R}_{\text{SLF}}^{0\text{U}} \text{F}_{\text{SLF}}^{\text{SLF}} + \text{R}_{\text{SLR}}^{0\text{U}} \text{F}_{\text{SLR}}^{\text{SLR}} + \text{R}_{4\text{RF}}^{0\text{H}} \text{f}_{4\text{RF}}^{4\text{RF}} + \text{R}_{4\text{RR}}^{0\text{H}} \text{f}_{4\text{RR}}^{4\text{RR}} + \\ & \text{R}_{4\text{LF}}^{0\text{H}} \text{f}_{4\text{LF}}^{4\text{LF}} + \text{R}_{\text{SRR}}^{0\text{U}} \text{f}_{3\text{RF}}^{3\text{RF}} + \text{R}_{\text{SLR}}^{0\text{R}} \text{f}_{3\text{RR}}^{3\text{RR}} + \text{R}_{4\text{RF}}^{0\text{H}} \text{f}_{4\text{RF}}^{4\text{RF}} + \text{R}_{4\text{RR}}^{0\text{H}} \text{f}_{4\text{RR}}^{4\text{RR}} + \\ & \text{R}_{4\text{LF}}^{0\text{H}} \text{f}_{4\text{LF}}^{4\text{LR}} + \text{R}_{1\text{L}}^{0\text{H}} \text{f}_{3\text{RF}}^{3\text{RF}} + \text{R}_{\text{SLR}}^{0\text{H}} \text{f}_{3\text{LF}}^{3\text{LF}} + \text{R}_{4\text{RF}}^{0\text{H}} \text{f}_{4\text{RR}}^{4\text{RR}} + \\ & \text{R}_{4\text{LF}}^{0\text{H}} \text{f}_{4\text{LF}}^{4\text{LF}} + \text{R}_{4\text{LR}}^{0\text{H}} \text{f}_{4\text{RR}}^{3\text{R}} + \\ & \text{R}_{4\text{LF}}^{0\text{H}} \text{f}_{4\text{LF}}^{4\text{RF}} + \text{R}_{4\text{LR}}^{0\text{H}} \text{f}_{3\text{LR}}^{3\text{LR}} + \\ & \text{R}_{4\text{LF}}^{0\text{H}} \text{f}_{4\text{RF}}^{4\text{RF}} + \text{R}_{4\text{R}}^{0\text{H}} \text{f}_{4\text{RR}}^{4\text{RR}} + \\ & \text{R}_{4\text{LF}}^{0\text{H}} \text{f}_{4\text{LF}}^{4\text{LR}} + \text{R}_{4\text{LR}}^{0\text{H}} \text{f}_{4\text{RR}}^{3\text{LR}} + \\ & \text{R}_{4\text{LF}}^{0\text{H}} \text{f}_{4\text{LF}}^{1\text{L}} + \text{R}_{4\text{L}}^{0\text{H}} \text{f}_{3\text{LR}}^{3\text{L}} + \\ & \text{R}_{4\text{$$

Where, equation F.5 gives the vector summation of the external normal forces exerted on the touching wheel with surface,

$$\sum_{cs=1}^{4} R_{cs}^{U} F_{cs}^{cs} = R_{SRF}^{U} F_{SRF}^{SRF} + R_{SRR}^{U} F_{SRR}^{SRR} + R_{SLF}^{U} F_{SLF}^{SLF} + R_{SLR}^{U} F_{SLR}^{SLR}$$
(F.6)

Also, it gives the vector summations of the forces exerted on center of link masses resulted from gravity and inertial accelerations

$$\sum_{i=0}^{4} R_{i}^{U} f_{i}^{i} = R_{4RF}^{U} f_{4RF}^{4RF} + R_{4RR}^{U} f_{4RR}^{4RR} + R_{4LF}^{U} f_{4LF}^{4LF} + R_{4LR}^{U} f_{4LR}^{4LR} + R_{3RF}^{U} f_{3RF}^{3RF} + R_{3RR}^{U} f_{3RR}^{3RR} + R_{3LF}^{U} f_{3LF}^{3LF} + R_{3LR}^{U} f_{3LR}^{3LR} + R_{2R}^{U} f_{2R}^{2R} + R_{2L}^{U} f_{2L}^{2L} + R_{1R}^{U} f_{1R}^{1R} + R_{1L}^{U} f_{1L}^{1L} + R_{0R}^{U} f_{0R}^{0R}$$
(F.7)

- 2. **The source of universal moment:** The moments exerted on platform are vector summations for:
  - Moment of exerted force on center of link masses resulted from gravity and inertial linear accelerations;  $r_{c,i}^{U} \times (R_i^{U} f_i^{i})$ .
  - Moment exerted on center of link masses resulted from inertial angular accelerations; R<sup>U</sup><sub>i</sub>τ<sup>i</sup><sub>i</sub>.
  - Moments of external normal forces exerted on the end-effectors;  $r_4^U \times (R_s^U F_s^S)$ , exerted on the touching wheel with surface.

Where, i represents link's frame starting from platform link and ending at the wheel links of four manipulators, link by link.

Successive substitutions inside equation E.28 starting from platform frame and ending at end-effectors,

$$\begin{split} T_{0R}^{U} &= r_{4RF}^{U} \times \left( R_{SRF}^{U} F_{SRF}^{SRF} \right) + r_{4RR}^{U} \times \left( R_{SRR}^{U} F_{SRR}^{SRR} \right) + r_{4LF}^{U} \times \left( R_{SLF}^{U} F_{SLF}^{SLF} \right) + r_{4LR}^{U} \times \left( R_{SLR}^{U} F_{SLR}^{SLR} \right) + \\ r_{c,4RF}^{U} \times \left( R_{4RF}^{U} f_{4RF}^{4RF} \right) + r_{c,3RF}^{U} \times \left( R_{3RF}^{U} f_{3RF}^{3RF} \right) + r_{c,4RR}^{U} \times \left( R_{4RR}^{U} f_{4RR}^{4RR} \right) + r_{c,3RR}^{U} \times \left( R_{3RR}^{U} f_{3RR}^{3RR} \right) + \\ r_{c,2R}^{U} \times \left( R_{2R}^{U} f_{2R}^{2R} \right) + r_{c,1R}^{U} \times \left( R_{1R}^{U} f_{1R}^{1R} \right) + r_{c,0R}^{U} \times \left( R_{0R}^{U} f_{0R}^{0R} \right) + r_{c,1L}^{U} \times \left( R_{1L}^{U} f_{1L}^{1L} \right) + r_{c,2L}^{U} \times \left( R_{2L}^{U} f_{2L}^{2L} \right) + \\ r_{c,3LF}^{U} \times \left( R_{3LF}^{U} f_{3LF}^{3LF} \right) + r_{c,4LF}^{U} \times \left( R_{4LF}^{U} f_{4LF}^{4LF} \right) + r_{c,3LR}^{U} \times \left( R_{3LR}^{U} f_{3LR}^{3LR} \right) + r_{c,4LR}^{U} \times \left( R_{4LR}^{U} f_{4LR}^{4LR} \right) + \\ R_{4RF}^{U} \tau_{4RF}^{4RF} + R_{3RF}^{U} \tau_{3RF}^{3RF} + R_{4RR}^{U} \tau_{3RR}^{4RR} + R_{3RR}^{U} \tau_{3RR}^{3RR} + R_{4LF}^{U} \tau_{4LF}^{4LF} + R_{3LF}^{U} \tau_{3LF}^{3LF} + R_{4LR}^{U} \tau_{4LR}^{4LR} + R_{3LR}^{U} \tau_{3LR}^{3LR} + \\ R_{2R}^{U} \tau_{2R}^{2R} + R_{1R}^{U} \tau_{1R}^{1R} + R_{2L}^{U} \tau_{2L}^{2L} + R_{1L}^{U} \tau_{1L}^{1L} + R_{0R}^{U} \tau_{0R}^{0R} \end{split}$$

#### Matlab Code

#### main\_menu\_dynamic.m

function m= main\_menu\_dynamic()

close all; Locomotion Case = 'DN'

while 1,

clc

which = menu('\_\_\_\_Dynamic Case\_\_\_\_', ... '1. Wheels Motion on Flat Surface......, .... '2. Wheels, RFDJ and RRDJ Motion on Flat Surface......', ... ' 3. Wheels, RCJ and LCJ Motion on Flat Surface......', ... '4. Wheels, RCJ,LCJ,RDJ,& LDJ Motion on Flat Surface ', ... ' 5. Wheels Motion on Step Flat Surface......, ... '6. Wheels, RFDJ and RRDJ Motions on Step Flat Surface..', ... '7. Wheels, RDJ and LDJ Motion on Inclined Surface...,,,,', ... '8. Wheels Motion on Flat & Inclined Surface......, .... '9. Wheels Motion on Sinusoidal Surface......, .... '10. Wheels Motion on Non-uniform Surface......', .... 'Exit'); if which == 1, close all; Surface\_geometry = 'F'; % GG1 Touch =  $[1 \ 1 \ 1 \ 1]';$ Rover 1 % q= Conf 0; q= Conf 1; vv = 2; processes; Moment1 = Momentt; Tow\_CRear1 = Tow\_CRear; Tow\_CFront1 = Tow\_CFront; Tow\_CLeft1 = Tow\_CLeft; Tow\_CRight1 = Tow\_CRight; Normal Forces1 = Normal Forces; qm1 = qm;Force1 = Force; fc1 = fc;fc Moment1 = fc Moment; towc1 = towc; f\_gravity1 = f\_gravity;  $f_{inertial1} = f_{inertial};$ f\_gravity\_Moment1 = f\_gravity\_Moment; f inertial Moment1 = f inertial Moment; NF\_moment1 = NF\_moment;

vv = 4; processes; Moment2 = Momentt; Tow CRear2 = Tow CRear; Tow CFront2 = Tow CFront; Tow CLeft2 = Tow CLeft; Tow CRight2 = Tow CRight; Normal Forces2 = Normal Forces; am2 = am;Force2 = Force; fc2 = fc;fc Moment2 = fc Moment; towc2 = towc;f gravity2 = f gravity; f inertial2 = f inertial; f gravity Moment2 = f gravity Moment; f inertial Moment2 = f inertial Moment; NF moment2 = NF moment; vv = 5; processes; Moment3 = Momentt; Tow CRear3 = Tow CRear; Tow CFront3 = Tow CFront; Tow CLeft3 = Tow CLeft; Tow CRight3 = Tow CRight; Normal Forces3 = Normal Forces; qm3 = qm;Force3 = Force; fc3 = fc;fc Moment3 = fc Moment; towc3 = towc;f\_gravity3 = f\_gravity; f\_inertial3 = f\_inertial; f\_gravity\_Moment3 = f\_gravity\_Moment; f\_inertial\_Moment3 = f\_inertial\_Moment; NF\_moment3 = NF\_moment; elseif which == 2, close all; Surface\_geometry = 'F'; % GG1 Touch =  $[1 \ 1 \ 1 \ 1]';$ Rover\_5 % q= Conf\_5; q= Conf\_0;

vv = 2; processes; Moment1 = Momentt; Tow\_CRear1 = Tow\_CRear; Tow\_CFront1 = Tow\_CFront; Tow\_CLeft1 = Tow\_CLeft; Tow\_CRight1 = Tow\_CRight; Normal\_Forces1 = Normal\_Forces; qm1 = qm; Force1 = Force; fc1 = fc; fc\_Moment1 = fc\_Moment; towc1 = towc; f\_gravity1 = f\_gravity; f\_inertial1 = f\_inertial; f\_gravity\_Moment1 = f\_gravity\_Moment;

f inertial Moment1 = f inertial Moment; NF moment1 = NF moment; vv = 4; processes; Moment2 = Momentt; Tow CRear2 = Tow CRear; Tow CFront2 = Tow CFront; Tow CLeft2 = Tow CLeft; Tow CRight2 = Tow CRight; Normal Forces2 = Normal Forces; qm2 = qm;Force2 = Force; fc2 = fc;fc Moment2 = fc Moment; towc2 = towc;f gravity2 = f gravity; f inertial2 = f inertial; f gravity Moment2 = f gravity Moment; f inertial Moment2 = f inertial Moment; NF moment2 = NF moment; vv = 5; processes; Moment3 = Momentt; Tow CRear3 = Tow CRear; Tow CFront3 = Tow CFront; Tow CLeft3 = Tow CLeft; Tow CRight3 = Tow CRight; Normal Forces3 = Normal Forces; qm3 = qm;Force3 = Force; fc3 = fc;fc\_Moment3 = fc\_Moment; towc3 = towc;f\_gravity3 = f\_gravity; f\_inertial3 = f\_inertial; f\_gravity\_Moment3 = f\_gravity\_Moment; f\_inertial\_Moment3 = f\_inertial\_Moment; NF\_moment3 = NF\_moment; elseif which == 3, close all; Surface\_geometry = 'F'; % GG1 Touch =  $[1 \ 1 \ 1 \ 1]';$ Rover\_7 % q= Conf\_6; q= Conf\_0; vv = 2; processes; Moment1 = Momentt; Tow CRear1 = Tow CRear; Tow CFront1 = Tow CFront; Tow CLeft1 = Tow CLeft; Tow CRight1 = Tow CRight; Normal Forces1 = Normal Forces;

Normal\_Forces1 = Normal\_Forces; qm1 = qm; Force1 = Force; fc1 = fc; fc\_Moment1 = fc\_Moment;

towc1 = towc;

233
f gravity1 = f gravity; f inertial1 = f inertial; f gravity Moment1 = f gravity Moment; f inertial Moment1 = f inertial Moment; NF moment1 = NF moment; vv = 4; processes; Moment2 = Momentt; Tow CRear2 = Tow CRear; Tow CFront2 = Tow CFront; Tow CLeft2 = Tow CLeft; Tow CRight2 = Tow CRight; Normal Forces2 = Normal Forces; qm2 = qm;Force2 = Force; fc2 = fc;fc Moment2 = fc Moment; towc2 = towc;f gravity2 = f gravity; f inertial2 = f inertial; f gravity Moment2 = f gravity Moment; f inertial Moment2 = f inertial Moment; NF moment2 = NF moment; vv = 5; processes; Moment3 = Momentt; Tow CRear3 = Tow CRear; Tow CFront3 = Tow CFront; Tow CLeft3 = Tow CLeft; Tow CRight3 = Tow CRight; Normal\_Forces3 = Normal\_Forces; qm3 = qm;Force3 = Force; fc3 = fc;fc\_Moment3 = fc\_Moment; towc3 = towc;f\_gravity3 = f\_gravity; f\_inertial3 = f\_inertial; f\_gravity\_Moment3 = f\_gravity\_Moment; f\_inertial\_Moment3 = f\_inertial\_Moment; NF\_moment3 = NF\_moment; elseif which == 4, close all; Surface\_geometry = 'F'; % GG1 Touch =  $[1 \ 1 \ 1 \ 1]';$ Rover\_8 % q= Conf\_7; q= Conf\_0; vv = 2; processes; Moment1 = Momentt; Tow CRear1 = Tow CRear; Tow CFront1 = Tow CFront; Tow CLeft1 = Tow CLeft; Tow CRight1 = Tow CRight;

Normal\_Forces1 = Normal\_Forces; qm1 = qm;

Force1 = Force;

fc1 = fc;fc Moment1 = fc Moment; towc1 = towc;f gravity1 = f gravity; f inertial1 = f inertial; f gravity Moment1 = f gravity Moment; f inertial Moment1 = f inertial Moment; NF moment1 = NF moment; vv = 4; processes; Moment2 = Momentt; Tow CRear2 = Tow CRear; Tow CFront2 = Tow CFront; Tow CLeft2 = Tow CLeft; Tow CRight2 = Tow CRight; Normal Forces2 = Normal Forces; qm2 = qm;Force2 = Force; fc2 = fc;fc Moment2 = fc Moment; towc2 = towc;f gravity2 = f gravity; f inertial2 = f inertial; f gravity Moment2 = f gravity Moment; f inertial Moment2 = f inertial Moment; NF moment2 = NF moment; vv = 5; processes; Moment3 = Momentt; Tow\_CRear3 = Tow\_CRear; Tow\_CFront3 = Tow\_CFront; Tow\_CLeft3 = Tow\_CLeft; Tow\_CRight3 = Tow\_CRight; Normal\_Forces3 = Normal\_Forces; qm3 = qm;Force3 = Force; fc3 = fc;fc Moment3 = fc Moment; towc3 = towc;f\_gravity3 = f\_gravity; f\_inertial3 = f\_inertial; f\_gravity\_Moment3 = f\_gravity\_Moment; f\_inertial\_Moment3 = f\_inertial\_Moment; NF\_moment3 = NF\_moment; elseif which == 5, close all; Surface\_geometry = 'S'; % GG2 Touch =  $[1 \ 0 \ 1 \ 1]';$ % q= Conf 5; q= Conf 5; Rover 6

vv = 2; processes; Moment1 = Momentt; Tow\_CRear1 = Tow\_CRear; Tow\_CFront1 = Tow\_CFront; Tow\_CLeft1 = Tow\_CLeft; Tow\_CRight1 = Tow\_CRight; Normal\_Forces1 = Normal\_Forces; qm1 = qm; Force1 = Force; fc1 = fc; fc\_Moment1 = fc\_Moment; towc1 = towc; f\_gravity1 = f\_gravity; f\_inertial1 = f\_inertial; f\_gravity\_Moment1 = f\_gravity\_Moment; NF\_moment1 = NF\_moment;

vv = 4; processes; Moment2 = Momentt; Tow\_CRear2 = Tow\_CRear; Tow\_CFront2 = Tow\_CFront; Tow\_CLeft2 = Tow\_CLeft; Tow\_CRight2 = Tow\_CRight; Normal\_Forces2 = Normal\_Forces; qm2 = qm; Force2 = Force; fc2 = fc; fc\_Moment2 = fc\_Moment; towc2 = towc; f\_gravity2 = f\_gravity; f\_inertial2 = f\_inertial; f\_gravity\_Moment2 = f\_gravity\_Moment; f\_inertial\_Moment2 = f\_inertial\_Moment; NF\_moment2 = NF\_moment;

vv = 5; processes; Moment3 = Momentt; Tow\_CRear3 = Tow\_CRear; Tow\_CFront3 = Tow\_CFront; Tow\_CLeft3 = Tow\_CLeft; Tow\_CRight3 = Tow\_CRight; Normal\_Forces3 = Normal\_Forces; qm3 = qm; Force3 = Force; fc3 = fc; fc\_Moment3 = fc\_Moment; towc3 = towc; f\_gravity3 = f\_gravity; f\_inertial3 = f\_inertial; f\_gravity\_Moment3 = f\_gravity\_Moment; f\_inertial\_Moment3 = f\_inertial\_Moment; NF\_moment3 = NF\_moment;

elseif which == 6, close all; Surface\_geometry = 'S'; % GG2 Touch = [1 0 1 1]';

Rover\_1 % q= Conf\_0; q= Conf\_1; vv = 2; processes; Moment1 = Momentt; 236

Tow\_CRear1 = Tow\_CRear; Tow\_CFront1 = Tow\_CFront; Tow\_CLeft1 = Tow\_CLeft; Tow\_CRight1 = Tow\_CRight; Normal\_Forces1 = Normal\_Forces; qm1 = qm; Force1 = Force; fc1 = fc; fc\_Moment1 = fc\_Moment; towc1 = towc; f\_gravity1 = f\_gravity; f\_inertial1 = f\_inertial; f\_gravity\_Moment1 = f\_gravity\_Moment; f\_inertial\_Moment1 = f\_inertial\_Moment; NF\_moment1 = NF\_moment;

% q= Conf 0; q= Conf 9; Rover 10 vv = 2; processes; Moment2 = Momentt; Tow CRear2 = Tow CRear; Tow CFront2 = Tow CFront; Tow CLeft2 = Tow CLeft; Tow CRight2 = Tow CRight; Normal Forces2 = Normal Forces; qm2 = qm;Force2 = Force; fc2 = fc;fc Moment2 = fc Moment; towc2 = towc;f gravity2 = f gravity; f\_inertial2 = f\_inertial; f\_gravity\_Moment2 = f\_gravity\_Moment; f\_inertial\_Moment2 = f\_inertial\_Moment; NF\_moment2 = NF\_moment;

% q= Conf\_0; q= Conf\_8; Rover 9 vv = 2; processes; Moment3 = Momentt; Tow\_CRear3 = Tow\_CRear; Tow\_CFront3 = Tow\_CFront; Tow\_CLeft3 = Tow\_CLeft; Tow\_CRight3 = Tow\_CRight; Normal Forces3 = Normal Forces; qm3 = qm;Force3 = Force; fc3 = fc;fc\_Moment3 = fc\_Moment; towc3 = towc;f\_gravity3 = f\_gravity; f inertial3 = f inertial; f\_gravity\_Moment3 = f\_gravity\_Moment; f inertial Moment3 = f inertial Moment; NF moment3 = NF moment;

elseif which == 7, close all; Surface\_geometry = 'I'; % GG9 Touch = [1 1 1 1]'; Rover 1 % q= Conf 0; q= Conf 1; vv = 2.5; processes; Moment1 = Momentt; Tow CRear1 = Tow CRear; Tow CFront1 = Tow CFront; Tow CLeft1 = Tow CLeft; Tow CRight1 = Tow CRight; Normal Forces1 = Normal Forces; am1 = am;Force1 = Force; fc1 = fc;fc Moment1 = fc Moment; towc1 = towc;f gravity1 = f gravity; f inertial1 = f inertial; f gravity Moment1 = f\_gravity\_Moment; f inertial Moment1 = f inertial Moment; NF moment1 = NF moment; % q= Conf\_0; q= Conf\_11; Rover 12 vv = 2.5; processes; Moment2 = Momentt; Tow CRear2 = Tow CRear; Tow CFront2 = Tow CFront; Tow CLeft2 = Tow CLeft; Tow CRight2 = Tow CRight; Normal Forces2 = Normal Forces; qm2 = qm;Force2 = Force; fc2 = fc;fc\_Moment2 = fc\_Moment; towc2 = towc;f\_gravity2 = f\_gravity; f\_inertial2 = f\_inertial; f\_gravity\_Moment2 = f\_gravity\_Moment; f\_inertial\_Moment2 = f\_inertial\_Moment; NF\_moment2 = NF\_moment; % q= Conf\_0; q= Conf\_10; Rover 11 vv = 2.5; processes; Moment3 = Momentt; Tow\_CRear3 = Tow\_CRear; Tow\_CFront3 = Tow\_CFront; Tow\_CLeft3 = Tow\_CLeft; Tow\_CRight3 = Tow\_CRight; Normal\_Forces3 = Normal\_Forces; qm3 = qm;Force3 = Force; fc3 = fc;fc Moment3 = fc Moment; towc3 = towc;f\_gravity3 = f\_gravity; f inertial3 = f inertial; f\_gravity\_Moment3 = f\_gravity\_Moment; f\_inertial\_Moment3 = f\_inertial\_Moment; NF\_moment3 = NF\_moment;

elseif which == 8, close all; Surface geometry = 'FI'; % GG5 Touch =  $[1 \ 1 \ 1 \ 1]';$ % q= Conf\_0; q= Conf\_1; Rover 1 vv = 0.5; processes; Moment1 = Momentt; Tow CRear1 = Tow CRear; Tow CFront1 = Tow CFront; Tow CLeft1 = Tow CLeft; Tow CRight1 = Tow CRight; Normal Forces1 = Normal Forces; qm1 = qm;Force1 = Force; fc1 = fc;fc Moment1 = fc Moment; towc1 = towc;f gravity1 = f gravity; f inertial1 = f inertial; f gravity Moment1 = f gravity Moment; f inertial Moment1 = f inertial Moment; NF moment1 = NF moment; Rover 14 % q= Conf 13; q= Conf 13; vv = 0.5; processes; Moment2 = Momentt; Tow CRear2 = Tow CRear; Tow CFront2 = Tow CFront; Tow\_CLeft2 = Tow\_CLeft; Tow\_CRight2 = Tow\_CRight; Normal\_Forces2 = Normal\_Forces; qm2 = qm;Force2 = Force; fc2 = fc;fc\_Moment2 = fc\_Moment; towc2 = towc;f\_gravity2 = f\_gravity; f\_inertial2 = f\_inertial; f\_gravity\_Moment2 = f\_gravity\_Moment; f\_inertial\_Moment2 = f\_inertial\_Moment; NF\_moment2 = NF\_moment; Rover\_13 % q= Conf\_12; q= Conf\_12; vv = 0.5; processes; Moment3 = Momentt; Tow CRear3 = Tow CRear; Tow CFront3 = Tow CFront; Tow\_CLeft3 = Tow\_CLeft; Tow\_CRight3 = Tow\_CRight; Normal Forces3 = Normal Forces; qm3 = qm;Force3 = Force; fc3 = fc;fc Moment3 = fc Moment; towc3 = towc;f\_gravity3 = f\_gravity; f inertial3 = f\_inertial;

f gravity Moment3 = f gravity Moment; f inertial Moment3 = f inertial Moment; NF moment3 = NF moment; elseif which == 9, close all; Surface geometry = 'D'; % GG7 Touch =  $[1 \ 1 \ 1 \ 1]';$ % q= Conf 15; q= Conf 15; Rover 14 vv = 1; processes; Moment1 = Momentt; Tow CRear1 = Tow CRear; Tow CFront1 = Tow CFront; Tow CLeft1 = Tow CLeft; Tow CRight1 = Tow CRight; Normal Forces1 = Normal Forces; qm1 = qm;Force1 = Force; fc1 = fc;fc\_Moment1 = fc\_Moment; towc1 = towc;f gravity1 = f gravity; f inertial1 = f inertial; f gravity Moment1 = f gravity Moment; f inertial Moment1 = f inertial Moment; NF moment1 = NF moment; % q= Conf\_15; q= Conf\_15; Rover\_14 vv = 2; processes; Moment2 = Momentt; Tow\_CRear2 = Tow\_CRear; Tow\_CFront2 = Tow\_CFront; Tow\_CLeft2 = Tow\_CLeft; Tow\_CRight2 = Tow\_CRight; Normal\_Forces2 = Normal\_Forces; qm2 = qm;Force2 = Force; fc2 = fc;fc Moment2 = fc Moment; towc2 = towc;f\_gravity2 = f\_gravity; f\_inertial2 = f\_inertial; f\_gravity\_Moment2 = f\_gravity\_Moment; f\_inertial\_Moment2 = f\_inertial\_Moment; NF\_moment2 = NF\_moment; Rover 14 % q= Conf\_15; q= Conf\_15; vv = 3; processes; Moment3 = Momentt; Tow CRear3 = Tow CRear; Tow CFront3 = Tow CFront; Tow CLeft3 = Tow CLeft; Tow CRight3 = Tow CRight; Normal Forces3 = Normal Forces; qm3 = qm;Force3 = Force; fc3 = fc;

fc\_Moment3 = fc\_Moment; towc3 = towc; f\_gravity3 = f\_gravity; f\_inertial3 = f\_inertial; f\_inertial\_Moment3 = f\_gravity\_Moment; f\_inertial\_Moment3 = f\_inertial\_Moment; NF\_moment3 = NF\_moment;

elseif which == 10, close all; Surface geometry = 'U'; % GG11 Touch =  $[1 \ 1 \ 1 \ 1]';$ % q= Conf 0; q= Conf 14; Rover 16 vv = 0.05; processes; Moment1 = Momentt; Tow CRear1 = Tow CRear; Tow CFront1 = Tow CFront; Tow CLeft1 = Tow CLeft; Tow CRight1 = Tow CRight; Normal Forces1 = Normal Forces; qm1 = qm;Force1 = Force; fc1 = fc;fc Moment1 = fc Moment; towc1 = towc;f gravity1 = f gravity; f\_inertial1 = f\_inertial; f\_gravity\_Moment1 = f\_gravity\_Moment; f\_inertial\_Moment1 = f\_inertial\_Moment; NF\_moment1 = NF\_moment; Rover\_16 % q= Conf\_0; q= Conf\_14; vv = 0.05; processes; Moment2 = Momentt; Tow\_CRear2 = Tow\_CRear; Tow\_CFront2 = Tow\_CFront;

Normal\_Forces2 = Normal\_Forces; qm2 = qm; Force2 = Force; fc2 = fc; fc\_Moment2 = fc\_Moment; towc2 = towc; f\_gravity2 = f\_gravity; f\_inertial2 = f\_inertial; f\_gravity\_Moment2 = f\_gravity\_Moment; f\_inertial\_Moment2 = f\_inertial\_Moment; NF\_moment2 = NF\_moment; Rover\_16 % q= Conf\_0; q= Conf\_14; vv = 1; processes; Moment3 = Momentt;

Tow\_CLeft2 = Tow\_CLeft; Tow\_CRight2 = Tow\_CRight;

Tow\_CRear3 = Tow\_CRear; Tow\_CFront3 = Tow\_CFront;

```
Tow CLeft3 = Tow CLeft; Tow CRight3 = Tow CRight;
    Normal Forces3 = Normal Forces;
     am3 = am;
     Force3 = Force;
     fc3 = fc;
     fc Moment3 = fc Moment;
    towc3 = towc;
     f gravity3 = f gravity;
     f inertial3 = f inertial;
     f gravity Moment3 = f gravity Moment;
     f inertial Moment3 = f inertial Moment;
     NF moment3 = NF moment;
  elseif which == 11,
     close all;
                 break;
         end
%
% Figures of Results
%
figure(1)
  subplot(3,1,1)
  hold on
  plot(t, qm1(1,1:np)*180/pi, 'k-', 'linewidth',2)
  plot(t, qm2(1,1:np)*180/pi, 'g-', 'linewidth',2)
  plot(t, qm3(1,1:np)*180/pi, 'b-', 'linewidth',2)
  title('platform orientation angles w/2 universal frame')
  ylabel('Psi (deg)')
  grid
  hold off
  subplot(3,1,2)
  hold on
  plot(t, qm1(2,1:np)*180/pi, 'k-', 'linewidth',2)
  plot(t, qm2(2,1:np)*180/pi, 'g-', 'linewidth',2)
  plot(t, qm3(2,1:np)*180/pi, 'b-', 'linewidth',2)
  ylabel('Phi (deg)')
  grid
  hold off
  subplot(3,1,3)
  hold on
  plot(t, qm1(3,1:np)*180/pi, 'k', 'linewidth',2)
  plot(t, qm2(3,1:np)*180/pi, 'g', 'linewidth',2)
  plot(t, qm3(3,1:np)*180/pi, 'b', 'linewidth',2)
  xlabel('Time (s)');
  ylabel('Theta (deg)')
  grid
  hold off
```

#### figure(2)

```
subplot(4,1,1)
 hold on
  %axis([.01 np-1 -1 30])
  plot(t, Normal Forces1(1,1:np), 'k-', 'linewidth',2)
 plot(t, Normal_Forces2(1,1:np), 'g-', 'linewidth',2)
 plot(t, Normal_Forces3(1,1:np), 'b-', 'linewidth',2)
  title('Normal force exerted on contact wheels')
 ylabel('FnRF')
 grid
 hold off
  subplot(4,1,2)
 hold on
  %axis([.01 np-1 -1 15])
  plot(t, Normal Forces1(2,1:np), 'k-', 'linewidth',2)
  plot(t, Normal_Forces2(2,1:np), 'g-', 'linewidth',2)
  plot(t, Normal Forces3(2,1:np), 'b-', 'linewidth',2)
 ylabel('FnRR')
  grid
 hold off
  subplot(4,1,3)
 hold on
  %axis([.01 np-1 -1 30])
  plot(t, Normal_Forces1(3,1:np), 'k-', 'linewidth',2)
  plot(t, Normal_Forces2(3,1:np), 'g-', 'linewidth',2)
  plot(t, Normal_Forces3(3,1:np), 'b-', 'linewidth',2)
 ylabel('FnLF')
  grid
  hold off
 subplot(4,1,4)
 hold on
  %axis([.01 np-1 -1 15])
  plot(t, Normal_Forces1(4,1:np), 'k-', 'linewidth',2)
  plot(t, Normal_Forces2(4,1:np), 'g-', 'linewidth',2)
  plot(t, Normal_Forces3(4,1:np), 'b-', 'linewidth',2)
  xlabel('Time (s)');
  ylabel('FnLR')
  grid
 hold off
figure(3)
 hold on
 plot(t, Tow_CRear1(1:np),'k--', 'linewidth',2)
 plot(t, Tow_CRear2(1:np),'g--', 'linewidth',2)
plot(t, Tow_CRear3(1:np),'b--', 'linewidth',2)
```

plot(t, Moment1(3,1:np),'k-', 'linewidth',2)

```
plot(t, Moment2(3,1:np),'g-', 'linewidth',2)
plot(t, Moment3(3,1:np),'b-', 'linewidth',2)
```

```
plot(t, Tow_CFront1(1:np),'k--', 'linewidth',2)
plot(t, Tow_CFront2(1:np),'g--', 'linewidth',2)
plot(t, Tow_CFront3(1:np),'b--', 'linewidth',2)
```

```
title('exerted Moment about zu-axis of universal frame')
ylabel('TU(3) (N.m)')
xlabel('Time (s)');
grid
hold off
```

figure(4)

```
hold on
 plot(t, Tow CLeft1(1:np),'k--', 'linewidth',2)
 plot(t, Tow_CLeft2(1:np),'g--', 'linewidth',2)
 plot(t, Tow_CLeft3(1:np),'b--', 'linewidth',2)
 plot(t, Moment1(2,1:np),'k-', 'linewidth',2)
 plot(t, Moment2(2,1:np),'g-', 'linewidth',2)
 plot(t, Moment3(2,1:np),'b-', 'linewidth',2)
 plot(t, Tow CRight1(1:np),'k--', 'linewidth',2)
 plot(t, Tow CRight2(1:np),'g--', 'linewidth',2)
 plot(t, Tow_CRight3(1:np),'b--', 'linewidth',2)
  title('exerted Moment about yu-axis of universal frame')
  ylabel('TU(2) (N.m)')
  xlabel('Time (s)');
  grid
 hold off
figure(5)
 subplot(3,1,1)
 hold on
 plot(t, Force1(1,1:np), 'k-', 'linewidth',2)
  plot(t, Force2(1,1:np), 'g-', 'linewidth',2)
  plot(t, Force3(1,1:np), 'b-', 'linewidth',2)
  title('Universal force (N)')
  ylabel('F_X_u')
  grid
 hold off
 subplot(3,1,2)
 hold on
 plot(t, Force1(2,1:np), 'k-', 'linewidth',2)
 plot(t, Force2(2,1:np), 'g-', 'linewidth',2)
plot(t, Force3(2,1:np), 'b-', 'linewidth',2)
  ylabel('F_Y_u')
  grid
```

hold off

```
subplot(3,1,3)
 hold on
  plot(t, Force1(3,1:np), 'k', 'linewidth',2)
 plot(t, Force2(3,1:np), 'g', 'linewidth',2)
 plot(t, Force3(3,1:np), 'b', 'linewidth',2)
  xlabel('Time (s)');
  ylabel('F Z u')
  grid
 hold off
figure(6)
 subplot(3,1,1)
 hold on
  plot(t, fc1(1,1:np), 'k-', 'linewidth',2)
 plot(t, fc2(1,1:np), 'g-', 'linewidth',2)
plot(t, fc3(1,1:np), 'b-', 'linewidth',2)
  title('Universal fc (N)')
  ylabel('fc_X_u')
  grid
 hold off
  subplot(3,1,2)
 hold on
 plot(t, fc1(2,1:np), 'k-', 'linewidth',2)
 plot(t, fc2(2,1:np), 'g-', 'linewidth',2)
  plot(t, fc3(2,1:np), 'b-', 'linewidth',2)
  ylabel('fc_Y_u')
  grid
  hold off
 subplot(3,1,3)
 hold on
 plot(t, fc1(3,1:np), 'k', 'linewidth',2)
 plot(t, fc2(3,1:np), 'g', 'linewidth',2)
 plot(t, fc3(3,1:np), 'b', 'linewidth',2)
  xlabel('Time (s)');
  ylabel('fc_Z_u')
  grid
 hold off
figure(7)
 subplot(3,1,1)
 hold on
 plot(t, fc_Moment1(1,1:np), 'k-', 'linewidth',2)
  plot(t, fc_Moment2(1,1:np), 'g-', 'linewidth',2)
  plot(t, fc_Moment3(1,1:np), 'b-', 'linewidth',2)
  title('Universal fc Moment (N.m)')
  ylabel('fc Moment_X_u')
  grid
  hold off
```

```
subplot(3,1,2)
 hold on
 plot(t, fc Moment1(2,1:np), 'k-', 'linewidth',2)
 plot(t, fc_Moment2(2,1:np), 'g-', 'linewidth',2)
 plot(t, fc_Moment3(2,1:np), 'b-', 'linewidth',2)
  ylabel('fc Moment_Y_u')
  grid
 hold off
 subplot(3,1,3)
 hold on
 plot(t, fc Moment1(3,1:np), 'k', 'linewidth',2)
 plot(t, fc Moment2(3,1:np), 'g', 'linewidth',2)
 plot(t, fc Moment3(3,1:np), 'b', 'linewidth',2)
  xlabel('Time (s)');
  ylabel('fc Moment_z_u')
  grid
 hold off
figure(8)
 subplot(3,1,1)
 hold on
 plot(t, towc1(1,1:np), 'k-', 'linewidth',2)
 plot(t, towc2(1,1:np), 'g-', 'linewidth',2)
 plot(t, towc3(1,1:np), 'b-', 'linewidth',2)
  title('Universal towc (N.m)')
 ylabel('towc_x_u')
  grid
  hold off
 subplot(3,1,2)
 hold on
 plot(t, towc1(2,1:np), 'k-', 'linewidth',2)
 plot(t, towc2(2,1:np), 'g-', 'linewidth',2)
 plot(t, towc3(2,1:np), 'b-', 'linewidth',2)
 ylabel('towc_y_u')
  grid
 hold off
 subplot(3,1,3)
 hold on
  plot(t, towc1(3,1:np), 'k', 'linewidth',2)
 plot(t, towc2(3,1:np), 'g', 'linewidth',2)
plot(t, towc3(3,1:np), 'b', 'linewidth',2)
  xlabel('Time (s)');
  ylabel('towc z u')
 grid
 hold off
```

figure(9)

```
subplot(3,1,1)
  hold on
  plot(t, f gravity1(1,1:np), 'k-', 'linewidth',2)
 plot(t, f_gravity2(1,1:np), 'g-', 'linewidth',2)
  plot(t, f_gravity3(1,1:np), 'b-', 'linewidth',2)
  title('Universal gravity force resulted from center of mass of links(N)')
  ylabel('Gravity force x u')
  grid
  hold off
  subplot(3,1,2)
  hold on
  plot(t, f gravity1(2,1:np), 'k-', 'linewidth',2)
  plot(t, f_gravity2(2,1:np), 'g-', 'linewidth',2)
plot(t, f_gravity3(2,1:np), 'b-', 'linewidth',2)
  ylabel('Gravity force y u')
  grid
  hold off
  subplot(3,1,3)
  hold on
  plot(t, f gravity1(3,1:np), 'k', 'linewidth',2)
  plot(t, f gravity2(3,1:np), 'g', 'linewidth',2)
  plot(t, f gravity3(3,1:np), 'b', 'linewidth',2)
  xlabel('Time (s)');
  ylabel('Gravity force z u')
  grid
  hold off
figure(10)
  subplot(3,1,1)
  hold on
  plot(t, f_inertial1(1,1:np), 'k-', 'linewidth',2)
  plot(t, f_inertial2(1,1:np), 'g-', 'linewidth',2)
  plot(t, f_inertial3(1,1:np), 'b-', 'linewidth',2)
  title('Universal inertial force resulted from center of mass of links(N)')
  ylabel('Inertial force x u')
  grid
  hold off
  subplot(3,1,2)
  hold on
  plot(t, f_inertial1(2,1:np), 'k-', 'linewidth',2)
  plot(t, f_inertial2(2,1:np), 'g-', 'linewidth',2)
plot(t, f_inertial3(2,1:np), 'b-', 'linewidth',2)
  ylabel('Inertial force y u')
  grid
  hold off
  subplot(3,1,3)
  hold on
  plot(t, f inertial1(3,1:np), 'k', 'linewidth',2)
```

```
plot(t, f inertial2(3,1:np), 'g', 'linewidth',2)
  plot(t, f inertial3(3,1:np), 'b', 'linewidth',2)
  xlabel('Time (s)');
  ylabel('Inertial force z u')
  grid
  hold off
figure(11)
  subplot(3,1,1)
  hold on
  plot(t, f_gravity_Moment1(1,1:np), 'k-', 'linewidth',2)
  plot(t, f gravity Moment2(1,1:np), 'g-', 'linewidth',2)
  plot(t, f_gravity_Moment3(1,1:np), 'b-', 'linewidth',2)
  title('Universal Moment of gravity force resulted from center of mass of links(N.m)')
  ylabel('Gravity Moment x u')
  grid
  hold off
  subplot(3,1,2)
  hold on
  plot(t, f gravity Moment1(2,1:np), 'k-', 'linewidth',2)
  plot(t, f gravity Moment2(2,1:np), 'g-', 'linewidth',2)
  plot(t, f gravity Moment3(2,1:np), 'b-', 'linewidth',2)
  ylabel('Gravity Moment y u')
  grid
  hold off
  subplot(3,1,3)
  hold on
  plot(t, f_gravity_Moment1(3,1:np), 'k', 'linewidth',2)
  plot(t, f_gravity_Moment2(3,1:np), 'g', 'linewidth',2)
  plot(t, f_gravity_Moment3(3,1:np), 'b', 'linewidth',2)
  xlabel('Time (s)');
  ylabel('Gravity Moment_z_u')
  grid
  hold off
figure(12)
  subplot(3,1,1)
  hold on
  plot(t, f_inertial_Moment1(1,1:np), 'k-', 'linewidth',2)
  plot(t, f_inertial_Moment2(1,1:np), 'g-', 'linewidth',2)
  plot(t, f_inertial_Moment3(1,1:np), 'b-', 'linewidth',2)
  title('Universal Moment of inertial force resulted from center of mass of links(N.m)')
  ylabel('Inertial Moment x u')
  grid
  hold off
  subplot(3,1,2)
  hold on
  plot(t, f_inertial_Moment1(2,1:np), 'k-', 'linewidth',2)
  plot(t, f_inertial_Moment2(2,1:np), 'g-', 'linewidth',2)
```

```
plot(t, f inertial Moment3(2,1:np), 'b-', 'linewidth',2)
 ylabel('Inertial Moment y u')
 grid
 hold off
 subplot(3,1,3)
 hold on
 plot(t, f inertial Moment1(3,1:np), 'k', 'linewidth',2)
 plot(t, f inertial Moment2(3,1:np), 'g', 'linewidth',2)
 plot(t, f inertial Moment3(3,1:np), 'b', 'linewidth',2)
 xlabel('Time (s)');
 ylabel('Inertial Moment z u')
 grid
 hold off
figure(13)
 subplot(3,1,1)
 hold on
 plot(t, NF_moment1(1,1:np), 'k-', 'linewidth',2)
 plot(t, NF_moment2(1,1:np), 'g-', 'linewidth',2)
 plot(t, NF_moment3(1,1:np), 'b-', 'linewidth',2)
 title('Universal moments resulted from normal forces (N.m)')
 ylabel('NF moment x u')
 grid
 hold off
 subplot(3,1,2)
 hold on
 plot(t, NF_moment1(2,1:np), 'k-', 'linewidth',2)
 plot(t, NF_moment2(2,1:np), 'g-', 'linewidth',2)
 plot(t, NF_moment3(2,1:np), 'b-', 'linewidth',2)
 ylabel('NF moment_y_u')
 grid
 hold off
 subplot(3,1,3)
 hold on
 plot(t, NF_moment1(3,1:np), 'k', 'linewidth',2)
 plot(t, NF_moment2(3,1:np), 'g', 'linewidth',2)
 plot(t, NF_moment3(3,1:np), 'b', 'linewidth',2)
 xlabel('Time (s)');
 ylabel('NF moment_z_u')
 grid
 hold off
```

```
end
```

### **Processes.m**

n = numrows(dh dyn); $d = dh_dyn(2:n,2);$  $a = dh_dyn(2:n,3);$  $alpha = dh_dyn(2:n,4);$ r = a(4);% radius of wheel TOL = 0.00001; % tolerance value % % create time vector % t = [0:1:200];np = numcols(t);%\_ % trajectory of joints % % joint coordinate trajectory of right front leg  $[q_RF,qd_RF,qdd_RF] = jtraj(q0(:,1), q1(:,1), t);$  $[q_RR,qd_RR,qdd_RR] = jtraj(q0(:,2), q1(:,2), t);$ % joint coordinate trajectory of right rear leg  $[q_LF,qd_LF,qd_LF] = jtraj(q0(:,3), q1(:,3), t);$ % joint coordinate trajectory of left front leg % joint coordinate trajectory of left rear leg  $[q_LR,qd_LR,qdd_LR] = jtraj(q0(:,4), q1(:,4), t);$ if Locomotion Case == 'ST' [At4RF, At4RR, At4LF, At4LR, Vt4RF, Vt4RR, Vt4LF, Vt4LR, d\_4RF, d\_4RR, d\_4LF, d\_4LR,... Thetadd RF, Thetadd RR, Thetadd LF, Thetadd LR, Thetad RF, Thetad RR, Thetad LF, Thetad LR .... Theta\_RF, Theta\_RR, Theta\_LF, Theta\_LR, tdelay\_R, tdelay\_L] = locomotion\_ST(vv, t, a, q0); elseif Locomotion Case == 'DN' [At4RF, At4RR, At4LF, At4LR, Vt4RF, Vt4RR, Vt4LF, Vt4LR, d 4RF, d 4RR, d 4LF, d 4LR,... Thetadd RF, Thetadd RR, Thetadd LF, Thetadd LR, Thetad RF, Thetad RR, Thetad LF, Thetad LR.... Theta RF, Theta RR, Theta LF, Theta LR, tdelay R, tdelay L] = locomotion DN(Touch, vv, t, a, q0); end  $q_RF(:,4) = Theta_RF; qd_RF(:,4) = Thetad_RF; qd_RF(:,4) = Thetadd_RF;$  $q_R(:,4) = \text{Theta}_RR; qd_RR(:,4) = \text{Theta}_RR; qd_RR(:,4) = \text{Theta}_RR;$  $q_LF(:,4) = Theta_LF; qd_LF(:,4) = Thetad_LF; qdd_LF(:,4) = Thetadd_LF;$  $q_LR(:,4) = Theta_LR; qd_LR(:,4) = Thetad_LR; qdd_LR(:,4) = Thetadd_LR;$  $\label{eq:rescaled_$  $q_LF = q_LF'; qd_LF = qd_LF'; qdd_LF = qdd_LF';$  $q_LR = q_LR'; qd_LR = qd_LR'; qdd_LR = qdd_LR';$ % % % Ground geometry

%

if Surface geometry == 'F'

[input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs, ... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG1(t, tdelay\_R, tdelay\_L, a, q0);

elseif Surface\_geometry == 'S'

[input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs, ... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG2(t, tdelay\_R, tdelay\_L, a, q0);

elseif Surface\_geometry == 'I'

[input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs, ... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG9(t, tdelay\_R, tdelay\_L, a, q0);

elseif Surface\_geometry == 'FI'

[input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs, ... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG5(t, tdelay\_R, tdelay\_L, a, q0);

elseif Surface\_geometry == 'D'

[input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs, ... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG7(t, tdelay\_R, tdelay\_L, a, q0);

elseif Surface geometry == 'U'

[input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs, ... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG11(t, tdelay\_R, tdelay\_L, a, q0,... d\_4RF, d\_4RR, d\_4LF, d\_4LR);

end

theta\_S\_zs = [beta\_SRF\_zs; beta\_SRR\_zs; beta\_SLF\_zs; beta\_SLR\_zs]; theta\_S\_ys = [beta\_SRF\_ys; beta\_SRR\_ys; beta\_SLF\_ys; beta\_SLR\_ys];

#### %\_

% surface flatness check % for p=1:np,

if (input\_RF(p) == input\_RR(p)) && (input\_LF(p) == input\_LR(p)) && (input\_RF(p) ==
input\_LF(p))
Flat\_surface(p) = 1;

```
else
Flat_surface(p) = 0;
end
end
```

#### %

```
% Platform Attitute %
```

Touches = [];

qm=[];

A\_0R\_0L =[1 0 0 0;... 0 -1 0 0;... 0 0 -1 0;... 0 0 0 1]; q\_RK = q\_RF; q\_LK = q\_LF; input RK = input RF;

B3\_RK = beta\_SRF\_zs; B3\_LK = beta\_SLF\_zs;

input LK = input LF;

-\_\_\_\_

for p=1:np,

% Pitch angle theta\_ $1R = q_RK(1, p)$ ; theta\_ $1L = q_LK(1, p)$ ;

 $Theta = (theta_1R - theta_1L)/2 + ... \\ asin((input_RR(p) - input_RF(p))/(-a(3)*sin(q_RF(3,p))+a(3)*sin(q_RR(3,p))));$ 

% Yaw angle theta\_4RK =  $q_RK(4, p)$ ; theta\_4LK =  $q_LK(4, p)$ ;

 $[A_0R_4RK] = Kinematic(q_RK(:,p)', d, a, alpha, B3_RK(p), Theta);$  $[A_0L_4LK] = Kinematic(q_LK(:,p)', d, a, alpha, -B3_LK(p), -Theta);$ 

 $r_0R_4RK = A_0R_4RK(1:3, 4);$   $A_0R_4LK = A_0R_0L * A_0L_4LK;$  $r_0R_4LK = A_0R_4LK(1:3, 4);$ 

 $Psi = (-r*theta_4LK - r*theta_4RK)/(r_0R_4RK(3) - r_0R_4LK(3));$ 

%\_\_\_\_Roll angle angle\_\_\_\_ r\_4RK\_G\_xU = -input\_RK(p);

```
r_4LK_G_xU = -input_LK(p);
          AA = r_0R_4RK(1) - r_0R_4LK(1);
          if abs(AA) < TOL
           AA = 0;
          end
          BB = r_0R_4RK(3) - r_0R_4LK(3) - 4*d(1);
          if abs(BB) < TOL
           BB = 0;
          end
          if(BB) \ge 0
              alfa = -atan2(AA,BB);
          elseif(BB) < 0
           if AA < 0
              alfa = -pi - atan2(AA,(BB));
           elseif AA \ge 0
              alfa = pi - atan2(AA,(BB));
           end
          end
  Phi = asin((-r_4RK_G_xU + r_4LK_G_xU)/...
     (sqrt((r_0R_4RK(1) - r_0R_4LK(1))^2 + (r_0R_4RK(3) - r_0R_4LK(3) - 4*d(1))^2))) - alfa;
   Ph= Phi *180/pi;
      angle_0 = [Psi; Phi; Theta];
       qm = [qm angle_0];
%
% Contact Points in case of random surface
%
if Surface_geometry == 'U'
 if Phi > 0
   Touch_Legs = [1 \ 0 \ 1 \ 1]';
 elseif Phi < 0
   Touch_Legs = [1 \ 1 \ 1 \ 0]';
 elseif Phi == 0
   Touch_Legs = Touch;
 end
else
  Touch Legs = Touch;
end
Touches = [Touches Touch_Legs];
% contact check
```

A\_U\_0R = roty(Phi) \* rotz(Theta) \* rotx(Psi);

 $A_U_4RK = A_U_0R * A_0R_4RK;$  $r_U_4RK = A_U_4RK(1:3, 4);$ 

 $r_4RK_G = [r_4RK_G_xU - r_U_4RK(2) - r_U_4RK(3)]';$ 

 $r_U_G = r_U_4RK + r_4RK_G;$ 

 $\begin{array}{cccccc} A\_U\_G = \begin{bmatrix} 1 & 0 & 0 & & r\_U\_G(1) \\ & 0 & 1 & 0 & & r\_U\_G(2) \\ & 0 & 0 & 1 & & r\_U\_G(3) \\ & 0 & 0 & 0 & & 1 & \end{bmatrix}; \end{array}$ 

 $A\_G\_U = inv(A\_U\_G);$   $A\_G\_4RK = A\_G\_U * A\_U\_0R * A\_0R\_4RK;$  ss='RF';  $A\_0R\_U = inv(A\_U\_0R);$   $A\_0R\_4RK = A\_0R\_U * A\_U\_G * A\_G\_4RK;$ theta RK = invkinematic(A 0R 4RK, ss);

%-----A\_U\_4LK = A\_U\_0R \* A\_0R\_4LK; r\_U\_4LK = A\_U\_4LK(1:3, 4);

 $r_4LK_G = [r_4LK_G_xU - r_U_4LK(2) - r_U_4LK(3)]';$ 

 $r_U_G = r_U_4LK + r_4LK_G;$ 

$A_U_G = [1 \ 0 \ 0]$	r_U_G(1)
0 1 0	$r_U_G(2)$
0 0 1	$r_U_G(3)$
$0 \ 0 \ 0$	1 ];
$\begin{array}{c} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array}$	r_U_G(3) 1 ];

 $A\_G\_U = inv(A\_U\_G);$   $A\_G\_4LK = A\_G\_U * A\_U\_0R * A\_0R\_0L * A\_0L\_4LK;$  ss='LF';  $A\_0L\_0R = inv(A\_0R\_0L);$   $A\_0L\_4LK = A\_0L\_0R * A\_0R\_U * A\_U\_G * A\_G\_4LK;$  $theta\_LK = invkinematic(A\_0L\_4LK, ss);$ 

end

qmd = zeros(3, np); qmdd = zeros(3, np); [nr, nc] = size(qm); % test for accuracy for i=1:nr

```
for j=1:nc
    if abs(qm(i,j)) < TOL
    qm(i,j) = 0;
    end
end
end</pre>
```

%

% Tansformation Matrix of wheel frame, universal wheel frame, surface frame

R4RF\_SRF = []; R4RR\_SRR = []; R4LF\_SLF = []; R4LR\_SLR = [];

for p=1:np

A U 0R = roty(qm(2,p)) \* rotz(qm(3,p)) \* rotx(qm(1,p));

 $[A_0R_4RF] = Kinematic(q_RF(:,p)', d, a, alpha, beta_SRF_zs(p), qm(3,p));$  $[A_0R_4RR] = Kinematic(q_RR(:,p)', d, a, alpha, beta_SRR_zs(p), qm(3,p));$  $[A_0L_4LF] = Kinematic(q_LF(:,p)', d, a, alpha, beta_SLF_zs(p), qm(3,p));$  $[A_0L_4LR] = Kinematic(q_LR(:,p)', d, a, alpha, beta_SLR_zs(p), qm(3,p));$  $[A_0L_4LR] = Kinematic(q_LR(:,p)', d, a, alpha, beta_SLR_zs(p), qm(3,p)); \\$ 

 $\begin{array}{l} A\_U\_4RF = A\_U\_0R * A\_0R\_4RF; \\ A\_U\_4RR = A\_U\_0R * A\_0R\_4RR; \\ A\_U\_4LF = A\_U\_0R * A\_0R\_0L * A\_0L\_4LF; \\ A\_U\_4LR = A\_U\_0R * A\_0R\_0L * A\_0L\_4LR; \end{array}$ 

% Right Front [alpha\_1,alpha\_2,alpha\_3] = HT\_2\_RPY(A\_U\_4RF); A\_WRF\_4RF = roty(alpha\_2)\*rotz(alpha\_3)\*rotx(alpha\_1); % Wheel Universal frame A\_WRF\_SRF = roty(beta\_SRF\_ys(p)) \* rotz(beta\_SRF\_zs(p)); % Surface Frame A\_4RF\_WRF = A\_WRF\_4RF'; A\_4RF\_SRF = A\_4RF\_WRF \* A\_WRF\_SRF; R\_4RF\_SRF = A\_4RF\_SRF(1:3,1:3);

% Right Rear [alpha\_1,alpha\_2,alpha\_3] = HT\_2\_RPY(A\_U\_4RR); A\_WRR\_4RR = roty(alpha\_2)\*rotz(alpha\_3)\*rotx(alpha\_1); % Wheel Universal frame A\_WRR\_SRR = roty(beta\_SRR\_vs(p)) \* rotz(beta\_SRR\_zs(p)); % Surface Frame A\_4RR\_WRR = A\_WRR\_4RR'; A\_4RR\_SRR = A\_4RR\_WRR \* A\_WRR\_SRR; R\_4RR\_SRR = A\_4RR\_SRR(1:3,1:3);

% Left Front [alpha\_1,alpha\_2,alpha\_3] = HT\_2\_RPY(A\_U\_4LF); A\_WLF\_4LF = roty(alpha\_2)\*rotz(alpha\_3)\*rotx(alpha\_1); % Wheel Universal frame A\_WLF\_SLF = roty(beta\_SLF\_ys(p)) \* rotz(beta\_SLF\_zs(p)); % Surface Frame A\_4LF\_WLF = A\_WLF\_4LF'; A\_4LF\_SLF = A\_4LF\_WLF \* A\_WLF\_SLF; R\_4LF\_SLF = A\_4LF\_SLF(1:3,1:3); % Left Rear [alpha\_1,alpha\_2,alpha\_3] = HT\_2\_RPY(A\_U\_4LR); A\_WLR\_4LR = roty(alpha\_2)\*rotz(alpha\_3)\*rotx(alpha\_1); % Wheel Universal frame A\_WLR\_SLR = roty(beta\_SLR\_ys(p)) \* rotz(beta\_SLR\_zs(p)); % Surface Frame A\_4LR\_WLR = A\_WLR\_4LR'; A\_4LR\_SLR = A\_4LR\_WLR \* A\_WLR\_SLR; R\_4LR\_SLR = A\_4LR\_SLR(1:3,1:3); R4RF\_SRF = [R4RF\_SRF\_R\_4RF\_SRF]; P4PP\_SPR = [P4PP\_SPR\_R\_4RF\_SRF];

```
R4RR\_SRR = [R4RR\_SRR \quad R\_4RR\_SRR];

R4LF\_SLF = [R4LF\_SLF \quad R\_4LF\_SLF];

R4LR\_SLR = [R4LR\_SLR \quad R\_4LR\_SLR];
```

end

%

% compute the force and moment exerted on the base OoR [Force, Moment, Normal Forces, towc, fc, fc Moment, NF moment,... Tow CRight, Tow CFront, Tow CLeft, Tow CRear, ... f gravity, f inertial, f gravity Moment, f inertial Moment] = ... rne base9(dh dyn, [qm; qmd; qmdd], ... [q RF; qd RF; qdd RF], [q RR; qd RR; qdd RR], ... [q LF; qd LF; qdd LF], [q LR; qd LR; qdd LR], ... Flat surface, theta S zs, theta S ys,... R4RF SRF, R4RR SRR, R4LF SLF, R4LR SLR, Touches); % test for accuracy for i=1:nr for j=1:nc if abs(Force(i,j)) < TOLForce(i,j) = 0;end if abs(Moment(i,j)) < TOL Moment(i,j) = 0;end if abs(Normal\_Forces(i,j)) < TOL Normal\_Forces(i,j) = 0;end if abs(fc(i,j)) < TOLfc(i,j) = 0;end if abs(towc(i,j)) < TOLtowc(i,j) = 0;end if abs(fc Moment(i,j)) < TOL  $fc_Moment(i,j) = 0;$ 

if abs(NF\_moment(i,j)) < TOL NF moment(i,j) = 0; end if abs(Tow\_CRight(i,j)) < TOL  $Tow_CRight(i,j) = 0;$ end if abs(Tow CFront(i,j)) < TOL Tow CFront(i,j) = 0;end if abs(Tow CLeft(i,j)) < TOL Tow\_CLeft(i,j) = 0; end if abs(Tow\_CRear(i,j)) < TOL Tow\_CRear(i,j) = 0;end if abs(f\_gravity(i,j)) < TOL  $f_gravity(i,j) = 0;$ end if abs(f\_inertial(i,j)) < TOL  $f_{inertial}(i,j) = 0;$ end if abs(f\_gravity\_Moment(i,j)) < TOL f\_gravity\_Moment(i,j) = 0; end if abs(f\_inertial\_Moment(i,j)) < TOL f\_inertial\_Moment(i,j) = 0; end end end Momentt = Moment; Tow\_CRight = Tow\_CRight(2,:); Tow\_CFront = Tow\_CFront(3,:); Tow\_CLeft = Tow\_CLeft(2,:);

Tow\_CRear = Tow\_CRear(3,:);

end

# Rne\_base9.m

function [Force, Moment, Normal\_Forces, Towc, Fc, Fc\_Moment, NF\_moment,... Tow\_Critical\_Right, Tow\_Critical\_Front, Tow\_Critical\_Left, Tow\_Critical\_Rear,... F\_gravity, F\_inertial, F\_gravity\_Moment, F\_inertial\_Moment] = ... rne(dh\_dyn, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13) n = numrows(dh dyn); % number of links in one manipulator np = numcols(a2)% number of time samples Q0 = a1(1:3,:);Q0d = a1(4:6:);Q0dd = a1(7:9,:); $Q_RF = a2(1:4,:);$  $Qd_RF = a2(5:8,:);$  $Qdd_RF = a2(9:12,:);$  $Q_RR = a3(1:4,:);$  $Qd_RR = a3(5:8,:);$ Qdd RR = a3(9:12,:);Q LF = a4(1:4,:);Qd LF = a4(5:8,:);Qdd LF = a4(9:12,:);Q LR = a5(1:4,:); $Qd_LR = a5(5:8,:);$  $Qdd_LR = a5(9:12,:);$ Flat surface = a6; theta S zu = a7; theta S yu = a8; R Right side = [a9; a10];R Left side = [a11; a12];Touching = a13; %\_ Initial Conditions radius =  $dh_dyn(5,3)$ ; %g = 9.81; %acceleration gravity on earth surface (m/s^2). g = 3.63; %acceleration gravity on Mars surface (m/s^2). V = zeros(3, np);Vd = zeros(3, np);Vp = zeros(4, np);Vpd = zeros(4, np);

for p=1:np,

```
Vp(:,p) = rotz(Q0(2,p))*rotz(Q0(3,p))*rotz(Q0(1,p))*...
      [0; (radius*Qd RF(4,p) - radius*Qd LF(4,p))/2; 0; 1];
                 Vpd(:,p) = rotz(Q0(2,p))*rotz(Q0(3,p))*rotz(Q0(1,p))*...
      [0; (radius*Qdd_RF(4,p) - radius*Qdd_LF(4,p))/2; 0; 1];
  end
  for p=1:np,
    V(:,p) = [0; 0; 0] + Vp(1:3,p);
                 Vd(:,p) = [-g; 0; 0] + Vpd(1:3,p);
  end
%
    w = zeros(3,1);
 %
        wd = zeros(3,1);
  %
  m = dh dyn(:,6);
                         % column vector of links' masses
  mass = m(1) + 2*m(2) + 2*m(3) + 4*m(4) + 4*m(5); % System total mass
        rc = dh dyn(:,7:9)';
                                  % matrix of COM data; row per link
        Im = [];
        for j=1:n,
                 I = [dh_dyn(j,10) dh_dyn(j,13) dh_dyn(j,15); ...
                          dh_dyn(j,13) dh_dyn(j,11) dh_dyn(j,14); ...
                          dh_dyn(j,15) dh_dyn(j,14) dh_dyn(j,12)];
                 Im = [Im I];
        end
  Force_Moment_NOM = [];
  f_NOM = [];
  tow_NOM = [];
  A_NOM = [];
  R_NOM = [];
  pstar_NOM = [];
for NOM=1:4,
  if NOM == 1,
    Q = Q RF;
          \overline{Qd} = Qd RF;
          Qdd = Qdd RF;
    sign = 1;
  elseif NOM == 2,
    Q = Q RR;
          \overline{Qd} = Qd RR;
          Qdd = Qdd_RR;
    sign = 1;
```

elseif NOM == 3, Q = Q LF; $\overline{Qd} = Qd LF;$  $Qdd = Qdd_LF;$ sign = -1; elseif NOM == 4,  $Q = Q_L R;$  $\overline{Qd} = Qd LR;$  $Qdd = Qdd_LR;$ sign = -1; end f\_p = []; tow\_p = []; A\_p = [];  $R_p = [];$ pstar\_p = []; for p=1:np, q0 = Q0(:,p);q0d = Q0d(:,p);q0dd= Q0dd(:,p); q = Q(:,p); $\mathbf{q}\mathbf{d} = \mathbf{Q}\mathbf{d}(:,\mathbf{p});$ qdd = Qdd(:,p);v = V(:,p); vd = Vd(:,p); w = q0d;wd = q0dd; fm = []; towm = []; pstarm = []; Am = []; Rm = []; theta = q;  $d = dh_dyn(2:n,2);$ a = dh\_dyn(2:n,3);  $alpha = dh_dyn(2:n,4);$ 

#### %\_

 $\begin{array}{l} A\_U\_0R = roty(q0(2)) * rotz(q0(3)) * rotx(q0(1)); \\ R\_U\_0R = A\_U\_0R(1:3,1:3); \\ R\_0R\_0L = [1 \ 0 \ 0; 0 \ -1 \ 0; 0 \ 0 \ -1]; \end{array}$ 

A OR  $0L = [1 \ 0 \ 0 \ 0; 0 \ -1 \ 0 \ 0; 0 \ 0 \ -1 \ 0; 0 \ 0 \ 0 \ 1];$ if (NOM == 1 || NOM == 2) Rm = [Rm R U 0R];Am = [Am A U 0R];elseif (NOM ==  $3 \parallel NOM == 4$ )  $R \cup OL = R \cup OR * R OR OL;$  $A_U_0L = A_U_0R * A_0R 0L;$ Rm = [Rm R U 0L];Am = [Am A U 0L];end p U 0 = [0; 0; 0];pstarm = [pstarm p\_U\_0]; % qd=[q0d [0 0 0 0; ... 0 0 0 0; ... qd']]; qdd=[q0dd [0 0 0 0; ... 0000;... qdd']]; %\_ % theta 4 manipulated in contact point of wheel with ground theta(4) = -theta(1) - theta(3) + sign\*(theta S zu(NOM, p) - q0(3)); theta(3) = theta(3) + pi;for j=1:n-1, A = DHtransformation(theta(j), d(j), a(j), alpha(j));pstar = [a(j); d(j)\*sin(alpha(j)); d(j)\*cos(alpha(j))];Am = [Am A];R = A(1:3,1:3);Rm = [Rm R];pstarm = [pstarm pstar]; end %------% % the forward recursion % for j=1:n, R = Rm(:,3\*j-2:3\*j)';pstar = pstarm(:,j); I = Im(:,3\*j-2:3\*j);

 $wd = R^*(wd + qdd(:,j) + cross(w,qd(:,j))); \\ w = R^*(w + qd(:,j));$ 

vd = cross(wd,pstar) + cross(w, cross(w,pstar)) + R\*vd;

 $vd_c = cross(wd,rc(:,j)) + cross(w,cross(w,rc(:,j))) + vd;$ 

```
f = m(j)*vd_c;
tow = I*wd + cross(w,I*w);
fm = [fm f];
towm = [towm tow];
```

end

 $f_p = [f_p fm];$ tow\_p = [tow\_p towm];

$$\begin{split} R_p &= [R_p \ Rm]; \\ A_p &= [A_p \ Am]; \\ pstar_p &= [pstar_p \ pstarm]; \end{split}$$

end

pstar\_NOM = [pstar\_NOM; pstar\_p]; A\_NOM = [A\_NOM; A\_p]; R\_NOM = [R\_NOM; R\_p]; f\_NOM = [f\_NOM; f\_p]; tow\_NOM = [tow\_NOM; tow\_p];

end f\_NOM;

%\_\_\_\_\_Normal Forces \_\_\_\_\_

Fext\_SRF\_SRF = [];
Fext\_SRR\_SRR = [];
Fext\_SLF\_SLF = [];
Fext\_SLR\_SLR = [];

```
towc_sys = []; fc_sys = []; fc_Moment_sys= []; NF_moment_sys = [];
tow_CRight_sys = []; tow_CFront_sys = []; tow_CLeft_sys = []; tow_CRear_sys = [];
f_gravity_sys = []; f_inertial_sys = [];
f_gravity_Moment_sys = []; f_inertial_Moment_sys = [];
```

for p=1:np,

%meu\_s = 0.6; %meu\_k = 0.15; q0 = Q0(:,p); nc = 4; Touch = Touching(:,p); [Fn\_SRF, Fn\_SRR, Fn\_SLF, Fn\_SLR, towc, fc, fc\_Moment, NF\_moment,...
tow\_Critical\_Right, tow\_Critical\_Front, tow\_Critical\_Left, tow\_Critical\_Rear,...
f\_gravity, f\_inertial, f\_gravity\_Moment, f\_inertial\_Moment] =...
Inertial11(f\_NOM(:,5\*p-4:5\*p), tow\_NOM(:,5\*p-4:5\*p),...
R\_NOM(:,15\*p-14:15\*p), A\_NOM(:,20\*p-19:20\*p),...
pstarm, dh\_dyn, theta\_S\_zu(:,p), theta\_S\_yu(:,p),...
Touch, m, q0(2),q0(3));

$\Gamma_{\rm SKI} = 0,701100$ s $\Gamma_{\rm SKI}$ ,	/0
Ff_SRR = 0;%meu_s * Fn_SRR;	% Static frictional force
Ff_SLF = 0;%meu_s * Fn_SLF;	%
Ff_SLR = 0;%meu_s * Fn_SLR;	%
$%Ff_SRF = meu_k * Fn_SRF;$	%
%Ff_SRR = meu_k * Fn_SRR;	% Dynamic frictional force
$%$ Ff_SLF = meu_k * Fn_SLF;	%
%Ff_SLR = meu_k * Fn_SLR;	%

% Generalized ground input forces and moments

$$\label{eq:FSRF_SRF} \begin{split} &FSRF\_[Fn\_SRF -Ff\_SRF 0]'; &\% \\ &FSRR\_SRR=[Fn\_SRR -Ff\_SRR 0]'; &\% \\ &FSLF\_SLF=[Fn\_SLF -Ff\_SLF 0]'; &\% \\ &FSLR\_SLR=[Fn\_SLR -Ff\_SLR 0]'; &\% \end{split}$$

TSRF SRF = [0]0 0]'; %  $TSRR_SRR = [0]$ 0]'; % External resultant moment 0 %  $TSLF_SLF = [0]$ 0 0]';  $TSLR_SLR = [0]$ % 0 0]';

Fext\_SRF\_SRF = [Fext\_SRF\_SRF [FSRF\_SRF; TSRF\_SRF]]; Fext\_SRR\_SRR = [Fext\_SRR\_SRR [FSRR\_SRR; TSRR\_SRR]]; Fext\_SLF\_SLF = [Fext\_SLF\_SLF [FSLF\_SLF; TSLF\_SLF]]; Fext\_SLR\_SLR = [Fext\_SLR\_SLR [FSLR\_SLR; TSLR\_SLR]];

towc\_sys = [towc\_sys towc]; fc\_sys = [fc\_sys fc]; fc\_Moment\_sys = [fc\_Moment\_sys fc\_Moment]; NF\_moment\_sys = [NF\_moment\_sys NF\_moment];

tow\_CRight\_sys = [tow\_CRight\_sys tow\_Critical\_Right]; tow\_CFront\_sys = [tow\_CFront\_sys tow\_Critical\_Front]; tow\_CLeft\_sys = [tow\_CLeft\_sys tow\_Critical\_Left]; tow\_CRear\_sys = [tow\_CRear\_sys tow\_Critical\_Rear];

f\_gravity\_sys = [f\_gravity\_sys f\_gravity]; f\_inertial\_sys = [f\_inertial\_sys f\_inertial]; f\_gravity\_Moment\_sys = [f\_gravity\_Moment\_sys f\_gravity\_Moment]; f\_inertial\_Moment\_sys = [f\_inertial\_Moment\_sys f\_inertial\_Moment]; end

Towc = towc\_sys; Fc = fc\_sys; Fc\_Moment = fc\_Moment\_sys; NF\_moment = NF\_moment\_sys;

```
Tow_Critical_Right = tow_CRight_sys;
Tow_Critical_Front = tow_CFront_sys;
Tow_Critical_Left = tow_CLeft_sys;
Tow_Critical_Rear = tow_CRear_sys;
```

```
F_gravity = f_gravity_sys;
F_inertial = f_inertial_sys;
F_gravity_Moment = f_gravity_Moment_sys;
F_inertial_Moment = f_inertial_Moment_sys;
```

%\_

```
Force Moment nps = [];
for s = 1:2
                   % Right/Left side
  if s == 1,
    pstar_s = pstar_NOM(1:6,:);
    \hat{R}_{s} = \hat{R}_{NOM}(1:6,:);
    f_s = f_NOM(1:6,:);
    tow_s = tow_NOM(1:6,:);
    Fext_s = [Fext_SRF_SRF; Fext_SRR_SRR];
    R_s_Surface = R_Right_side;
  elseif s == 2,
    pstar_s = pstar_NOM(7:12,:);
    R_s = R_NOM(7:12,:);
    f_s = f_NOM(7:12,:);
    tow_s = tow_NOM(7:12,:);
    Fext_s = [Fext_SLF_SLF; Fext_SLR_SLR];
    R_s_Surface = R_Left_side;
  end
  Force_Moment_NOM = [];
  for NOM = 1:2,
                      % Front/Rear Leg
    if NOM == 1,
      pstar = pstar_s(1:3,:);
      Rot = R s(1:3,:);
      f = f_s(1:3,:);
      tow = tow_s(1:3,:);
      Fext = Fext_s(1:6,:);
```

```
R Surface = R s Surface(1:3,:);
elseif NOM ==2,
  pstar = pstar s(4:6,:);
  Rot = R s(4:6,:);
  f = f s(4:6,:);
  tow = tow_s(4:6,:);
  Fext = Fext s(7:12,:);
  R Surface = R s Surface(4:6,:);
end
Force Moment np = [];
for p=1:np,
  R_4_S = R_Surface(:,3*p-2:3*p);
  rm = pstar(:,5*p-4:5*p);
  Rm = Rot(:,15*p-14:15*p);
  fm = f(:,5*p-4:5*p);
  towm = tow(:,5*p-4:5*p);
                              % force/moments at end of end-effector
  F = Fext(1:3,p);
            T = Fext(4:6,p);
  Moment n = [];
  Force_n =[];
  for j =n:-1:3
    r = rm(:,j);
    if j == n,
       R = R_4_S;
    else
                                 R = Rm(:,3*j+1:3*j+3);
    end
    if (j == 5) \parallel (j == 4) \parallel ((j == 3)\&\&(NOM == 1))
       T = R^{*}(T + cross(R'^{*}r,F)) + cross(r + rc(:,j),fm(:,j)) + towm(:,j);
       F = R*F + fm(:,j);
    elseif (j == 3)&&(NOM == 2),
       T = R^{*}(T + cross(R'^{*}r,F));
       F = R * F;
    end
    Moment_n = [Moment_n T];
```

end

```
Force_Moment_np = [Force_Moment_np [Force_n; Moment_n]];
end
```

Force\_Moment\_NOM = [Force\_Moment\_NOM; Force\_Moment\_np];

end

%\_\_\_Conjunctional joint\_\_\_\_ for p=1:np, Force\_Moment\_NOM(1:6,3\*p) = Force\_Moment\_NOM(1:6,3\*p) + Force\_Moment\_NOM(7:12,3\*p); end

%Force Moment NOM

Force\_Moment\_NOM\_s = Force\_Moment\_NOM(1:6,:);

%\_

```
pstar = pstar s(1:3,:);
Rot = R s(1:3,:);
f = f s(1:3,:);
tow = tow s(1:3,:);
Moment ns = [];
Force_ns = [];
for p = 1:np,
  Rm = Rot(:,15*p-14:15*p);
  rm = pstar(:,5*p-4:5*p);
  fm = f(:,5*p-4:5*p);
  towm = tow(:,5*p-4:5*p);
  F = Force_Moment_NOM_s(1:3,3*p);
  T = Force_Moment_NOM_s(4:6,3*p);
  j = 2;
  r = rm(:,j);
  R = Rm(:,3*j+1:3*j+3);
  T = R^{*}(T + cross(R'^{*}r,F)) + cross(r + rc(:,j),fm(:,j)) + towm(:,j);
  \mathbf{F} = \mathbf{R} * \mathbf{F} + \mathbf{fm}(:,\mathbf{j});
  Moment ns = [Moment ns T];
  Force ns = [Force ns F];
```

end

Force\_Moment\_nps = [Force\_Moment\_nps; [Force\_ns; Moment\_ns]];

end

F1R 1R = Force Moment nps(1:3,:);T1R 1R = Force Moment nps(4:6,:);F1L 1L = Force Moment nps(7:9,:); $T1L_1L = Force_Moment_nps(10:12,:);$ fOR OR = f NOM(1:3,:);tow0R 0R = tow NOM(1:3,:);ROR  $0L = [1 \ 0 \ 0; 0 \ -1 \ 0; 0 \ 0 \ -1];$ Rot R = R NOM(1:3,:); Rot\_L =  $R_NOM(7:9,:);$ pstar\_R = pstar\_NOM(1:3,:); Moment\_ns = []; Force\_ns = []; for p = 1: np, Rm R = Rot R(:,15\*p-14:15\*p);  $Rm_L = Rot_L(:,15*p-14:15*p);$  $rm = pstar_R(:,5*p-4:5*p);$ fm =  $f0R_0R(:,5*p-4:5*p);$  $towm = tow0R_0R(:,5*p-4:5*p);$  $TR = T1R_1R(:,p);$  $FR = F1R_1R(:,p);$  $TL = T1L_1L(:,p);$  $FL = F1L_1L(:,p);$ j=1; r = rm(:,j); $R_R = Rm_R(:,3*j+1:3*j+3);$  $R_L = Rm_L(:,3*j+1:3*j+3);$  $T = R_R^*(TR + cross(R_R^*, FR)) + cross(r+rc(:,j), fm(:,j)) + towm(:,j) + \dots$  $R0R_0L^*R_L^*(TL + cross(R_L'^*r,FL));$  $F = R \overline{R} * FR + fm(:,j) + R0R \overline{0L} * R L*FL;$ Moment ns = [Moment ns T];Force ns = [Force ns F]; end

Force\_Moment\_np0 = [Force\_ns; Moment\_ns];

Normal\_Forces = [Fext\_SRF\_SRF(1,:); Fext\_SRR\_SRR(1,:);... Fext\_SLF\_SLF(1,:); Fext\_SLR\_SLR(1,:)];

## Inertial11

function [FnSRF, FnSRR, FnSLF, FnSLR, towc, fc, fc\_Moment, Normal\_Forces\_moments,... tow\_Critical\_Right, tow\_Critical\_Front, tow\_Critical\_Left, tow\_Critical\_Rear,... f\_gravity, f\_inertial, f\_gravity\_Moment, f\_inertial\_Moment] =... Inertial11(f, tow, R, A, pstarm, dh\_dyn, theta\_S\_zu, theta\_S\_yu, Touch, m, Roll, Pitch)

touchRF = Touch(1); touchRR = Touch(2); touchLF = Touch(3); touchLR = Touch(4);

m0 = m(1); m1 = m(2); m2 = m(3); m3 = m(4); m4 = m(5);Mass = m0 + 2\*(m1 + m2) + 4\*(m3 + m4);

TOL = 0.00001; % tolerance value

g = 3.63; meu=0;

fOR = f(1:3,1); f1R = f(1:3,2); f2R = f(1:3,3); f3RF = f(1:3,4); f4RF = f(1:3,5); f3RR = f(4:6,4);f4RR = f(4:6,5);

f0L = f(7:9,1); f1L = f(7:9,2); f2L = f(7:9,3);f3LF = f(7:9,4); f4LF = f(7:9,5);f3LR = f(10:12,4); f4LR = f(10:12,5);

tow0R = tow(1:3,1); tow1R = tow(1:3,2); tow2R = tow(1:3,3); tow3RF = tow(1:3,4); tow4RF = tow(1:3,5); tow3RR = tow(4:6,4); tow4RR = tow(4:6,5);

tow0L = tow(7:9,1); tow1L = tow(7:9,2); tow2L = tow(7:9,3);

tow3LF = tow(7:9,4); tow4LF = tow(7:9,5);tow3LR = tow(10:12,4); tow4LR = tow(10:12,5);tow4RF; tow4RR; tow4LF; tow4LR; r0 = pstarm(:, 1);r1 = pstarm(:, 2);r2 = pstarm(:, 3);r3 = pstarm(:, 4);r4 = pstarm(:, 5); $rc = dh_dyn(:,7:9)';$ rc0 = rc(:,1);rc1 = rc(:,2);rc2 = rc(:,3);rc3 = rc(:,4);rc4 = rc(:,5);AU 4RF = A(1:4,1:4) \* A(1:4,5:8) \* A(1:4,9:12) \* A(1:4,13:16) \* A(1:4,17:20);AU 3RF = A(1:4,1:4) \* A(1:4,5:8) \* A(1:4,9:12) \* A(1:4,13:16); $AU_2R = A(1:4,1:4) * A(1:4,5:8) * A(1:4,9:12);$  $AU_1R = A(1:4,1:4) * A(1:4,5:8);$  $AU_0R = A(1:4,1:4);$  $rU_4RF = AU_4RF(1:3,4);$  $rU_3RF = AU_3RF(1:3,4);$  $rU_2R = AU_2R(1:3,4);$  $rU_1R = AU_1R(1:3,4);$  $rU_0R = AU_0R(1:3,4);$  $rcU_4RF = rU_3RF; \% - AU_4RF(1:3,1:3)*rc4;$  $rcU_3RF = rU_2R - AU_3RF(1:3,1:3)*rc3;$  $rcU_2R = rU_1R - AU_2R(1:3,1:3)*rc2;$  $rcU_1R = rU_0R - AU_1R(1:3,1:3)*rc1;$  $rcU_0R = r0;$ AU\_4RR = A(5:8,1:4) \* A(5:8,5:8) \* A(5:8,9:12) \* A(5:8,13:16) \*A(5:8,17:20);  $AU^{-}3RR = A(5:8,1:4) * A(5:8,5:8) * A(5:8,9:12) * A(5:8,13:16);$ rU 4RR = AU 4RR(1:3,4); $rU_3RR = AU_3RR(1:3,4);$ rcU 4RR = rU 3RR;% - AU 4RR(1:3,1:3)\*rc4; $rcU_3RR = rU_2R - AU_3RR(1:3,1:3)*rc3;$ 

269
AU 4LF = A(9:12,1:4) \* A(9:12,5:8) \* A(9:12,9:12) \* A(9:12,13:16) \* A(9:12,17:20);AU 3LF = A(9:12,1:4) \* A(9:12,5:8) \* A(9:12,9:12) \* A(9:12,13:16);AU 2L = A(9:12,1:4) \* A(9:12,5:8) \* A(9:12,9:12);AU 1L = A(9:12,1:4) \* A(9:12,5:8);AU 0L = A(9:12,1:4);rU 4LF = AU 4LF(1:3.4);rU 3LF = AU 3LF(1:3,4);rU 2L = AU 2L(1:3,4);rU 1L = AU 1L(1:3,4); $rU \ 0L = AU \ 0L(1:3,4);$ rcU 4LF = rU 3LF;% - AU 4LF(1:3,1:3)\*rc4; rcU 3LF = rU 2L - AU 3LF(1:3,1:3)\*rc3;rcU 2L = rU 1L - AU 2L(1:3,1:3)\*rc2;rcU 1L = rU 0R - AU 1L(1:3,1:3)\*rc1;AU 4LR = A(13:16,1:4) \* A(13:16,5:8) \* A(13:16,9:12) \* A(13:16,13:16) \* A(13:16,17:20);AU 3LR = A(13:16,1:4) \* A(13:16,5:8) \* A(13:16,9:12) \* A(13:16,13:16);rU 4LR = AU 4LR(1:3,4);rU 3LR = AU 3LR(1:3,4);rcU 4LR = rU 3LR; % - AU 4LR(1:3,1:3)\*rc4;rcU 3LR = rU 2L - AU 3LR(1:3,1:3)\*rc3;%\_ % Right Front  $beta_SRF_zu = theta_S_zu(1);$  $beta_SRF_yu = theta_S_yu(1);$ [alpha\_1,alpha\_2,alpha\_3] = HT\_2\_RPY(AU\_4RF); AWRF\_4RF = roty(alpha\_2)\*rotz(alpha\_3)\*rotx(alpha\_1); % Wheel Universal frame AWRF\_SRF = roty(beta\_SRF\_yu) \* rotz(beta\_SRF\_zu); % Surface Frame A4RF WRF = AWRF 4RF'; A4RF SRF = A4RF WRF \* AWRF SRF; AU SRF = AU 4RF \* A4RF SRF; % Right Rear beta\_SRR\_zu = theta\_S\_zu(2); beta\_SRR\_yu = theta\_S\_yu(2); [alpha\_1,alpha\_2,alpha\_3] = HT\_2\_RPY(AU\_4RR); AWRR\_4RR = roty(alpha\_2)\*rotz(alpha\_3)\*rotx(alpha\_1); % Wheel Universal frame AWRR SRR = roty(beta SRR yu) \* rotz(beta SRR zu); % Surface Frame A4RR WRR = AWRR 4RR'; A4RR SRR = A4RR WRR \* AWRR SRR; AU SRR = AU 4RR \* A4RR SRR; % Left Front beta SLF zu =theta S zu(3); beta\_SLF\_yu = theta\_S\_yu(3); [alpha\_1,alpha\_2,alpha\_3] = HT\_2\_RPY(AU\_4LF); AWLF\_4LF = roty(alpha\_2)\*rotz(alpha\_3)\*rotx(alpha\_1); % Wheel Universal frame

270

```
AWLF SLF = roty(beta SLF yu) * rotz(beta SLF zu);
                                                                                                                                                                                % Surface Frame
      A4LF WLF = AWLF 4LF';
       A4LF SLF = A4LF WLF * AWLF SLF;
      AU SLF = AU 4LF * A4LF SLF;
% Left Rear
       beta SLR zu = theta S zu(4);
       beta SLR yu = theta S yu(4);
      [alpha_1, alpha_2, alpha_3] = HT_2 RPY(AU 4LR);
      AWLR 4LR = roty(alpha 2)*rotz(alpha 3)*rotx(alpha 1); % Wheel Universal frame
      AWLR SLR = roty(beta SLR yu) * rotz(beta SLR zu);
                                                                                                                                                                                   % Surface Frame
      A4LR WLR = AWLR 4LR';
       A4LR SLR = A4LR WLR * AWLR SLR;
      AU SLR = AU 4LR * A4LR SLR;
%_
inv(AU 0R(1:3,1:3))*AU SRF(1:3,1:3);
inv(AU 0R(1:3,1:3))*AU SRR(1:3,1:3);
inv(AU 0R(1:3,1:3))*AU SLF(1:3,1:3);
inv(AU_0R(1:3,1:3))*AU_SLR(1:3,1:3);
AU SRF(1:3,1:3);
AU SRR(1:3,1:3);
AU SLF(1:3,1:3);
AU SLR(1:3,1:3);
H1 = AU_SRF(1:3,1:3)*[1 - meu 0]';
H2 = AU_SRR(1:3,1:3)*[1 -meu 0]';
H3 = AU_SLF(1:3,1:3)*[1 - meu 0]';
H4 = AU_SLR(1:3,1:3)*[1 -meu 0]';
System Force 0R = inv(AU \ 0R(1:3,1:3))*...
        (AU_4RF(1:3,1:3)*f4RR + AU_3RF(1:3,1:3)*f3RF + AU_4RR(1:3,1:3)*f4RR +
AU 3RR(1:3,1:3)*f3RR + ...
         AU 2R(1:3,1:3)*f2R + AU 1R(1:3,1:3)*f1R + AU 4LF(1:3,1:3)*f4LF + AU 3LF(1:3,1:3)*f3LF + ...
         AU_{4LR(1:3,1:3)}*f4LR + AU_{3LR(1:3,1:3)}*f3LR + AU_{2L(1:3,1:3)}*f2L + AU_{1L(1:3,1:3)}*f1L + \dots + AU_{2L(1:3,1:3)}*f2L + AU_{2L(1:3,1
         AU_0R(1:3,1:3)*f0R);
System_Force_U = AU_4RF(1:3,1:3)*f4RR + AU_3RF(1:3,1:3)*f3RF + AU_4RR(1:3,1:3)*f4RR + AU_
AU_3RR(1:3,1:3)*f3RR + ...
         AU_2R(1:3,1:3)*f2R + AU_1R(1:3,1:3)*f1R + AU_4LF(1:3,1:3)*f4LF + AU_3LF(1:3,1:3)*f3LF + ...
         AU 4LR(1:3,1:3)*f4LR + AU 3LR(1:3,1:3)*f3LR + AU 2L(1:3,1:3)*f2L + AU 1L(1:3,1:3)*f1L + ...
         AU 0R(1:3,1:3)*f0R;
   A1 = touchRF * cross(rU 4RF, AU SRF(1:3,1:3)*[1 - meu 0]');
   A2 = touchRR * cross(rU_4RR, AU_SRR(1:3,1:3)*[1 -meu 0]');
   A3 = touchLF * cross(rU_4LF, AU_SLF(1:3,1:3)*[1 -meu 0]');
   A4 = touchLR * cross(rU 4LR, AU SLR(1:3,1:3)*[1 - meu 0]');
      M U = cross(rcU 4RF, AU 4RF(1:3,1:3)*f4RF) + cross(rcU 3RF, AU 3RF(1:3,1:3)*f3RF)+...
```

cross(rcU\_4RR, AU\_4RR(1:3,1:3)\*f4RR) + cross(rcU\_3RR, AU\_3RR(1:3,1:3)\*f3RR)+...

cross(rcU 2R, AU 2R(1:3,1:3)\*f2R) + cross(rcU 1R, AU 1R(1:3,1:3)\*f1R)+... cross(rcU 0R, AU 0R(1:3,1:3)\*f0R) + cross(rcU 1L, AU 1L(1:3,1:3)\*f1L)+... cross(rcU 2L, AU 2L(1:3,1:3)\*f2L) + cross(rcU 3LF, AU 3LF(1:3,1:3)\*f3LF)+... cross(rcU 4LF, AU 4LF(1:3,1:3)\*f4LF) + cross(rcU 3LR, AU 3LR(1:3,1:3)\*f3LR)+... cross(rcU 4LR, AU 4LR(1:3,1:3)\*f4LR) + ... AU 4RF(1:3,1:3)\*tow4RF + AU 3RF(1:3,1:3)\*tow3RF + AU 4RR(1:3,1:3)\*tow4RR + AU 3RR(1:3,1:3)\*tow3RR+... AU 4LF(1:3,1:3)\*tow4LF + AU\_3LF(1:3,1:3)\*tow3LF + AU\_4LR(1:3,1:3)\*tow4LR + AU 3LR(1:3,1:3)\*tow3LR+.. AU 2R(1:3,1:3)\*tow2R + AU 1R(1:3,1:3)\*tow1R + AU 2L(1:3,1:3)\*tow2L + AU 1L(1:3,1:3)\*tow1L +... AU 0R(1:3,1:3)\*tow0R; rU 4RR - rU 4RF; rU 4LF - rU 4RF; rU 4LR - rU 4RF; B1 = touchRR \* cross(rU 4RR - rU 4RF, AU SRR(1:3,1:3)\*[1 - meu 0]');B2 = touchLF \* cross(rU 4LF - rU 4RF, AU SLF(1:3,1:3)\*[1 - meu 0]');B3 = touchLR \* cross( $rU_4LR - rU_4RF$ , AU\_SLR(1:3,1:3)\*[1 -meu 0]'); M1 = cross(rcU 4RF - rU 4RF, AU 4RF(1:3,1:3)\*f4RF) + cross(rcU 3RF - rU 4RF,AU 3RF(1:3,1:3)\*f3RF)+... cross(rcU 4RR - rU 4RF, AU 4RR(1:3,1:3)\*f4RR) + cross(rcU 3RR - rU 4RF, AU 3RR(1:3,1:3)\*f3RR)+... cross(rcU 2R - rU 4RF, AU 2R(1:3,1:3)\*f2R) + cross(rcU 1R - rU 4RF, AU 1R(1:3,1:3)\*f1R)+...  $cross(rcU_0R - rU_4RF, AU_0R(1:3,1:3)*f0R) + cross(rcU_1L - rU_4RF,$ AU 1L(1:3,1:3)\*f1L)+...  $cross(rcU_2L - rU_4RF, AU_2L(1:3,1:3)*f2L) + cross(rcU_3LF - rU_4RF, AU_2L(1:3,1:3)*f2L) + cross(rcU_3LF - rU_4RF, AU_4RF, A$ AU 3LF(1:3,1:3)\*f3LF)+... cross(rcU\_4LF - rU\_4RF, AU\_4LF(1:3,1:3)\*f4LF) + cross(rcU\_3LR - rU\_4RF, AU 3LR(1:3,1:3)\*f3LR)+... cross(rcU 4LR - rU 4RF, AU 4LR(1:3,1:3)\*f4LR) + ... AU 4RF(1:3,1:3)\*tow4RF + AU 3RF(1:3,1:3)\*tow3RF + AU 4RR(1:3,1:3)\*tow4RR + AU 3RR(1:3,1:3)\*tow3RR+... AU 4LF(1:3,1:3)\*tow4LF + AU 3LF(1:3,1:3)\*tow3LF + AU 4LR(1:3,1:3)\*tow4LR + AU\_3LR(1:3,1:3)\*tow3LR+... AU\_2R(1:3,1:3)\*tow2R + AU\_1R(1:3,1:3)\*tow1R + AU\_2L(1:3,1:3)\*tow2L + AU\_1L(1:3,1:3)\*tow1L +... AU\_0R(1:3,1:3)\*tow0R; C1 = touchRF \* cross(rU 4RF - rU 4RR, AU SRF(1:3,1:3)\*[1 - meu 0]');C2 = touchLF \* cross(rU 4LF - rU 4RR, AU SLF(1:3,1:3)\*[1 - meu 0]');C3 = touchLR \* cross(rU 4LR - rU 4RR, AU SLR(1:3,1:3)\*[1 - meu 0]');M2 = cross(rcU 4RR - rU 4RR, AU 4RR(1:3,1:3)\*f4RR) + cross(rcU 3RR - rU 4RR,AU 3RR(1:3,1:3)\*f3RR)+... cross(rcU 4RF - rU 4RR, AU 4RF(1:3,1:3)\*f4RF) + cross(rcU 3RF - rU 4RR, AU 3RR(1:3,1:3)\*f3RR)+... cross(rcU 2R - rU 4RR, AU 2R(1:3,1:3)\*f2R) + cross(rcU 1R - rU 4RR, AU 1R(1:3,1:3)\*f1R)+...

272

 $cross(rcU \ 0R \ -rU \ 4RR, AU \ 0R(1:3,1:3)*f0R) \ + cross(rcU \ 1L \ -rU \ 4RR, AU \ 4RR$ AU 1L(1:3,1:3)\*f1L)+... cross(rcU 2L - rU 4RR, AU 2L(1:3,1:3)\*f2L) + cross(rcU 3LF - rU 4RR, AU 3LF(1:3,1:3)\*f3LF)+... cross(rcU 4LF - rU 4RR, AU 4LF(1:3,1:3)\*f4LF) + cross(rcU 3LR - rU 4RR, AU 3LR(1:3,1:3)\*f3LR)+... cross(rcU 4LR - rU 4RR, AU 4LR(1:3,1:3)\*f4LR) +... AU 4RF(1:3,1:3)\*tow4RF + AU 3RF(1:3,1:3)\*tow3RF + AU 4RR(1:3,1:3)\*tow4RR + AU 3RR(1:3,1:3)\*tow3RR+... AU 4LF(1:3,1:3)\*tow4LF + AU 3LF(1:3,1:3)\*tow3LF + AU 4LR(1:3,1:3)\*tow4LR + AU 3LR(1:3,1:3)\*tow3LR+... AU 2R(1:3,1:3)\*tow2R + AU 1R(1:3,1:3)\*tow1R + AU 2L(1:3,1:3)\*tow2L + AU 2R(1:3,1:3)\*tow2R + AU 1R(1:3,1:3)\*tow1R + AU 2L(1:3,1:3)\*tow2L + AU 1R(1:3,1:3)\*tow1R + AU 1R(1:3,1:3)\*tow2L + AU 1R(1:3,1:3)\*tow1R + AU 1R(1:3,1:3)\*tow1RAU 1L(1:3,1:3)\*tow1L +... AU 0R(1:3,1:3)\*tow0R; D1 = touchRF \* cross(rU 4RF - rU 4LF, AU SRF(1:3,1:3)\*[1 - meu 0]');D2 = touchRR \* cross(rU 4RR - rU 4LF, AU SRR(1:3,1:3)\*[1 - meu 0]');D3 = touchLR \* cross(rU 4LR - rU 4LF, AU SLR(1:3,1:3)\*[1 - meu 0]');M3 = cross(rcU 4RR - rU 4LF, AU 4RR(1:3,1:3)\*f4RR) + cross(rcU 3RR - rU 4LF,AU 3RR(1:3,1:3)\*f3RR)+... cross(rcU 4RF - rU 4LF, AU 4RF(1:3,1:3)\*f4RF) + cross(rcU 3RF - rU 4LF, AU 3RR(1:3,1:3)\*f3RR)+... cross(rcU 2R - rU 4LF, AU 2R(1:3,1:3)\*f2R) + cross(rcU 1R - rU 4LF, AU 1R(1:3,1:3)\*f1R)+... cross(rcU 0R - rU 4LF, AU 0R(1:3,1:3)\*f0R) + cross(rcU 1L - rU 4LF, AU 1L(1:3,1:3)\*f1L)+...  $cross(rcU_2L - rU_4LF, AU_2L(1:3,1:3)*f2L) + cross(rcU_3LF - rU_4LF, AU_2L(1:3,1:3)*f2L) + cross(rcU_3LF - rU_4LF, AU_4LF, A$ AU 3LF(1:3,1:3)\*f3LF)+... cross(rcU\_4LF - rU\_4LF, AU\_4LF(1:3,1:3)\*f4LF) + cross(rcU\_3LR - rU\_4LF, AU 3LR(1:3,1:3)\*f3LR)+... cross(rcU 4LR - rU 4LF, AU 4LR(1:3,1:3)\*f4LR) + ... AU 4RF(1:3,1:3)\*tow4RF + AU 3RF(1:3,1:3)\*tow3RF + AU 4RR(1:3,1:3)\*tow4RR + AU 3RR(1:3,1:3)\*tow3RR+ ... AU 4LF(1:3,1:3)\*tow4LF + AU 3LF(1:3,1:3)\*tow3LF + AU 4LR(1:3,1:3)\*tow4LR + AU 3LR(1:3,1:3)\*tow3LR+... AU 2R(1:3,1:3)\*tow2R + AU 1R(1:3,1:3)\*tow1R + AU 2L(1:3,1:3)\*tow2L + AU\_1L(1:3,1:3)\*tow1L +... AU\_0R(1:3,1:3)\*tow0R;  $E1 = touchRF * cross(rU_4RF - rU_4LR, AU_SRF(1:3,1:3)*[1 - meu 0]');$  $E2 = touchRR * cross(rU_4RR - rU_4LR, AU_SRR(1:3,1:3)*[1 - meu 0]');$ E3 = touchLF \* cross(rU 4LF - rU 4LR, AU SLF(1:3,1:3)\*[1 - meu 0]');M4 = cross(rcU 4RR - rU 4LR, AU 4RR(1:3,1:3)\*f4RR) + cross(rcU 3RR - rU 4LR, AU 4RR(1:3,1:3)\*f4RR) + cross(rcU 3RR) + crosR) + cross(rCU 3RR) + cross(rCU 3RR) + crosR)AU 3RR(1:3,1:3)\*f3RR)+... cross(rcU 4RF - rU 4LR, AU 4RF(1:3,1:3)\*f4RF) + cross(rcU 3RF - rU 4LR, AU 3RR(1:3,1:3)\*f3RR)+... cross(rcU 2R - rU 4LR, AU 2R(1:3,1:3)\*f2R) + cross(rcU 1R - rU 4LR, AU 1R(1:3,1:3)\*f1R)+... cross(rcU 0R - rU 4LR, AU 0R(1:3,1:3)\*f0R) + cross(rcU 1L - rU 4LR, AU 1L(1:3,1:3)\*f1L)+...

 $\begin{array}{l} cross(rcU_2L - rU_4LR, AU_2L(1:3,1:3)*f2L) + cross(rcU_3LF - rU_4LR, \\ AU_3LF(1:3,1:3)*f3LF)+... \\ cross(rcU_4LF - rU_4LR, AU_4LF(1:3,1:3)*f4LF) + cross(rcU_3LR - rU_4LR, \\ AU_3LR(1:3,1:3)*f3LR)+... \\ cross(rcU_4LR - rU_4LR, AU_4LR(1:3,1:3)*f4LR) + ... \\ AU_4RF(1:3,1:3)*tow4RF + AU_3RF(1:3,1:3)*tow3RF + AU_4RR(1:3,1:3)*tow4RR + \\ AU_3RR(1:3,1:3)*tow3RR+... \\ AU_4LF(1:3,1:3)*tow4LF + AU_3LF(1:3,1:3)*tow3LF + AU_4LR(1:3,1:3)*tow4LR + \\ AU_3LR(1:3,1:3)*tow3LR+... \\ AU_2R(1:3,1:3)*tow3LR+... \\ AU_2R(1:3,1:3)*tow3LR + ... \\ AU_0R(1:3,1:3)*tow1L + ... \\ AU_0R(1:3,1:3)*tow0R; \end{array}$ 

#### %

# % Right Legs & Left Legs are in contact with ground %

if (touchRF == 1 && touchRR == 1)&& (touchLF == 1 && touchLR == 1)if (Roll  $\sim = 0$  && Pitch == 0) Fn SRF = -M3(2)/(D1(2)+D2(2));Fn SRR = Fn SRF; Fn SLF = -M1(2)/(B2(2)+B3(2));Fn SLR = Fn SLF;  $xxx = Fn\_SRF + Fn\_SRR + Fn\_SLF + Fn\_SLR;$ elseif (Roll == 0 && Pitch  $\sim$ = 0) || (Roll == 0 && Pitch == 0)  $Fn_SRF = -M2(3)/(C1(3)+C2(3));$ Fn SRR = -M1(3)/(B1(3)+B3(3)); $Fn\_SLF = Fn\_SRF;$ Fn SLR = Fn SRR;  $xxx = Fn\_SRF + Fn\_SRR + Fn\_SLF + Fn\_SLR;$ end end % % Right Legs in contact with ground & Left Legs without contact % if (touchRF == 1 && touchRR == 1) && (touchLF == 0 && touchLR == 0)Coefficient =  $\begin{bmatrix} 0 & B1(3); \dots \end{bmatrix}$ C1(3) 0 ]; b = [-M1(3) - M2(3)]';x = inv(Coefficient)\*b;Fn SRF = x(1);

```
Fn\_SRR = x(2);

Fn\_SLF = 0;

Fn\_SLR = 0;
```

end

```
%
% Left Legs in contact with ground & Right Legs without contact
%
if (touchRF == 0 \&\& touchRR == 0) \&\& (touchLF == 1 \&\& touchLR == 1)
    Coefficient = \begin{bmatrix} 0 & D3(3); \dots \end{bmatrix}
              E3(3) 0 ];
    b = [-M3(3) - M4(3)]';
    x = inv(Coefficient)*b;
    Fn SRF = 0;
    Fn SRR = 0;
    Fn SLF = x(1);
    Fn SLR = x(2);
end
%
% Right Legs in contact with ground & Either Left Front or Rear Leg without contact
%
if (touchRF == 1 && touchRR == 1) &&...
    ((\text{touchLF} == 1 \&\& \text{touchLR} == 0) || (\text{touchLF} == 0 \&\& \text{touchLR} == 1))
    Coefficient = [0]
                        B1(2) B2(2) B3(2);...
              0 B1(3) B2(3) B3(3);...
              C1(3) 0 C2(3) C3(3);...
              H1(1) H2(1) H3(1) H4(1)];
    b = [-M1(2) - M1(3) - M2(3) - System_Force_U(1)]';
    x = inv(Coefficient)*b;
    Fn_SRF = x(1);
    Fn\_SRR = x(2);
    Fn_SLF = x(3);
    Fn\_SLR = x(4);
%
% Either Right Front or Rear Leg in contact with ground & Left Legs with contact
%
elseif ((touchRF == 0 && touchRR == 1)||(touchRF == 1 && touchRR == 0))...
  && (touchLF == 1 && touchLR == 1)
```

 $b = [-M3(2) - M3(3) - M4(3) - System_Force_U(1)]';$ 

x = inv(Coefficient)\*b;Fn\_SRF = x(1); Fn\_SRR = x(2); Fn\_SLF = x(3); Fn\_SLR = x(4);

```
end
```

```
%
% tolerance
%
if abs(Fn SRF) < TOL
 Fn SRF = 0;
end
if abs(Fn_SRR) < TOL
 Fn SRR = 0;
end
if abs(Fn SLF) < TOL
 Fn SLF = 0;
end
if abs(Fn SLR) < TOL
 Fn SLR = 0;
end
%_
% Constraints: poistive Normal forces
% touching point
if (Fn_SRR <= 0 && Fn_SLR <= 0)
  touchRR = 0;
  touchLR = 0;
  Fn_SRF = -M3(2)/D1(2);
  Fn_SLF = -M1(2)/B2(2);
elseif (Fn_SRF <= 0 && Fn_SLF <= 0)
  touchRF = 0;
  touchLF = 0;
  Fn_SRR = -M4(2)/E2(2);
  Fn_SLR = -M2(2)/C3(2);
elseif (Fn_SRF <= 0 && Fn_SRR <= 0)
  touchR\overline{F} = 0;
  touchRR = 0;
  Fn_SLF = -M4(3)/E3(3);
  Fn SLR = -M3(3)/D3(3);
elseif (Fn SLF <= 0 && Fn SLR <= 0)
  touchL\overline{F} = 0;
  touchLR = 0;
  Fn_SRF = -M2(3)/C1(3);
  Fn\_SRR = -M1(3)/B1(3);
end
```

if Fn SRF <= 0 touchRF = 0; Fn SRF = 0; end if Fn SRR <= 0 touchRR = 0; Fn SRR = 0; end if Fn SLF  $\leq 0$ touchLF = 0; Fn SLF = 0; end if Fn SLR  $\leq 0$ touchLR = 0; Fn SLR = 0; end FnSRF = Fn SRF;FnSRR = Fn SRR;FnSLF = Fn SLF;FnSLR = Fn SLR;xxx = FnSRF + FnSRR + FnSLF + FnSLR;% rU CM = (rcU 0R\*m0 + rcU 1R\*m1 + rcU 2R\*m2 + rcU 3RF\*m3 + rcU 4RF\*m4 + rcU 3RR\*m3 + rcU 4RF\*m4 + rcU 3RF\*m3 + rcU 4RF\*m4 + rcU 4RF\*m4 + rcU 3RF\*m3 + rcU 4RF\*m4 + rcU 3RF\*m3 + rcU 4RF\*m4 + rcU 4RF\*m4rcU 4RR\*m4 ... +  $rcU_1L*m1 + rcU_2L*m2 + rcU_3LF*m3 + rcU_4LF*m4 + rcU_3LR*m3 + rcU_4LF*m4 + rcU_4LF*m4 + rcU_3LR*m3 + rcU_4LF*m4 + rcU_4LF*m4 + rcU_4LF*m4 + rcU_3LR*m3 + rcU_4LF*m4 + rcU$ rcU\_4LR\*m4)/ ... (m0 + 2\*(m1 + m2) + 4\*(m3 + m4)); $rCM_4RF = rU_4RF - rU_CM;$  $rCM_4RR = rU_4RR - rU_CM;$  $rCM_4LF = rU_4LF - rU_CM;$  $rCM_4LR = rU_4LR - rU_CM;$ r0R\_4LF = inv(AU\_0R(1:3, 1:3)) \* rU\_4RF;  $rOR_CM = inv(AU_0R(1:3, 1:3)) * rU_CM;$  $rCM0_4LF = r0R_4LF - r0R_CM;$ alfa\_LF = atan2(rCM0\_4LF(1), rCM0\_4LF(3))\*180/pi + 90; alfa\_yU\_RF = atan2(rCM\_4RF(1), rCM\_4RF(3))\*180/pi + 90; alfa\_yU\_RR = atan2(rCM\_4RR(1), rCM\_4RR(3))\*180/pi + 90; alfa\_yU\_LF = atan2(rCM\_4LF(1), rCM\_4LF(3))\*180/pi + 90; alfa yU LR = atan2(rCM 4LR(1), rCM 4LR(3))\*180/pi + 90; alpha\_yU\_Front = alfa\_yU\_RF -alfa\_yU\_LF; alpha\_yU\_Rear = alfa\_yU\_RR -alfa\_yU\_LR; alfa\_zU\_RF = atan2(rCM\_4RF(1), rCM\_4RF(2))\*180/pi + 90; alfa\_zU\_RR = atan2(rCM\_4RR(1), rCM\_4RR(2))\*180/pi + 90; alfa\_zU\_LF = atan2(rCM\_4LF(1), rCM\_4LF(2))\*180/pi + 90;

alfa zU LR = atan2(rCM 4LR(1), rCM 4LR(2))\*180/pi + 90;

alpha\_zU\_Front = alfa\_zU\_RF -alfa\_zU\_LF; alpha\_zU\_Rear = alfa\_zU\_RR -alfa\_zU\_LR;

alfa weight = atan2(System Force U(1), System Force U(3))\*180/pi + 90;

```
w CM = [-43.56; 0; 0];
M4RF = cross(rCM \ 4RF, w \ CM);
M4RR = cross(rCM \ 4RR, w \ CM);
M4LF = cross(rCM \ 4LF, w \ CM);
M4LR = cross(rCM \ 4LR, w \ CM);
MU CM = cross(rU CM, w CM);
MU nSRF = cross(rU 4RF, [FnSRF; 0; 0]);
MU nSRR = cross(rU 4RR, [FnSRR; 0; 0]);
MU nSLF = cross(rU \ 4LF, [FnSLF; 0; 0]);
MU nSLR = cross(rU 4LR, [FnSLR; 0; 0]);
%
% Center of Mass Position vector with respect to universal farme
%
    rU CM = (rcU 0R*m0 + rcU 1R*m1 + rcU 2R*m2 + rcU 3RF*m3 + rcU 4RF*m4 + rcU 3RR*m3 + rcU 4RF*m4 + rcU 3RF*m3 + rcU 4RF*m4 + rcU 4RF*m4 + rcU 3RF*m3 + rcU 4RF*m4 + rcU 3RF*m3 + rcU 4RF*m4 + rcU 4RF*m4
rcU 4RR*m4 ...
             + rcU 1L*m1 + rcU 2L*m2 + rcU 3LF*m3 + rcU 4LF*m4 + rcU 3LR*m3 + rcU 4LR*m4)/...
              (m0 + 2*(m1 + m2) + 4*(m3 + m4));
%
% System Forces
%
FU = touchRF * AU_SRF(1:3,1:3)*[FnSRF - meu 0]' + ...
      touchRR * AU_SRR(1:3,1:3)*[FnSRR -meu 0]' + ...
      touchLF * AU_SLF(1:3,1:3)*[FnSLF -meu 0]' + ...
      touchLR * AU_SLR(1:3,1:3)*[FnSLR -meu 0]' + ...
      AU 4RF(1:3,1:3)*f4RR + AU 3RF(1:3,1:3)*f3RF + AU 4RR(1:3,1:3)*f4RR +
AU 3RR(1:3,1:3)*f3RR + ...
      AU 2R(1:3,1:3)*f2R + AU 1R(1:3,1:3)*f1R + AU 4LF(1:3,1:3)*f4LF + AU 3LF(1:3,1:3)*f3LF + ...
      AU_4LR(1:3,1:3)*f4LR + AU_3LR(1:3,1:3)*f3LR + AU_2L(1:3,1:3)*f2L + AU_1L(1:3,1:3)*f1L + ...
      AU_0R(1:3,1:3)*f0R;
fc = AU_4RF(1:3,1:3)*f4RF + AU_3RF(1:3,1:3)*f3RF + ...
      AU_4RR(1:3,1:3)*f4RR + AU_3RR(1:3,1:3)*f3RR + ...
      AU^{2}R(1:3,1:3)*f^{2}R + AU_{1}R(1:3,1:3)*f^{1}R + ...
      AU_0R(1:3,1:3)*f0R + AU_1L(1:3,1:3)*f1L + ...
      AU_2L(1:3,1:3)*f2L + AU_3LF(1:3,1:3)*f3LF + ...
      AU 4LF(1:3,1:3)*f4LF + AU 3LR(1:3,1:3)*f3LR + ...
      AU 4LR(1:3,1:3)*f4LR;
f gravity = [-Mass*g; 0; 0];
f inertial = fc - [-Mass*g; 0; 0];
%_
```

#### % System Moments

%

- MU = touchRF \* cross(rU 4RF, AU SRF(1:3,1:3)\*[FnSRF -meu\*FnSRF 0]') + ...touchRR \* cross(rU 4RR, AU SRR(1:3,1:3)\*[FnSRR -meu\*FnSRR 0]') + ... touchLF \* cross(rU 4LF, AU SLF(1:3,1:3)\*[FnSLF -meu\*FnSLF 0]') + ... touchLR \* cross(rU 4LR, AU SLR(1:3,1:3)\*[FnSLR -meu\*FnSLR 0]') + ... cross(rcU 4RF, AU 4RF(1:3,1:3)\*f4RF) + cross(rcU 3RF, AU 3RF(1:3,1:3)\*f3RF)+... cross(rcU 4RR, AU 4RR(1:3,1:3)\*f4RR) + cross(rcU 3RR, AU 3RR(1:3,1:3)\*f3RR)+... cross(rcU 2R, AU 2R(1:3,1:3)\*f2R) + cross(rcU 1R, AU 1R(1:3,1:3)\*f1R)+...cross(rcU 0R, AU 0R(1:3,1:3)\*f0R) + cross(rcU 1L, AU 1L(1:3,1:3)\*f1L)+... cross(rcU 2L, AU 2L(1:3,1:3)\*f2L) + cross(rcU 3LF, AU 3LF(1:3,1:3)\*f3LF)+... cross(rcU 4LF, AU 4LF(1:3,1:3)\*f4LF) + cross(rcU 3LR, AU 3LR(1:3,1:3)\*f3LR)+... cross(rcU 4LR, AU 4LR(1:3,1:3)\*f4LR) + ...AU 4RF(1:3,1:3)\*tow4RF + AU 3RF(1:3,1:3)\*tow3RF + ... AU 4RR(1:3,1:3)\*tow4RR + AU 3RR(1:3,1:3)\*tow3RR + ... AU 4LF(1:3,1:3)\*tow4LF + AU 3LF(1:3,1:3)\*tow3LF + ... AU 4LR(1:3,1:3)\*tow4LR + AU 3LR(1:3,1:3)\*tow3LR + ... AU 2R(1:3,1:3)\*tow2R + AU 1R(1:3,1:3)\*tow1R + ... AU 2L(1:3,1:3)\*tow2L + AU 1L(1:3,1:3)\*tow1L + ... AU 0R(1:3,1:3)\*tow0R;
- $towc = AU_4RF(1:3,1:3)*tow4RF + AU_3RF(1:3,1:3)*tow3RF + ... \\ AU_4RR(1:3,1:3)*tow4RR + AU_3RR(1:3,1:3)*tow3RR + ... \\ AU_4LF(1:3,1:3)*tow4LF + AU_3LF(1:3,1:3)*tow3LF + ... \\ AU_4LR(1:3,1:3)*tow4LR + AU_3LR(1:3,1:3)*tow3LR + ... \\ AU_2R(1:3,1:3)*tow2R + AU_1R(1:3,1:3)*tow1R + ... \\ AU_2L(1:3,1:3)*tow2L + AU_1L(1:3,1:3)*tow1L + ... \\ AU_0R(1:3,1:3)*tow0R;$
- $\begin{array}{l} fc\_Moment = cross(rcU\_4RF, AU\_4RF(1:3,1:3)*f4RF) + cross(rcU\_3RF, AU\_3RF(1:3,1:3)*f3RF) + ... \\ cross(rcU\_4RR, AU\_4RR(1:3,1:3)*f4RR) + cross(rcU\_3RR, AU\_3RR(1:3,1:3)*f3RR) + ... \\ cross(rcU\_2R, AU\_2R(1:3,1:3)*f2R) & + cross(rcU\_1R, AU\_1R(1:3,1:3)*f1R) + ... \\ cross(rcU\_0R, AU\_0R(1:3,1:3)*f0R) & + cross(rcU\_1L, AU\_1L(1:3,1:3)*f1L) + ... \\ cross(rcU\_2L, AU\_2L(1:3,1:3)*f2L) & + cross(rcU\_3LF, AU\_3LF(1:3,1:3)*f3LF) + ... \\ cross(rcU\_4LF, AU\_4LF(1:3,1:3)*f4LF) + cross(rcU\_3LR, AU\_3LR(1:3,1:3)*f3LR) + ... \\ cross(rcU\_4LR, AU\_4LR(1:3,1:3)*f4LR); \end{array}$

f\_gravity\_Moment = cross(rU\_CM, f\_gravity);

f\_inertial\_Moment = cross(rU\_CM, f\_inertial);

$$\label{eq:Normal_Forces_moments} \begin{split} &\text{Normal_Forces_moments} = \text{touchRF} * \text{cross}(rU_4RF, AU_SRF(1:3,1:3)*[FnSRF -meu 0]') + ... \\ & \quad \text{touchRR} * \text{cross}(rU_4RR, AU_SRR(1:3,1:3)*[FnSRR -meu 0]') + ... \\ & \quad \text{touchLF} * \text{cross}(rU_4LF, AU_SLF(1:3,1:3)*[FnSLF -meu 0]') + ... \\ & \quad \text{touchLR} * \text{cross}(rU_4LR, AU_SLR(1:3,1:3)*[FnSLR -meu 0]'); \end{split}$$

%

% Masses on the four legs

%

mSRF = FnSRF / (g \* cos(Pitch)); mSRR = FnSRR / (g \* cos(Pitch)); mSLF = FnSLF / (g \* cos(Pitch));

```
 \begin{array}{l} \text{function for closs(rU_4RR, AU_4RF(1:3,1:3)*f4RF) + cross(rcU_3RF, AU_3RF(1:3,1:3)*f3RF) + ... \\ \text{cross(rcU_4RR, AU_4RR(1:3,1:3)*f4RR) + cross(rcU_3RR, AU_3RR(1:3,1:3)*f3RF) + ... \\ \text{cross(rcU_2R, AU_2R(1:3,1:3)*f2R) + cross(rcU_1R, AU_1R(1:3,1:3)*f1R) + ... \\ \text{cross(rcU_0R, AU_0R(1:3,1:3)*f0R) + cross(rcU_1L, AU_1L(1:3,1:3)*f1L) + ... \\ \text{cross(rcU_2L, AU_2L(1:3,1:3)*f2L) + cross(rcU_3LF, AU_3LF(1:3,1:3)*f3LF) + ... \\ \text{cross(rcU_4LF, AU_4LF(1:3,1:3)*f4LF) + cross(rcU_3LR, AU_3LR(1:3,1:3)*f3LR) + ... \\ \text{cross(rcU_4LR, AU_4LF(1:3,1:3)*f4LR) + ... \\ \text{AU_4RF(1:3,1:3)*tow4RF + AU_3RF(1:3,1:3)*tow3RF + ... \\ \text{AU_4LF(1:3,1:3)*tow4LF + AU_3RR(1:3,1:3)*tow3LF + ... \\ \text{AU_4LF(1:3,1:3)*tow4LF + AU_3LF(1:3,1:3)*tow3LF + ... \\ \text{AU_2R(1:3,1:3)*tow4LF + AU_3LR(1:3,1:3)*tow3LF + ... \\ \text{AU_2L(1:3,1:3)*tow2R + AU_1R(1:3,1:3)*tow1R + ... \\ \text{AU_2L(1:3,1:3)*tow0R; \\ \end{array}
```

```
if FNSRF <= 0
FNSRF = 0;
end
if FNSLF <= 0
FNSLF = 0;
end
```

FNSLF = -M1(2)/(touchLF\*B2(2));

```
 \begin{array}{l} tow\_Critical\_Front = touchRF * cross(rU\_4RF, AU\_SRF(1:3,1:3)*[FNSRF -meu*FNSRF 0]') + ... \\ touchLF * cross(rU\_4LF, AU\_SLF(1:3,1:3)*[FNSLF -meu*FNSLF 0]') + ... \\ cross(rcU\_4RF, AU\_4RF(1:3,1:3)*f4RF) + cross(rcU\_3RF, AU\_3RF(1:3,1:3)*f3RF) + ... \\ cross(rcU\_4RR, AU\_4RR(1:3,1:3)*f4RR) + cross(rcU\_3RR, AU\_3RR(1:3,1:3)*f3RR) + ... \\ cross(rcU\_2R, AU\_2R(1:3,1:3)*f2R) + cross(rcU\_1R, AU\_1R(1:3,1:3)*f1R) + ... \\ cross(rcU\_0R, AU\_0R(1:3,1:3)*f0R) + cross(rcU\_1L, AU\_1L(1:3,1:3)*f1L) + ... \\ cross(rcU\_2L, AU\_2L(1:3,1:3)*f2L) + cross(rcU\_3LF, AU\_3LF(1:3,1:3)*f3LF) + ... \\ cross(rcU\_4LF, AU\_4LF(1:3,1:3)*f4LF) + cross(rcU\_3LR, AU\_3LR(1:3,1:3)*f3LR) + ... \\ cross(rcU\_4LR, AU\_4LF(1:3,1:3)*f4LR) + ... \\ cross(rcU\_4LR, AU\_4LR(1:3,1:3)*f4LR) + ... \\ cross(rcU\_4LR, AU\_4LR(1:3,1:3)*f4LR) + ... \\ AU\_4RF(1:3,1:3)*tow4RF + AU\_3RF(1:3,1:3)*tow3RF + ... \\ \end{array}
```

```
AU 4RR(1:3,1:3)*tow4RR + AU 3RR(1:3,1:3)*tow3RR + ...
           AU 4LF(1:3,1:3)*tow4LF + AU 3LF(1:3,1:3)*tow3LF + ...
           AU 4LR(1:3,1:3)*tow4LR + AU 3LR(1:3,1:3)*tow3LR + ...
           AU 2R(1:3,1:3)*tow2R + AU 1R(1:3,1:3)*tow1R + ...
           AU 2L(1:3,1:3)*tow2L + AU 1L(1:3,1:3)*tow1L + ...
           AU 0R(1:3,1:3)*tow0R;
%FNSRF
%FNSLF
FNSLF = -M4(3)/(touchLF*E3(3));
FNSLR = -M3(3)/(touchLR*D3(3));
if FNSLF \leq 0
 FNSLF = 0;
end
if FNSLR \leq 0
 FNSLR = 0;
end
tow Critical Left = touchLF * cross(rU 4LF, AU SLF(1:3,1:3)*[FNSLF -meu*FNSLF 0]') + ...
          touchLR * cross(rU 4LR, AU SLR(1:3,1:3)*[FNSLR -meu*FNSLR 0]') + ...
          cross(rcU 4RF, AU 4RF(1:3,1:3)*f4RF) + cross(rcU 3RF, AU 3RF(1:3,1:3)*f3RF)+...
          cross(rcU 4RR, AU 4RR(1:3,1:3)*f4RR) + cross(rcU 3RR, AU 3RR(1:3,1:3)*f3RR)+...
          cross(rcU 2R, AU 2R(1:3,1:3)*f2R) + cross(rcU 1R, AU 1R(1:3,1:3)*f1R)+...
          cross(rcU_0R, AU_0R(1:3,1:3)*f0R) + cross(rcU_1L, AU_1L(1:3,1:3)*f1L)+...
          cross(rcU_2L, AU_2L(1:3,1:3)*f2L) + cross(rcU_3LF, AU_3LF(1:3,1:3)*f3LF)+...
          cross(rcU_4LF, AU_4LF(1:3,1:3)*f4LF) + cross(rcU_3LR, AU_3LR(1:3,1:3)*f3LR)+...
          cross(rcU_4LR, AU_4LR(1:3,1:3)*f4LR) + ...
          AU 4RF(1:3,1:3)*tow4RF + AU_3RF(1:3,1:3)*tow3RF + ...
          AU_4RR(1:3,1:3)*tow4RR + AU_3RR(1:3,1:3)*tow3RR + ...
          AU_4LF(1:3,1:3)*tow4LF + AU_3LF(1:3,1:3)*tow3LF + ...
          AU_4LR(1:3,1:3)*tow4LR + AU_3LR(1:3,1:3)*tow3LR + ...
          AU_2R(1:3,1:3)*tow2R + AU_1R(1:3,1:3)*tow1R + ...
          AU 2L(1:3,1:3)*tow2L + AU 1L(1:3,1:3)*tow1L + ...
          AU 0R(1:3,1:3)*tow0R;
FNSRR = -M4(2)/(touchRR*E2(2));
FNSLR = -M2(2)/(touchLR*C3(2));
if FNSRR <= 0
 FNSRR = 0;
end
if FNSLR <= 0
 FNSLR = 0;
end
tow Critical Rear = touchRR * cross(rU 4RR, AU SRR(1:3,1:3)*[FNSRR -meu*FNSRR 0]') + ...
          touchLR * cross(rU 4LR, AU SLR(1:3,1:3)*[FNSLR -meu*FNSLR 0]') + ...
          cross(rcU_4RF, AU_4RF(1:3,1:3)*f4RF) + cross(rcU_3RF, AU_3RF(1:3,1:3)*f3RF)+...
          cross(rcU_4RR, AU_4RR(1:3,1:3)*f4RR) + cross(rcU_3RR, AU_3RR(1:3,1:3)*f3RR)+...
          cross(rcU 2R, AU 2R(1:3,1:3)*f2R) + cross(rcU 1R, AU 1R(1:3,1:3)*f1R)+...
```

```
cross(rcU 0R, AU 0R(1:3,1:3)*f0R) + cross(rcU 1L, AU 1L(1:3,1:3)*f1L)+...
         cross(rcU 2L, AU 2L(1:3,1:3)*f2L) + cross(rcU 3LF, AU 3LF(1:3,1:3)*f3LF)+...
         cross(rcU_4LF, AU_4LF(1:3,1:3)*f4LF) + cross(rcU_3LR, AU_3LR(1:3,1:3)*f3LR)+...
         cross(rcU 4LR, AU 4LR(1:3,1:3)*f4LR) + ...
         AU 4RF(1:3,1:3)*tow4RF + AU 3RF(1:3,1:3)*tow3RF + ...
         AU 4RR(1:3,1:3)*tow4RR + AU 3RR(1:3,1:3)*tow3RR + ...
         AU 4LF(1:3,1:3)*tow4LF + AU 3LF(1:3,1:3)*tow3LF + ...
         AU 4LR(1:3,1:3)*tow4LR + AU 3LR(1:3,1:3)*tow3LR + ...
         AU 2R(1:3,1:3)*tow2R + AU 1R(1:3,1:3)*tow1R + ...
         AU 2L(1:3,1:3)*tow2L + AU 1L(1:3,1:3)*tow1L + ...
         AU 0R(1:3,1:3)*tow0R;
%FNSRR
%FNSLR
        pause
```

## DH.m

% A Denavit Hartenberg Parameters describes the kinematics of a manipulator

%

%

% these DH Parameters are filled in matrix, each row represents one link of

% the manipulator

% our mobile robot have no prismatic joint. so, the variable joints are joints' angles % represented in theta.

% All joints' angles are defined in radians.

function dh = DH(q)

theta = q;

%theta(4); % theta 4 does not effect on the manipulation.

% theta %	d	a	alpha	sigma
dh=[ 0	0	0	0	0
theta(1)	0.2	0	-pi/2	0
theta(2)	0	0	pi/2	0
theta(3)	0	0.4	0	0
theta(4)	0	0.05	0	0];

## DHtransformation.m

 $%T = [\cos(\text{theta}) - \sin(\text{theta}) \cos(\text{alpha}) \sin(\text{theta}) \sin(\text{alpha}) a \cos(\text{theta})$ % sin(theta) cos(theta)\*cos(alpha) -cos(theta)\*sin(alpha) a\*sin(theta) % 0 sin(alpha) cos(alpha) d % 0 0 0 1 ];

function [T] = DHtransformation(theta, d, a, alpha)

T = rotz(theta) \* translation(0,0,d) \* translation(a,0,0) \* rotx(alpha);

#### **Dynamics.m**

function D = Dynamics(d, a)

d1 = d(1);d2 = 0;a3 = a(3);a4 = a(4);a = 0.30; % half length of the platform in meter b = 0.02; % half sickness of the platform in meter %------ Platform Mass, Volume and Density ------% kg m0 = 4;m1 = 1; % kg % kg  $m^2 = 0;$ m3 = 1; % kg %------ wheel Mass, Volume and Density -----a4 ex = a4;a4\_in = 0.03; %density = 79.577471; density = 98.999999; Volume4\_ex =  $pi * a4_ex^2$ ; Volume4\_in =  $pi * a4_in^2$ ; m4\_ex = density \* Volume4\_ex; % kg m4\_in = density \* Volume4\_in; % kg % kg,  $m4 = m4_{ex} - m4_{in} = 0.4976$ ; m4 = 0.5;m = [m0, m1, m2, m3, m4]'; %

```
rc 0 = [0]
               0
                     0]';
              0.5*d1 0 ]':
rc 1 = [0]
rc 2 = [0]
               0
                     -0.5*d2]';
                     0 ]';
rc_3 = [-0.5*a3]
               0
rc_4 = [-a4]
              0
                      0 ]';
rc = [rc_0, rc_1, rc_2, rc_3, rc_4]';
%-----
% Inertia matrices
I0 = (m0/3)*[a^2+d1^2 0 0; 0 b^2+d1^2 0; 0 a^2+b^2];
I1 = (m1*d1^2/12)*[1\ 0\ 0;\ 0\ 0\ 0;\ 0\ 0\ 1];
I2 = (m2*d2^2/12)*[1 0 0; 0 1 0; 0 0 0];
I3 = (m3*a3^2/12)*[0\ 0\ 0;\ 0\ 1\ 0;\ 0\ 0\ 1];
I4 ex = (m4 ex^{2}) [1/4 0 0; 0 1/4 0; 0 0 1/2];
I4_in = (m4_in*a4_in^2)*[1/4 \ 0 \ 0; \ 0 \ 1/4 \ 0; \ 0 \ 0 \ 1/2];
I4 = I4_{ex} - I4_{in};
I = [I0(1,1) I0(2,2) I0(3,3) I0(1,2) I0(2,3) I0(1,3);...
   I1(1,1) I1(2,2) I1(3,3) I1(1,2) I1(2,3) I1(1,3);...
   I2(1,1) I2(2,2) I2(3,3) I2(1,2) I2(2,3) I2(1,3);...
   I3(1,1) I3(2,2) I3(3,3) I3(1,2) I3(2,3) I3(1,3);...
   I4(1,1) I4(2,2) I4(3,3) I4(1,2) I4(2,3) I4(1,3)];
%-----
```

% m rx ry rz Ixx Iyy Izz I D= [m, rc, I];

## HT\_2\_RPY.m

function [alpha\_1,alpha\_2,alpha\_3] = HT\_2\_RPY(A\_0\_4)

%-----  $nx = A_0_4(1,1);$   $ny = A_0_4(2,1);$   $nz = A_0_4(3,1);$   $ox = A_0_4(1,2);$   $oy = A_0_4(2,2);$   $oz = A_0_4(3,2);$   $ax = A_0_4(1,3);$   $ay = A_0_4(2,3);$   $az = A_0_4(3,3);$ %------

 $alpha_2 = atan_2(-nz, nx);$ 

```
alpha_3 = atan2(ny, (cos(alpha_2)*nx - sin(alpha_2)*nz));

alpha_1 = atan2((sin(alpha_2)*ox + cos(alpha_2)*oz), (sin(alpha_2)*ax + cos(alpha_2)*az));

%------
```

## Invkinematic.m

```
function theta = invkinematic(T, s)
TOL = 0.001;
[nr, nc] = size(T);
% test for accuracy
for i=1:nr
  for j=1:nc
     if abs(T(i,j)) \leq TOL
     T(i,j) = 0;
     end
  end
end
d1 = 0.2;
a3 = 0.4;
a4 = 0.05;
  nx = T(1,1);
        ny = T(2,1);
        nz = T(3,1);
        ox = T(1,2);
        oy = T(2,2);
        oz = T(3,2);
        ax = T(1,3);
        ay = T(2,3);
        az = T(3,3);
        px = T(1,4);
        py = T(2,4);
        pz = T(3,4);
  theta_1 = atan2(ay,ax);
```

theta\_2 =  $atan2(-(-pz+d1), -(cos(theta_1)*px + sin(theta_1)*py));$ 

 $K1 = \cos(\text{theta}_1) * \cos(\text{theta}_2) * px + \sin(\text{theta}_1) * \cos(\text{theta}_2) * py - \sin(\text{theta}_2) * pz + \sin(\text{theta}_2) * d1;$  $K2 = -\sin(\text{theta}_1) * px + \cos(\text{theta}_1) * py;$ 

if (s == 'RF')

theta\_4 =  $acos((K1^2 + K2^2 - (a4^2+a3^2))/(2*a3*a4));$ elseif (s == 'LF') theta\_4 =  $-acos((K1^2 + K2^2 - (a4^2+a3^2))/(2*a3*a4));$ 

end

```
\begin{array}{l} K3 = a4*cos(theta_4) + a3;\\ K4 = a4*sin(theta_4);\\ theta_3 = atan2((K1*K4 - K2*K3), -(K1*K3 + K2*K4));\\ \end{array}
```

```
theta = [theta_1; theta_2; theta_3; theta_4]*180/pi;
```

#### Kinematic.m

function  $[A \ 0 \ 4] =$  Kinematic(theta, d, a, alpha, B3, pitch)

theta(4) = -theta(1) - theta(3) + B3 - pitch; % theta\_4 manipulated in contact point of wheel with ground theta(3) = theta(3) + pi;

for i=1:4

A = DHtransformation(theta(i), d(i), a(i), alpha(i));

if (i==1)  

$$A_0_1 = A;$$
  
elseif (i==2)  
 $A_1_2 = A;$   
elseif (i==3)  
 $A_2_3 = A;$   
elseif (i==4)  
 $A_3_4 = A;$ 

end;

end;

 $A_0_4 = A_0_1 * A_1_2 * A_2_3 * A_3_4$ ; %Homogeneous Transformation from base to end-effector frame

### locomotion\_DN.m

%

% increasing velocity linearly = vv\*t(p), and constant acceleration = vv %

function [A\_4RF, A\_4RR, A\_4LF, A\_4LR, V\_4RF, V\_4RR, V\_4LF, V\_4LR, d\_4RF, d\_4RR, d\_4LF, d\_4LR,...

Tdd\_RF, Tdd\_RR, Tdd\_LF, Tdd\_LR, Td\_RF, Td\_RR, Td\_LF, Td\_LR,...

T\_RF, T\_RR, T\_LF, T\_LR, tdelay\_R, tdelay\_L] = locomotion\_DN(Touch, vv, t, a, q0)

np = numcols(t);

At4RF = zeros(np,1); At4RR = zeros(np,1); At4LF = zeros(np,1); At4LR = zeros(np,1); Vt4RF = zeros(np,1); Vt4RR = zeros(np,1); Vt4LR = zeros(np,1); Vt4LR = zeros(np,1); D\_4RF = zeros(1,np); D\_4RR = zeros(1,np); D\_4LR = zeros(1,np); D\_4R = zeros(1,np)

Thetadd\_RF = zeros(np,1); Thetadd\_RR = zeros(np,1); Thetadd\_LF = zeros(np,1); Thetadd\_LR = zeros(np,1); Thetad\_RF = zeros(np,1); Thetad\_RR = zeros(np,1); Thetad\_LF = zeros(np,1); Thetad\_LR = zeros(np,1); Theta RF = zeros(np,1); Theta RR = zeros(np,1); Theta LF = zeros(np,1); Theta LR = zeros(np,1);

for p=1:np,

At4RF(p) = Touch(1)\*vv; % m/(sec\*sec)At4RR(p) = Touch(2)\*vv; % m/(sec\*sec)At4LF(p) = -Touch(3)\*vv; % m/(sec\*sec)At4LR(p) = -Touch(4)\*vv; % m/(sec\*sec)Vt4RF(p) = Touch(1)\*vv\*t(p); % m/secVt4RR(p) = Touch(2)\*vv\*t(p); % m/secVt4LF(p) = -Touch(3)\*vv\*t(p); % m/secVt4LR(p) = -Touch(4)\*vv\*t(p); % m/secD 4RF(p) = Touch(1)\*vv\*( $0.5*t(p)^2$ ); % m D  $4RR(p) = Touch(2)*vv*(0.5*t(p)^2); \% m$ D 4LF(p) = Touch(3)\*vv\*(-0.5\*t(p)^2); % m  $D_4LR(p) = Touch(4)*vv*(-0.5*t(p)^2); \% m$ Thetadd\_RF(p) = At4RF(p)/a(4); % rad/(sec\*sec)Thetadd\_RR(p) = At4RR(p)/a(4); % rad/(sec\*sec)Thetadd\_LF(p) = At4LF(p)/a(4); % rad/(sec\*sec) Thetadd\_LR(p) = At4LR(p)/a(4); % rad/(sec\*sec) Thetad RF(p) = Vt4RF(p)/a(4); % rad/sec Thetad RR(p) = Vt4RR(p)/a(4); % rad/sec Thetad\_LF(p) = Vt4LF(p)/a(4); % rad/sec Thetad LR(p) = Vt4LR(p)/a(4); % rad/sec Theta\_RF(p) =  $D_4RF(p)/a(4)$ ; % rad Theta\_RR(p) =  $D_4RR(p)/a(4)$ ; % rad Theta\_LF(p) =  $D_4LF(p)/a(4)$ ; % rad Theta\_LR(p) =  $D_4LR(p)/a(4)$ ; % rad

end

%

% time delay occured between the front and rear legs on both sides; right % and front sides % tdelay\_R = sqrt((-a(3)\*sin(q0(3,1)) + a(3)\*sin(q0(3,2)))/(vv\*0.5)); tdelay\_L = sqrt(( a(3)\*sin(q0(3,3)) - a(3)\*sin(q0(3,4)))/(vv\*0.5)); % A 4RF = At4RF; A 4RR = At4RR; A 4LF = At4LF; A 4LR = At4LR;  $V_{4RF} = Vt4RF; V_{4RR} = Vt4RR; V_{4LF} = Vt4LF; V_{4LR} = Vt4LR;$ 

 $d_4RF = D_4RF; d_4RR = D_4RR;$  $d_4LF = D_4LF; d_4LR = D_4LR;$ 

Tdd\_RF = Thetadd\_RF; Tdd\_RR = Thetadd\_RR; Tdd\_LF = Thetadd\_LF; Tdd\_LR = Thetadd\_LR;

Td\_RF = Thetad\_RF; Td\_RR = Thetad\_RR; Td\_LF = Thetad\_LF; Td\_LR = Thetad\_LR;

 $T_RF = Theta_RF;$   $T_RR = Theta_RR;$  $T_LF = Theta_LF;$   $T_LR = Theta_LR;$ 

#### Rotx.m

% % homogeneous transformation for a rotation of t about the x-axis. % function r = rotx(t)

## Roty.m

% % homogeneous transformation for a rotation of t about the y-axis. %

function r = roty(t)

r = [cos(t)]	0	sin(t)	0
0	1	0	0
-sin(t)	0	$\cos(t)$	0
0	0	0	1];

#### Rotz.m

% % homogeneous transformation for a rotation of t about the z-axis. %

function r = rotz(t)

r =	[cos(t)	-sin(t)	0	0
	sin(t)	$\cos(t)$	0	0
	0	0	1	0
	0	0	0	1]

# GG1.m

% % Flat Surface % function [input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs,... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG15(t, tdelay\_R, tdelay\_L, a, q0), ns = numcols(t);tm = t(ns)/2; $tp_R = tdelay_R;$ %time delay b/w RF and RR  $tp_L = tdelay_L;$ %time delay b/w LF and LR  $LEG_RF = zeros(1,ns);$  $LEG_RR = zeros(1,ns);$  $LEG_LR = zeros(1,ns);$ LEG\_LF = zeros(1,ns); BETA\_SRF\_zs = zeros(1, ns); BETA\_SRR\_zs = zeros(1, ns); BETA\_SLF\_zs = zeros(1, ns); BETA SLR zs = zeros(1, ns);BETA\_SRF\_ys = zeros(1, ns); BETA\_SRR\_ys = zeros(1, ns); BETA\_SLF\_ys = zeros(1, ns); BETA\_SLR\_ys = zeros(1, ns); for p=1:ns LEG RF(p) = 3; $LEG_RR(p) = 3;$  $LEG_LF(p) = 3;$  $LEG_LR(p) = 3;$ end for p=1:ns  $BETA\_SRF\_zs(p) = 0;$  $BETA_SRR_z(p) = 0;$  $BETA\_SLF\_zs(p) = 0;$  $BETA_SLR_zs(p) = 0;$  $BETA_SRF_ys(p) = 0;$  $BETA_SRR_ys(p) = 0;$  $BETA\_SLF\_ys(p) = 0;$  $BETA_SLR_ys(p) = 0;$ end BETA\_SRR\_zs\*180/pi;

input\_RF = LEG\_RF; input\_RR = LEG\_RR; input\_LF = LEG\_LF; input\_LR = LEG\_LR; beta\_SRF\_zs = BETA\_SRF\_zs; beta\_SRR\_zs = BETA\_SRR\_zs; beta\_SLF\_zs = BETA\_SLF\_zs; beta\_SLR\_zs = BETA\_SLR\_zs; beta\_SRF\_ys = BETA\_SLR\_zs; beta\_SRF\_ys = BETA\_SRF\_ys; beta\_SRR\_ys = BETA\_SRR\_ys;

beta\_SRR\_ys = BETA\_SRR\_ys; beta\_SLF\_ys = BETA\_SLF\_ys; beta\_SLR\_ys = BETA\_SLR\_ys;

## GG2.m

% % Step flat-inclined surface %

function [input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs,... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG15(t, tdelay\_R, tdelay\_L, a, q0),

ns = numcols(t);

tp\_R = tdelay\_R; %time delay b/w RF and RR tp\_L = tdelay\_L; %time delay b/w LF and LR

 $LEG_RF = zeros(1,ns);$  $LEG_RR = zeros(1,ns);$  $LEG_LR = zeros(1,ns);$ LEG\_LF = zeros(1,ns);  $BETA\_SRF\_zs = zeros(1, ns);$ BETA\_SRR\_zs = zeros(1, ns); BETA\_SLF\_zs = zeros(1, ns); BETA\_SLR\_zs = zeros(1, ns); BETA\_SRF\_ys = zeros(1, ns); BETA\_SRR\_ys = zeros(1, ns); BETA\_SLF\_ys = zeros(1, ns); BETA\_SLR\_ys = zeros(1, ns); for p=1:ns  $LEG_RF(p) = 3.2;$  $LEG_{RR(p)} = 3.2;$ LEG LF(p) = 3;LEG LR(p) = 3;

end

for p=1:ns BETA SRF zs(p) = 0; BETA SRR zs(p) = 0;BETA SLF zs(p) = 0;  $BETA_SLR_zs(p) = 0;$ BETA SRF  $y_s(p) = 0$ ; BETA SRR ys(p) = 0;BETA SLF ys(p) = pi/8;BETA SLR ys(p) = pi/8;end %BETA SRR zs\*180/pi; input RF = LEG RF; input RR = LEG RR; input LF = LEG LF; input\_LR = LEG\_LR; beta SRF zs = BETA SRF zs; beta SRR zs = BETA SRR zs; beta SLF zs = BETA SLF zs; beta\_SLR\_zs = BETA\_SLR\_zs; beta SRF ys = BETA SRF ys; beta\_SRR\_ys = BETA\_SRR\_ys; beta\_SLF\_ys = BETA\_SLF\_ys;

beta\_SLR\_ys = BETA\_SLR\_ys;

# GG9.m

% Inclined surface %

function [input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs,... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG9(t, tdelay\_R, tdelay\_L, a, q0),

ns = numcols(t);

tp\_R = tdelay\_R; %time delay b/w RF and RR tp\_L = tdelay\_L; %time delay b/w LF and LR

LEG\_RF = zeros(1,ns); LEG\_RR = zeros(1,ns); LEG\_LR = zeros(1,ns); LEG\_LF = zeros(1,ns); 291

BETA SRF zs = zeros(1, ns);BETA SRR zs = zeros(1, ns);BETA SLF zs = zeros(1, ns);BETA\_SLR\_zs = zeros(1, ns); BETA SRF ys = zeros(1, ns); BETA\_SRR\_ys = zeros(1, ns); BETA SLF ys = zeros(1, ns);BETA SLR ys = zeros(1, ns);for p=1:ns LEG RF(p) = 3.2;LEG RR(p) = 3; LEG LF(p) = 3.2; $LEG_LR(p) = 3;$ end for p=1:ns BETA\_SRF\_zs(p) = -20.70808185\*pi/180; BETA SRR zs(p) = -20.70808185\*pi/180; BETA SLF zs(p) = -20.70808185\*pi/180;BETA SLR zs(p) = -20.70808185\*pi/180; BETA SRF ys(p) = 0;BETA SRR ys(p) = 0; $BETA\_SLF\_ys(p) = 0;$  $BETA_SLR_ys(p) = 0;$ end input\_RF = LEG\_RF; input\_RR = LEG\_RR; input\_LF = LEG\_LF; input\_LR = LEG\_LR; beta\_SRF\_zs = BETA\_SRF\_zs; beta\_SRR\_zs = BETA\_SRR\_zs; beta\_SLF\_zs = BETA\_SLF\_zs; beta\_SLR\_zs = BETA\_SLR\_zs; beta\_SRF\_ys = BETA\_SRF\_ys; beta\_SRR\_ys = BETA\_SRR\_ys; beta\_SLF\_ys = BETA\_SLF\_ys; beta\_SLR\_ys = BETA\_SLR\_ys;

## GG5.m

% flat surface, then inclined surface %

292

function [input RF, input RR, input LF, input LR, ... beta SRF zs, beta SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs,... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG5(t, tdelay\_R, tdelay\_L, a, q0), ns = numcols(t);tm = t(ns)/2;tp R = tdelay R; %time delay b/w RF and RR tp L = tdelay L;%time delay b/w LF and LR LEG RF = zeros(1,ns);LEG RR = zeros(1,ns); LEG LR = zeros(1,ns);LEG LF = zeros(1,ns);BETA SRF zs = zeros(1, ns);BETA SRR zs = zeros(1, ns);BETA SLF zs = zeros(1, ns); $BETA\_SLR\_zs = zeros(1, ns);$ BETA SRF ys = zeros(1, ns);BETA SRR ys = zeros(1, ns);BETA SLF ys = zeros(1, ns);BETA SLR ys = zeros(1, ns);theta R = zeros(1, ns);theta\_R = zeros(1, ns); $slope_R = zeros(1, ns);$  $slope_L = zeros(1, ns);$ % % Slope % for p=1:ns  $if t(p) \le tm$ theta\_R = 0; theta\_L = 0; elseif t(p) > tmtheta\_R = pi/6; theta\_L = pi/6; end slope R = ((-a(3)\*sin(q0(3,1))+a(3)\*sin(q0(3,2)))\*sin(theta R))/tp R; $slope_L = ((a(3)*sin(q0(3,3))-a(3)*sin(q0(3,4)))*sin(theta_L))/tp_L;$ end % % Surface Function % for p=1:ns

```
if t(p) \le tm
    LEG RF(p) = 3;
    LEG RR(p) = 3;
    LEG LF(p) = 3;
    LEG LR(p) = 3;
  elseif (t(p) > tm) & (t(p) <= (tm + tp_R))
    LEG RF(p) = 3 + slope R^{*}(t(p) - tm);
    LEG RR(p) = 3;
    LEG LF(p) = 3 + slope L^*(t(p) - tm);
    LEG LR(p) = 3;
  elseif t(p) > (tm + tp R)
    LEG RF(p) = 3 + slope R^{*}(t(p) - tm);
    LEG RR(p) = 3 + slope R^{*}(t(p) - tp R - tm);
    LEG LF(p) = 3 + slope L^*(t(p) - tm);
    LEG_LR(p) = 3 + slope_L^*(t(p) - tp_L - tm);
  end
end
for p=1:ns
  if t(p) \le tm
    BETA\_SRF\_zs(p) = 0;
    BETA\_SRR\_zs(p) = 0;
    BETA\_SLF\_zs(p) = 0;
    BETA_SLR_zs(p) = 0;
  elseif (t(p) > tm) & (t(p) <= (tm + tp_R))
    BETA_SRF_zs(p) = -theta_R; %-atan(slope_R);
                                 %-atan(slope_R);
    BETA\_SRR\_zs(p) = 0;
    BETA_SLF_zs(p) = -theta_L; %-atan(slope_L);
    BETA\_SLR\_zs(p) = 0;
                                 %-atan(slope L);
  elseif t(p) > (tm + tp_R)
    BETA_SRF_zs(p) = -theta_R; %-atan(slope_R);
    BETA_SRR_zs(p) = -theta_R; %-atan(slope_R);
    BETA_SLF_zs(p) = -theta_L; %-atan(slope_L);
    BETA_SLR_zs(p) = -theta_L; %-atan(slope_L);
  end
  BETA\_SRF\_ys(p) = 0;
  BETA_SRR_ys(p) = 0;
  BETA_SLF_ys(p) = 0;
  BETA_SLR_ys(p) = 0;
end
input_RF = LEG_RF;
input_RR = LEG_RR;
```

input\_LR = LEG\_LR; beta\_SRF\_zs = BETA\_SRF\_zs; beta\_SRR\_zs = BETA\_SRR\_zs; beta\_SLF\_zs = BETA\_SLF\_zs; beta\_SLR\_zs = BETA\_SLR\_zs; beta\_SRF\_ys = BETA\_SRF\_ys; beta\_SRR\_ys = BETA\_SRR\_ys; beta\_SLF\_ys = BETA\_SRR\_ys; beta\_SLF\_ys = BETA\_SLF\_ys; beta\_SLR\_ys = BETA\_SLR\_ys;

input LF = LEG LF;

## GG7.m

% % Sinusoidal Surface % function [input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs,... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG16(t, tdelay\_R, tdelay\_L, a, q0), ns = numcols(t);tm = t(ns)/2; $tp_R = tdelay_R$ ; %time delay b/w RF and RR tp\_L = tdelay\_L; %time delay b/w LF and LR LEG RF = zeros(1,ns); $LEG_RR = zeros(1,ns);$  $LEG_LR = zeros(1,ns);$ LEG\_LF = zeros(1,ns);  $BETA\_SRF\_zs = zeros(1, ns);$  $BETA\_SRR\_zs = zeros(1, ns);$ BETA\_SLF\_zs = zeros(1, ns); BETA\_SLR\_zs = zeros(1, ns);

BETA\_SRF\_ys = zeros(1, ns); BETA\_SRR\_ys = zeros(1, ns); BETA\_SLF\_ys = zeros(1, ns); BETA\_SLR\_ys = zeros(1, ns);

 $n=20; \\ Am =0.5; \\ for p=1:ns \\ LEG_RF(p) = 3 + Am*sin((pi*(t(p) + tp_R/2))/(n*tp_R)); \\ LEG_RR(p) = 3 + Am*sin((pi*(t(p) - tp_R/2))/(n*tp_R)); \\ LEG_LF(p) = 3 + Am*sin((pi*(t(p) + tp_R/2))/(n*tp_L)); \\ LEG_LR(p) = 3 + Am*sin((pi*(t(p) - tp_L/2))/(n*tp_L)); \\ end$ 

295

```
for p=1:ns
  theta(1,p) = asin((3 + Am*sin((pi*(t(p) - tp R/2))/(n*tp R))...)
     -3 - \text{Am} \sin((\text{pi}^*(t(p) + tp_R/2))/(n)))/(-a(3) \sin(q0(3,1))+a(3) \sin(q0(3,2))));
end
for p=1:ns
  BETA SRF zs(p) = asin((Am*sin((pi*t(p))/(n*tp R))) - ...
     \operatorname{Am*sin}((\operatorname{pi*}(t(p) + tp R)))/(n*tp R)))/(-a(3)*\sin(q0(3,1))+a(3)*\sin(q0(3,2))));
  BETA SRR zs(p) = asin((Am*sin((pi*(t(p) - tp R)))/(n*tp R)) - ...
     \operatorname{Am*sin}((\operatorname{pi*t}(p))/(\operatorname{n*tp} R)))/(-a(3)*\sin(q0(3,1))+a(3)*\sin(q0(3,2))));
  BETA\_SLF\_zs(p) = asin((Am*sin((pi*t(p))/(n*tp\_L)) - ...
     Am*sin((pi*(t(p) + tp_L))/(n*tp_L)))/(a(3)*sin(q0(3,3))-a(3)*sin(q0(3,4))));
  BETA SLR zs(p) = asin((Am*sin((pi*(t(p) - tp L)))/(n*tp L)) - ...
     \operatorname{Am*sin}((\operatorname{pi*t}(p))/(\operatorname{n*tp} L)))/(a(3)*\sin(q0(3,3))-a(3)*\sin(q0(3,4))));
  BETA SRF ys(p) = 0;
  BETA SRR ys(p) = 0;
  BETA SLF ys(p) = 0;
  BETA SLR ys(p) = 0;
end
input_RF = LEG_RF;
input_RR = LEG_RR;
input_LF = LEG_LF;
input_LR = LEG_LR;
beta_SRF_zs = BETA_SRF_zs;
beta_SRR_zs = BETA_SRR_zs;
beta_SLF_zs = BETA_SLF_zs;
beta_SLR_zs = BETA_SLR_zs;
beta_SRF_ys = BETA_SRF_ys;
beta_SRR_ys = BETA_SRR_ys;
beta_SLF_ys = BETA_SLF_ys;
beta_SLR_ys = BETA_SLR_ys;
```

theta = zeros(1,ns);

## GG11.m

% % Random surface %

function [input\_RF, input\_RR, input\_LF, input\_LR, ... beta\_SRF\_zs, beta\_SRR\_zs, beta\_SLF\_zs, beta\_SLR\_zs,... beta\_SRF\_ys, beta\_SRR\_ys, beta\_SLF\_ys, beta\_SLR\_ys] = GG11(t, tdelay\_R, tdelay\_L, a, q0,... d 4RF, d 4RR, d 4LF, d 4LR),

ns = numcols(t);

tp\_R = round(tdelay\_R) %time delay b/w RF and RR tp\_L = round(tdelay\_L) %time delay b/w LF and LR

LEG\_RF = zeros(1,ns); LEG\_RR = zeros(1,ns); LEG\_LR = zeros(1,ns); LEG\_LF = zeros(1,ns);

BETA\_SRF\_zs = zeros(1, ns); BETA\_SRR\_zs = zeros(1, ns); BETA\_SLF\_zs = zeros(1, ns); BETA\_SLR\_zs = zeros(1, ns);

BETA\_SRF\_ys = zeros(1, ns); BETA\_SRR\_ys = zeros(1, ns); BETA\_SLF\_ys = zeros(1, ns); BETA\_SLR\_ys = zeros(1, ns);

- $$\begin{split} \text{LEG}_{\mathbf{R}F} &= [3.20, 3$$
- $$\begin{split} \text{LEG}_{RR} &= [3.20, 3.20$$

 $18.05, 18.10, 18.15, 18.20, 18.25, 18.30, 18.35, 18.40, 18.45, 18.50, 18.55, 18.60, 18.65, 18.70, \ldots \\18.75, 18.80, 18.85, 18.90, 19.0, 19.10, 19.20, 19.30, 19.40, 19.50, 19.60, 19.70, 19.80, 19.90, 20.0, 20.20, 20.40, 20.60, 20.80, 21.0, \ldots \\60, 20.80, 21.0, \ldots \\$ 

$$\begin{split} \text{LEG}\_LF &= [3.20, 3.20,$$

$$\begin{split} \text{LEG\_LR} &= [3.20, 3.20,$$

#### 60,20.80,21.0,...

for p=1:ns-1

$$\begin{split} &\text{BETA}\_\text{SRF}\_zs(p) = -asin((\text{LEG}\_\text{RF}(p+1)-\text{LEG}\_\text{RF}(p))/(d\_4\text{RF}(p+1)-d\_4\text{RF}(p)));\\ &\text{BETA}\_\text{SRR}\_zs(p) = -asin((\text{LEG}\_\text{RR}(p+1)-\text{LEG}\_\text{RR}(p))/(d\_4\text{LR}(p+1)-d\_4\text{LR}(p)));\\ &\text{BETA}\_\text{SLF}\_zs(p) = -asin((\text{LEG}\_\text{LF}(p+1)-\text{LEG}\_\text{LF}(p))/-(d\_4\text{LF}(p+1)-d\_4\text{LF}(p)));\\ &\text{BETA}\_\text{SLR}\_zs(p) = -asin((\text{LEG}\_\text{LR}(p+1)-\text{LEG}\_\text{LR}(p))/-(d\_4\text{LR}(p+1)-d\_4\text{LR}(p)));\\ \end{split}$$

 $\begin{array}{l} BETA\_SRF\_ys(p)=0;\\ BETA\_SRR\_ys(p)=0;\\ BETA\_SLF\_ys(p)=0;\\ BETA\_SLR\_ys(p)=0; \end{array}$ 

end

input\_RF = LEG\_RF; input\_RR = LEG\_RR; input\_LF = LEG\_LF; input\_LR = LEG\_LR; beta\_SRF\_zs = BETA\_SRF\_zs; beta\_SRR\_zs = BETA\_SRR\_zs; beta\_SLF\_zs = BETA\_SLF\_zs; beta\_SLR\_zs = BETA\_SLR\_zs; beta\_SRF\_ys = BETA\_SRF\_ys;

beta\_SRR\_ys = BETA\_SRR\_ys; beta\_SLF\_ys = BETA\_SLF\_ys; beta\_SLR\_ys = BETA\_SLR\_ys; beta\_SLR\_ys = BETA\_SLR\_ys;

## Rover\_1.m

q= Conf\_0;

 $DH_RF = DH(q(:,1));$  $q_RF = DH_RF(2:5,1);$ 

 $DH_RR = DH(q(:,2));$  $q_RR = DH_RR(2:5,1);$ 

 $DH_LF = DH(q(:,3));$  $q_LF = DH_LF(2:5,1);$ 

 $DH_LR = DH(q(:,4));$  $q_LR = DH_LR(2:5,1);$ 

 $q0 = [q\_RF, q\_RR, q\_LF, q\_LR];$ 

Dynamic\_Parameters = Dynamics(DH\_RF(2:5, 2), DH\_RF(2:5, 3));

% theta d a sigma m rx ry rz Ixx Iyy Izz Ixy Iyz Ixz dh dyn = [DH RF, Dynamic Parameters];

%

 $q = Conf_1;$ 

DH\_RF = DH(q(:,1)); q\_RF = DH\_RF(2:5,1);

DH\_RR = DH(q(:,2)); q\_RR = DH\_RR(2:5,1);

 $DH_LF = DH(q(:,3));$  $q_LF = DH_LF(2:5,1);$ 

 $DH_LR = DH(q(:,4));$  $q_LR = DH_LR(2:5,1);$   $q1 = [q_RF, q_RR, q_LF, q_LR];$ 

%

# Conf\_0.m

function Configuration = Conf 0()

%------ Right Side -----theta 1R = 0;theta 2R = 0;theta 3RF = -pi/4;theta 3RR = pi/4;theta 4R = 0; %------ Left Side -----theta\_1L = 0; theta2L = 0;theta\_3LF = pi/4; theta\_3LR = -pi/4; theta\_4L = 0; %-----------% theta\_1, theta3, theta 4 theta\_2, Configuration\_ $RF = [theta_1R]$ theta\_2R theta\_3RF theta\_4R]'; Configuration\_ $RR = [theta_1R]$ theta\_2R theta\_3RR theta\_4R]'; Configuration\_LF = [theta\_1L] theta\_2L theta\_3LF theta\_4L]'; Configuration\_LR = [theta\_1L] theta\_2L theta\_3LR theta\_4L]'; RF Leg % RR Leg, LF Leg, LR Leg Configuration\_RF Configuration\_RR Configuration\_LF Configuration\_LR];