

Development and Testing of the Mars Rover Mobility Platform for Educational and Research Purposes

Luiz Filipe Barbosa¹, Jevgenijs Trunins², Yahya H Zweiri³, Malcolm Claus⁴, Niklaus Kamm⁵

School of Mechanical and Automotive Engineering, Kingston University, London, UK^{1,2,3}

School of Aerospace Engineering, Kingston University, London, UK^{4,5}

Abstract: Mars exploration has a long history, but there were only four roving vehicles which successfully operated on its surface (e.g. [1]). Main reasons for this are the mission cost and complexity. This paper describes a Mars Rover Mobility Platform for educational and research purposes developed at Kingston University of London. This platform utilises off-the-shelf components to minimise the cost of the project, and is designed to allow for future improvement. The rover is targeted to meet university research and educational objectives. This paper describes the design, manufacturing and control system of a robotic vehicle. The emphasis of this paper is the implementation of the control system. The investigation in locomotive sub-system and its traction performance was done [4]. The rover was manufactured in-house and its manufacturing method and its main components will be described. The control of the vehicle was done using python programming language and implemented on Raspberry Pi 2B+ controller. The communication was done via Wi-Fi using socket connection stream to identify the TCP/IP of the server and connect to the client. Finally, the testing operation was conducted by producing a qualitative comparison between the actual performance and the specified requirements. The rover design reported here achieved climbing capability for the slopes of 23°, the turning radius of zero degrees. The final mass of the rover is 18 kg including allowance for the payload. The rover is able to reach a velocity of 5 cm/s.

Keywords: Mars rover, Space exploration, Robotic control, Educational platform.

I. INTRODUCTION

The aim of this project was to develop and build a Mars Rover Mobility Platform (e.g. Fig. 1) for scientific research and educational purposes. The rover will be used to conduct research in extra-terrestrial robotics with a focus on automation of the platform. Also, it is aimed to be part of educational outreach and to be used as educational platform for university students. The design incorporates simple rocker-bogie configuration with six independently driven wheels (e.g. Fig. 1) as a part of locomotive sub-system. The design of the vehicle was created for the exploration of rough terrain therefore the control system was designed to fulfil this purpose.

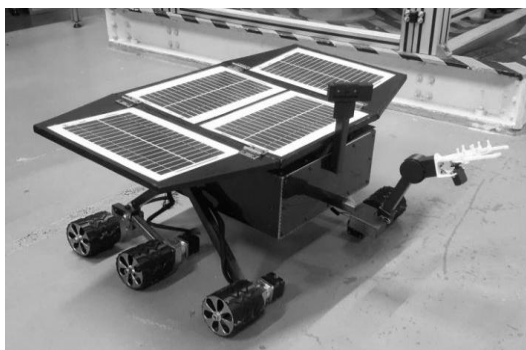


Fig1 Mars Rover Mobility Platform fully assembled

The rover is controlled remotely with a future vision of simulating real delayed radio response.

A. Background

Space exploration has been underway for more than 58

years starting with a launch of “Sputnik-1” satellite. It has continuously provided humankind with knowledge not only about space environment but also about our planet, Earth. The first successful Mars exploration rover landed in 1996. This started development of a robotic vehicle by many other space agencies and research organisations around the world. In the last decade the chances has increased that in near future humans will have to find the means to live beyond Earth, due to the dramatic increase in human population alongside with extensive use of Earth resources. Some scientists believe that Mars, being close to us, is our most viable option. As envisioned by NASA, the main objectives of exploring the surfaces of Mars are (e.g. [2]):

- Determine plausibility and feasibility of life in Mars
- Analyse Mars Climate
- Analyse Mars Geology
- Preparation for human exploration

Even though, a rover based mission presents a high cost and long lead time for its development, they also present the possibility of a wider area for exploration, identifying the resources necessary for future human missions and achieving the above goals (e.g. [2]). The control of the rover from Earth does present a number of difficulties. The transmission of a signal takes an average of 7 to 46 minutes (e.g. [3]) due to the great distance from Mars which creates a practical problem of obstacle avoidance. The rover wheels could become stuck in loose sand, or even worse tumble over a large obstacle. Therefore, the

design consideration of the rover includes high traversability in uneven, rough terrain without losing stability.

The selected design was chosen based on previous research (e.g. [4]) which adopts the rocker-bogie mechanism.

B. Aims and the Problem Statement

Main focus of the project was to develop and build Mars Rover Mobility System for academic research, education and outreach application. Therefore it will have to be reliable, easy to program and easy to control. In addition, the rover was planned as a testing platform, which means the design should allow for the implementation of design changes in case of future developments.

The rover wheels were required to continuously maintain the contact with a soil while the robot is in motion through uneven terrain. The vehicle is able to overcome longitudinal and lateral slopes of 20-25 degrees on different types of soil (associated with Mars soil properties). The rover is able to turn around its centre of gravity point using torque steering.

Specifically in early development stage was agreed that the rover should be controlled by an operator using hand controller.

C. Objectives and Requirements Specification

This part identifies the objectives of the project at the development stage. The initial design considerations were divided into five categories described by [4], in order to make the selection of design and components to be used on each sub-system. The requirement specification was used to create constraints that the project would have to follow. These constraints describes specification to the mechanical design, software design, components selection and testing operations. For the current part of project the following were used:

- The rover should be self-autonomous as much as possible
- The design should be as simple as possible
- The software should have the capacity to be reprogrammed during the mission
- The rover should operate on slopes of up to 20 degrees
- The rover should be able to traverse obstacles of up to 1.5 wheel diameter
- The mass of the rover should be kept below 20 kg
- The maximum length of the rover should not be greater than 60 cm and width no more than 45 cm
- The bus ground clearance should be more than 12 cm
- The minimum turning radius should be 0 m (tank turning)
- Batteries should be recharged using solar panels
- Data storage should keep all generated information while operating the rover. Multitasking capability should be available
- Rover should be able to communicate via wireless transmission

The objectives for the following development part were stated as:

- Redesign and build tractive and stability systems based on specification outlined

- Individual components selection and its description for their future integration
- Description of final design including any alteration after testing
- Development of remote control strategy based on communication via Wi-Fi between two Raspberry Pis using Python code
- Implement the remote control strategy for the tractive system control
- Steering strategy to derive a formula that will assign the reference signals to the tractive system
- Kinematic analysis of angular position of the robotic arm
- Implement remote control strategy for the robotic arm and camera
- Development of stability control
- Set the testing procedure to validate rover operation
- Analyse results based on requirement specification
- Discuss project difficulties and future implementation

II. STRUCTURAL DESIGN OF THE ROVER

The structural design of the rover vehicle will be described in this section and is divided into four main sub-parts:

- Design of the suspension sub-system
- Design of the tractive sub-system
- Design of the bus structure
- Design of the robotic arm and camera mast

A. Mobility sub-system design

The design process was conducted by analysing and comparing different mobility systems designs (e.g. [4]). The selected suspension design was the rocker bogie mechanism. The selection method of traction was conducted based on a performance study of different tractive methods using rocker bogie suspension design (e.g. Fig. 2). The performance of 2-axles, 3-axles and 4-axles vehicles was analysed by the study performed by [5]. The maximum obstacle angle with zero ground inclination which vehicle would be able to traverse is 52 degrees when the bogie is places on the front axle of the suspension system. In comparison, the maximum traversable angle that the vehicle could produce while having the bogie in the rear is 68 degrees.

The final design decision was to select the 3-axels and with the bogie on the rear

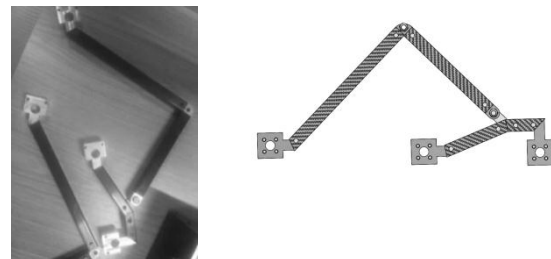


Fig. 2 CAD model of final design and assembled model

B. Wheels design and manufacturing

The wheel size study was based on drawbar pull which is by definition is the difference between available tractive and resisting forces. The selection was based on the

consideration the smallest wheel diameter with a drawbar pull to weight ratio of 20% (e.g. [4]). The initial constrains of wheels size were given as:

- Wheels diameter: 80 – 200 mm
- Wheels width: 60 – 160 mm

The soil properties values were acquired from actual Mars mission (e.g. [4]): Viking Landing 1 and 2 (VL-1 and VL-2), Mars Exploration Rover (MER-B) and soil simulant developed by German Aerospace Center (DLR-A and DLR-B). The calculation of the resistive force was conducted by adding three main resistances to motion:

- Compaction resistance
- Gravitational resistance
- Bulldozing resistance

The Fig 3 below shows the relation of drawbar pull for wheels of different diameter and width.

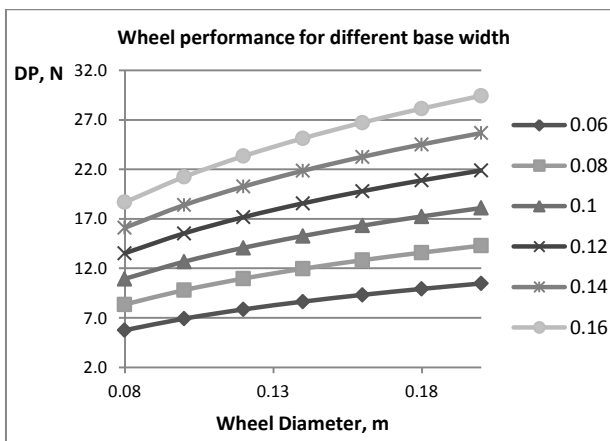


Fig. 3 Wheel performance comparison graph (e.g. [4])

The graph of drawbar pull (DP) versus wheel diameter shows the six different curves where each of them represents a wheel width. The wheel width produced approximately 20% ratio of DP over weight of rover with smallest wheel diameter was with 80mm width with 80mm diameter.

The manufacturing method chosen was to 3-D print (e.g. Fig. 4) the wheel profile and CNC profile supports. The reason for the selection of 3-D printing was due to cost and time constrains. Also the 3-D printing allows for easy interchangeable wheel profiles, for future profile performance investigation.

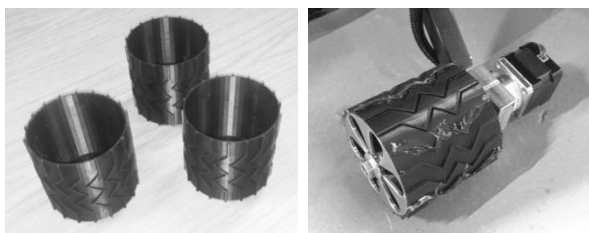


Fig. 4 3-D printed wheels and fully assembled wheel

The profile supports were manufactured using 3mm aluminium sheets. The cutting of excess material was done using CNC machine. All shafts used in this project were manufactured using mild steel. The reason for the

selection of material was due to the fact that steel has a better life span with regards fatigue than aluminium therefore helping to prevent plastic deformation of the shafts.

C. Suspension Design and Manufacturing

Based on research studies and CAD simulations conducted by [4] some design considerations were taken in order to increase the climbing capability (e.g. Fig. 5). The positioning of the middle wheel should be vertically collinear to the centre of gravity. Also the bogie link should be connected at a 45 degree angle to the wheel. An additional beam was included in the front bogie to increase traversability.

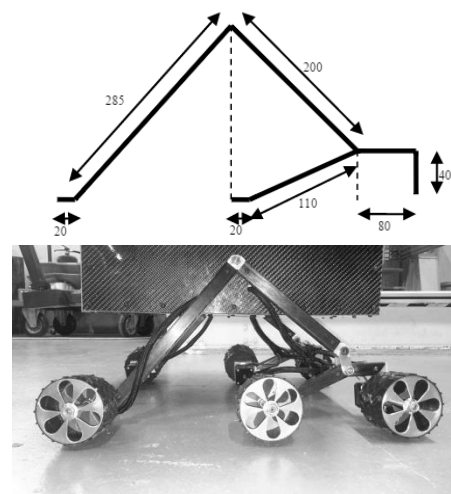


Fig. 5 Rocker bogie configuration

By doing so the beam would provide additional climbing moment since the orientation of the force pushing the wheel down does not go through the wheels centre point. It should also be mentioned that the link length should not exceed half of the wheel radius of the wheels.

The manufacturing suspension was done using off-the-shelf carbon fibre hollow square profiles with aluminium inserts to connect them (e.g. Fig. 2). The reason for selection of these materials was to maintain the low weight of the structure. The square tubes used for the bogie were cut to dimension. The shafts used for the suspension (rocker bogie) were made of mild steel.

D. Design of the Bus Structure

The initial bus structure design was done by [4].The design was updated to incorporate the off-the-shelf components (e.g. Fig. 6).



Fig. 6 Bus structure assembly

Bus structure includes 20 by 20 mm Bosch Rexroth strut profiles as a supporting structure and carbon fibre 2 mm thick sheets. The acrylic sheet of 2.5 mm was introduced at the base to be used as a payload support. The support base was perforated to easier the mounting of components.

E. Design of the Robotic Arm and Camera Mast

Camera mast and robotic arm were built using the carbon fibre hollow square rods, with the 3-D printed joints. The servos were installed inside the joints.

The selected camera for the operation is Raspberry Pi camera module.

III. SELECTION OF ELECTRICAL AND ELECTRONIC COMPONENTS

The selection of the components to be used in this project was done based on power budget and cost budget. The main electrical/electronic components are:

- Computer (main controller)
- Stepper motors
- Servo motors
- Servo controller
- Stepper controllers

A. Computer selection

The selection of the computer which is used to control all subsystems was conducted by evaluating the requirement specification mentioned in Section I. Two options of controllers were analysed: Raspberry Pi 2B+ and the Arduino Due. The reason for selecting these two was due to low cost and available tutorials, forums and libraries online, as the projects tend to be for educational purpose. Also both controllers would have to be able to control stepper motors and servo motors using external drive board. The first requirement mentioned was that the software should have a capability to be re-programmed remotely during operation. This will also allow the project to run with minimum intervention to hardware. Therefore, the controller would have the capability to edit the control scripts remotely using Wi-Fi. To allow for code alteration was chosen the secured socket shell (SSH) connection. This method allows the users to edit and save scripts when both the host and target are connected to the same network.

The next requirement analysed was multitasking capability. The main decision parameter was how well the controller can perform multitasking. Therefore, the controllers would have to be capable of running multi-threaded scripts using multiple processing units.

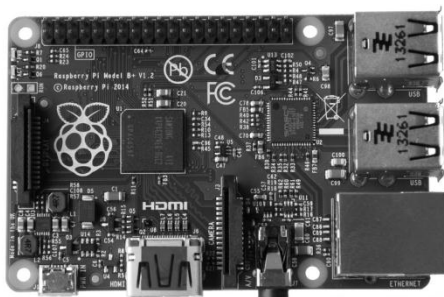


Fig. 7 Raspberry Pi 2B+ (courtesy of Raspberry Pi)

The Arduino Due is the most powerful Arduino available. It has 96 KB of RAM memory and runs at 84 MHz while the Raspberry Pi 2 has 1GB of RAM memory and runs at 900 MHz Also the Raspberry Pi 2 has 4 processing units allowing multi-threading more friendlier. By using multi-threading capability the Raspberry Pi code execution increased 7 times. For the reasons mentioned above the selected controller was the Raspberry Pi 2B+ (e.g. Fig. 7).

B. Stepper Motors Selection

The stepper motor was the selected type of motors to be used for the tractive system of the rover. The reason for selecting this type is the simplicity of open-loop control without the necessity of feedback sensors. This does lead to errors due to not producing enough torque (which should happen in exceptional circumstances), but at the same time is much cheaper option.

The selection of the stepper motors was based on calculations to find out the required torque that the motors can produce. The requirement of velocity was based on a typical velocity produced by a mars rover vehicle which in this case used the operational speed of the MER-B Curiosity of 5cm/s (e.g. [6]). The other requirement that needed to be specified was if the rover would have enough torque to overcome slopes of required angle. The calculations were based on the resisting forces acting on the rover for the worst case scenario. The following initial parameter were assumed or measured:

- The coefficient of rolling resistance for soft soil: $\mu \approx 0.23$ (e.g. [7])
- Vehicle mass with a payload: $M_v=18\text{kg}$
- Wheel radius: $r=40\text{ mm}$
- Maximum acceleration of the rover: $a = 0.25\text{ m/s}^2$
- Acceleration due to gravity: $g=9.81\text{ m/s}^2$ (was used Earth gravity as the test environment)
- Maximum slope angle: $\theta=25^\circ$
- Maximum vehicle velocity: $v = 5\text{ cm/s}$

The equation that relates all resistance forces when producing motion is:

$$F_r = R_r + R_i + R_g$$

Where rolling resistance, inertial resistance and gravitational resistance respectively are:

$$R_r = \mu M_v \cos \theta$$

$$R_i = a \frac{M_v}{g}$$

$$R_g = M_v \sin \theta$$

The overall resistance force that the vehicle should overcome would per motor is given:

$$F_r = F_T/6$$

Therefore, the minimum torque (T_R) that each motor would have to overcome is:

$$T_R = F_r r$$

The calculation for the angular velocity (ω) was conducted assuming the maximum linear velocity produced by the vehicle, with no slip:

$$\omega = v/r$$

The selected motor for the tractive would have to be able to produce the required torque and speed while complying with the current demand of the driver board. The driver board available to be used with the Raspberry Pi 2B+ to

drive the steppers were the Adafruit DC/Stepper motor HAT (e.g. Fig. 8).

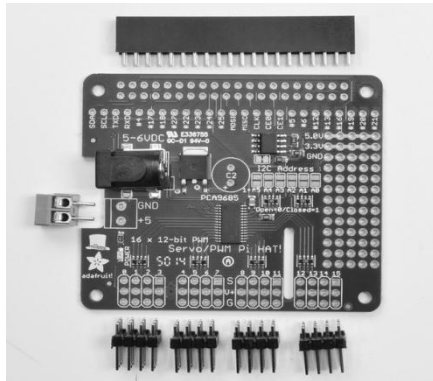


Fig. 8 Adafruit DC / Stepper motor HAT (courtesy of Adafruit)

Each motor HAT has the capability to drive two stepper motors therefore 3 HATs were used for the tractive system. The operating current that the HAT could produce was 1.2 - 2 Amps per pole with a peak current of 3 Amps. The motors operating current and peak current specs would have to comply with those parameters to allow optimum performance. The selected stepper motor is widely manufactured 2-phase NEMA 17 (e.g. Fig. 9).



Fig. 9 NEMA 17 Motor with 5-to-1 gear ratio installed

This motor can produce up to 0.63 Nm torque (T_M) at rated 2 amps using double coil excitation. Therefore, the implementation of a gear box was necessary in order to comply with the required torque previously calculated.

The gear ratio was calculated using:

$$T_R = T_M \eta \epsilon$$

Where η – efficiency of gear ratio and ϵ – gear ratio

The reduction gear selected followed a ratio of 5:1.

C. Servo Motors Selection

Two servo motors would be placed in each of the two joints of the robotic arm (e.g. Fig. 10). Another servo would be placed in the claw bracket (e.g. Fig. 10) to allow it to open and close.

Finally, two other servos were placed on the camera mast (e.g. Fig. 10) to allow pitch angle control and yaw angle control. The next step was to select the drive board that would be used to produce the PWM variable voltage input

to control the servos position based on digital signal from Raspberry Pi.

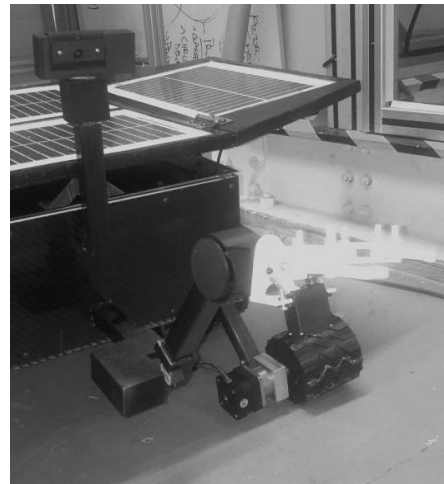


Fig. 10 Robotic arm and Camera mast

IV. SOFTWARE CONTROL

The scripts used to control the stepper motors of the tractive system and the servo motors of the robotic arm and camera mast was developed using python programming language. The scripts were implemented directly in the Raspberry Pi and due to its capability of multi-threading allowing running the programmes simultaneously.

A. Tractive System Control

The control of the tractive system was done using two Raspberry Pis that can communicate between themselves using TCP/IP communication via sockets. The first one would collect input from a hand controller with two analogue variable resistors and a press switch (e.g. Fig. 11).

The signal is then sent to an external analogue-to-digital-converter soldered on a “protoperma” board. The digital signals were produced based on the user input on the Y-axis to command the rover to move forward or backward and on the X-axis to command the rover to steer left and right. After that the signal calibration was done, in order to allow equal range of signal and to differ the signals from right and left to be negative and positive, respectively.



Fig. 11 Hand controller

The assignment of the left and right speed was based on equations:

$$\begin{aligned} speedRight &= 'poty' + \frac{'potx'}{2} \\ speedLeft &= 'poty' + 'potx/2' \end{aligned}$$

Where: $poty$ – Calibrated y-axis input from hand controller
 $potx$ – Calibrated x-axis input from hand controller

The data transmission was done using a shared Wi-Fi network. The socket connection works by combining all data into a long sting composed by multiple inputs then sending the whole string which produced fragmentation of data. The sending command was placed in a while loop. The reception of data was done using a client script. This script is saved in the second Raspberry Pi placed inside the rover. This script uses three independent threads. All threads run simultaneously. Therefore, one thread collects the input sent by the remote control Raspberry Pi and saves the value into global variables. Then the other two threads are used to assign the speed to the three motors in the left and the three motors in the right.

B. Robotic Arm and Camera Mast Control

The servo motors control uses input from the keyboard via SSH connection. The client script assigns an initial value to both servos placed in the joints of the arm and to the both servos placed in the pitch and yaw joint of the camera. This causes the arm and the camera joints to go to its mid position. By pressing the control keys the angular steps are added or subtracted from the initial starting position of the servos.

V. RESULTS

The validation of the rover performance was conducted by analysing the measured parameters, and then compared to requirement specifications (mentioned in Section I). At this stage of the project the rover autonomy is still under development. The main scope of the project was to create a solid platform for future development. The design of all sub-system was kept as simple as possible. As mentioned on the software implementation, the controller placed in the rover can be edited via SSH therefore allowing re-programming during operation. A travesibility test was conducted. The testing track had an initial ascending and descending slope of 23° . Also, the performance of step climbing capability (only 50% of the wheel diameter) was checked.

The rover performed very well during the ascending and descending of a slope, producing a continuous climb until the peak. Even though some slip was observed the rover managed to pass the peak. The step test presented a struggle by the wheel to produce grip since the step presented very little friction with the plastic wheels. The steering test was conducted where turning radius of 0° (tank steer) was used with good performance results.

The mass of the rover was kept at $\sim 18\text{Kg}$. The dimensions also did not require any changes and fulfilled the requirements. The solar panels were installed but full capability was not yet investigated. The Raspberry Pi does have the capacity of gathering and storing data while running the control task simultaneously although it has not yet been test to its full capability. A reset option can be implemented since the user can access the rover Raspberry Pi via SSH and re-download the original source script from the internet.

It is also considered in a future to implement the adapting stability system in order to maintain continues traction while traversing the terrain.

ACKNOWLEDGMENT

The authors would like to thank the team of technicians working for Kingston University for helping and instruction during the manufacturing stages. And finally, we would like to say thank you for Kingston University for financial support in this project.

REFERENCES

- [1] SPACE.com. (2011) Mars Explored: Landers and Rovers Since 1971 (Infographic), [Online]. Available: <http://www.space.com/12404-mars-explored-landers-rovers-1971.html>
- [2] National Aeronautical and Space Administration. (2015) MARS 2020, NASA Facts, [Online]. Available: http://mars.nasa.gov/mars2020/files/mars2020/Mars2020_Fact_Sheet.pdf
- [3] S. Gerard. (2012) Data communication and Mars missions. 12th European Mars Conference, EMC12, Munich. [Online]. Available: <http://www.marssociety-europa.eu>
- [4] J. Trunins, A. Curley and B.Osborne, "Design of a Mars Rover Mobility System", JBIS, vol.65, pp.87-97, 2012
- [5] X.C. Potau, M. Comellas, M. Nogues and J. Roca, "Comparison of different bogie configurations for a vehicle operating rough terrain", Journal of Terramechanics, vol. 48, issue 1, pp. 75-84, Feb 2011
- [6] Jet Propulsion Laboratory. (2015) Spacecraft: Surface Operation: Rover, Rover speed. [Online]. Available: http://mars.nasa.gov/mer/mission/spacecraft_rover_wheels.html
- [7] J.Y. Wong, Terramechanics and Off-Road Vehicle Engineering, 2nd ed., Ed.: Elsevier, 2009.