www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 16(3), May-Aug 2025, 229-253. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2025.v16i3.609

Path Planning, Control and Optimization for Differential Drive Space Exploration Rover (DDSER)

Muhammed Mirac Özer [1]
[1] Unmanned/Intelligent Systems Lab, Faculty of Aeronautics and Astronautics, Tarsus University, Mersin 33400, Turkey
muhammed_mirac@tarsus.edu.tr

Abstract. Space exploration rovers, one of the most important tools in today's exploration studies, are designed to obtain scientific data by examining the surfaces of different planets and to seek answers to questions of great importance to humanity. In this study, a controller with nonlinear governing equations and path planning are carried out on advanced avionic systems for difficult conditions in order to autonomously select the shortest path at a desired trajectory and speed for a differentially driven space exploration rover. While the rover leaves its traces on the difficult surfaces of different planets, it undertakes many important tasks from the analysis of geological formations to the potential traces of life. Differential drive allows these rovers to move flawlessly even on rough surfaces without making sharp turns. It is observed that the RRT algorithm can be used in the implementation of path planning, unlike other path planning algorithms, to reach the target in a shorter time. This opens the door to new scientific understandings by carrying space exploration to previously inaccessible areas.

Keywords: Space Exploration Rover, Differential Drive, Autonomous Systems, Robotics, Control Systems, Path Planning, Optimization.

Article Info
Received January 27, 2025
Accepted March 18, 2025

1 Nomenclature

x_b	: Body Frame Longitudinal Coordinate Axis
y_b	: Body Frame Lateral Coordinate Axis
x_e	: Earth Frame x Axis
y_e	: Earth Frame y Axis
w	: Body Width from Body Frame
c_1	: Contact Point of Right Tire with the Ground
c_2	: Contact Point of Left Tire with the Ground
h_1	: Center of Right Tire
h_1	: Center of Left Tire
$\vec{V}_{c_1}$	: Right Tire Contact Point Velocity with the Ground
$\vec{V}_{c_2}$	: Left Tire Contact Point Velocity with the Ground
$\vec{V}_{h_1}$	: Velocity of Center of Right Tire
$\vec{V}_{h_2}$	: Velocity of Center of Left Tire
$\vec{V}_{b_1}$	: Velocity of Center of Body for Right Wheel
$\vec{V}_{b_2}$	: Velocity of Center of Body for Left Wheel
$\vec{V}_r$	: Velocity of the Center of the Body Frame
$\vec{\omega}$	: Angular Velocity
$\vec{\omega}_{b_1}$	: Angular Velocity Center of Body for Right Wheel
$\vec{\omega}_{b_2}$	: Angular Velocity Center of Body for Left Wheel
$\vec{\omega}_r$	: Angular Velocity of the Center of the Body Frame
$\vec{r}_{h_1}$	: Radius of Right Wheel Center w.r.t Center of Mass
x_d	: Desire Longitudinal Coordinate Axis
y_d	: Desire Lateral Coordinate Axis
$\dot{x}_d = V_x$	: Desire x Derivative of Axis with Respect to Time
$\dot{y}_d = V_y$	: Desire y Derivative of Axis with Respect to Time
r	: Radius of the Two Wheels or the Two Tires
$\vec{T}$	: Torque Vector
$R_{Z,\theta}$	: Rotation Matrix at z-axis with θ Angle
q	: Control Signal (Generalized coordinate Vector)
$\vec{\dot{q}}$	: Contraints Forces or Moments
$\ddot{q}$	: Control Acceleration

$\ddot{q}_d$	: Feed Forward (Acceleration of the Joint Desired)
q_d	: Desire Control Signal
q_e	: Control Signal Error
M	: Acceleration (Matrix)
B	: Any Centripetal veya Coriolis Accelarations
$C^T\lambda$	: Term due to the Constraints of the Rover
C	: Faffian Restriction Term (Matrix)
$C(q,\dot{q})\dot{q}$	: Friction Term
λ	: Lagrangian Multiplier
τ	: Input to the System (Input Torque)
τ_1	: Torque Applied to Right Wheel
τ_2	: Torque Applied to Left Wheel
m_b	: Mass of Rover Body
m_w	: Mass of Rover Wheel
m_t	: Mass of Total Rover
$2w^*$	: Optimum Wheel-to-Wheel Distance
t	: Thickness of Tire
V_d	: Speed of the Rover along the Path
$y_d = x_d^2$	: Desired Trajectory
a_q	: Desire Acceleration (Outer Loop)
u	: Inner Loop (Control Signal)
K_p	: PID Control Proportional Control
$K_ı$	: PID Control Integral Control
K_d	: PID Control Derivative Control
ω_i	: Natural Frequency
$\dot{\theta}$	: Angle Change Rate
$\dot{\theta}_d$	: Desire Angle Change Rate
$\dot{\phi}_1$	: Rate of Change or Angular Velocity of Right Tire
$\dot{\phi}_2$	: Rate of Change or Angular Velocity of Left Tire
$\dot{\phi}_{d,1}$	: Angular Velocity of Right Tire
$\dot{\phi}_{d,2}$	: Angular Velocity of Left Tire
I_b	: Moment of Inertia Matrix of Body
I_w	: Moment of Inertia Matrix of Wheels
I_t	: Moment of Inertia Matrix of Total Rover
X	: End-Effector Pose using the Minimal Orientation
$\tilde{X}$	: Error
J_a	: Analytical Jacobian
d	: End Effector Position
α	: End Effector Orientation
$\dot{\vec{X}}$	: End Effector

2 Introduction

Human curiosity and scientific discovery desire directed to the depths of space exploration missions cause the rapid development and evolution of space exploration technologies [1,2]. Rovers are remotely operated vehicles designed for space exploration and scientific research [3]. These unique technologies offer many exciting opportunities to unravel the mysteries of space and to delve into the depths of planets. The main purpose of rovers is to enable the exploration of distant planets that are challenging and dangerous for humans, to collect scientific data and to search for traces of extraterrestrial life with new findings [4]. The areas of use of rovers are quite wide [5]. First of all, they are mounted on spacecraft sent to examine the surfaces of Mars and other planets. They are used to map the surfaces of these planets, examine geological features, collect samples and potentially search for signs of life [6-8]. In addition, rovers can examine the surfaces of the Moon, Mars and other celestial bodies in order to discover extraterrestrial resources [9]. These resources are important for supporting the sustainability of future space missions. The features of rovers are carefully designed to facilitate their work on distant planets. These vehicles, usually equipped with wheels, allow them to move easily on the surface [10]. In addition, they make long-term missions possible due to their energy sources such as solar panels or radioisotope thermoelectric generators. Equipped with scientific research equipment such as forward-looking cameras, spectrometers and probes, rovers have the capacity to solve the mysteries of planets and celestial bodies [11-13]. In this context, rovers represent an important part of space exploration and offer great potential for future scientific research [14]. Thanks to these tools, the mysteries of distant planets and celestial bodies can be solved [15]. In addition, traces of extraterrestrial life can be investigated and humanity can have the opportunity to look deeper into the universe.

An important component for the successful operation of these vehicles is an effective controller design. The controller includes a set of algorithms, software, and hardware components required to regulate the rover's movements, communication, and missions. First, the controller ensures that the rover moves safely on the surface of the planet or other space environments. The ability to adapt to the variable conditions on the surfaces of planets such as Mars emphasizes the importance of controller design [16]. It also helps the rover to avoid collisions and overcome harmful obstacles by sensing its environment. The controller also helps the rover to effectively perform its scientific tasks. When precise measurements are required during scientific research, the controller ensures that the vehicle remains in a fixed position or approaches a specific target with precision. This enables the successful implementation of sampling, analysis, and other scientific studies. The successful implementation of the controller design affects the success of scientific discoveries and space exploration.

The latest stop on this exciting journey is a magnificent combination of differential drive mobile robots and space exploration rovers. Rovers are autonomous systems where wheels are creatively steered, aiming to unravel the mysteries behind geological folds [17]. One of the important milestones of this evolution is that differential drive space exploration rovers are becoming an important component of interplanetary research in order to effectively study the surfaces of distant planets and make scientific discoveries [18]. Space exploration rovers are defined as autonomous robotic vehicles designed to effectively study the surfaces of distant and harsh planets [19]. Differential drive rovers, which are usually wheeled, use different drive systems while traveling on planetary surfaces [20]. At this point, the differential drive system operates using a basic principle that allows rovers to move in a balanced manner on rough surfaces without making sharp turns. The differential drive system allows the wheels on both sides of the rover to be controlled independently. By adjusting the speed of both wheels differently, the rover can be directed in the desired direction. In this way, the rover can move forward in a balanced manner on rough and sloping surfaces [21]. The differential drive system also provides energy efficiency and is an ideal option for rovers that offer mechanical simplicity and long life and durability. In this context, the differential drive system enables rovers to move forward in a balanced manner on complex surfaces, increasing their capacity to expand the exploration area and reach previously inaccessible areas. In addition to differential driving, intelligent path planning strategies are also important for the successful operation of space exploration rovers [22-24]. In this sense, path planning provides a process used to determine how the rover will reach a given destination in the most efficient and safe way [25, 26]. A path plan is created by considering the spacecraft's location, target, obstacles and road conditions [27-29]. This involves a combination of algorithm-based and data-driven approaches [30]. Techniques such as local mapping, image processing and sensor data analysis help the rover understand its environment and enable it to choose a safe path [31, 32].

As a result, space exploration rovers can be defined as vehicles with great potential to explore the surfaces of different planets and collect scientific data [33, 34]. Differential drive systems help these rovers to move effectively even in difficult conditions, while intelligent path planning strategies are also an important solution to ensure safe and efficient exploration. In the future, further development and improvement of these technologies will enable space exploration to be in-depth and comprehensive. In the next section of this study, since there is no previous study in the literature directly addressing the avionics system for differential drive rovers and the controller and path planning required for this, the path planning algorithms implemented on rovers will be reviewed and the benefits of the study will be discussed. In the fourth section, the avionics system hardware parts of the rover will be discussed in detail. In the fifth section, the mathematical equations that will provide differential drive will be detailed and the controller will be designed via MATLAB Simulink. Then, the results and benefits of the system will be discussed. In the sixth section, the details of the RRT algorithm used for path planning are given and the rover, whose mathematical equations are implemented, is provided to perform autonomous path traversal. In the seventh section, the conclusion section regarding the study is given and the process implemented regarding the study is discussed and the study will be completed.

3 Related Work

In [35], an improved A* algorithm is presented for optimizing the exploration of space rovers. The aim is to develop an A* algorithm that takes into account the rover environmental factors (such as surface slope and roughness) and the mobility of the rover. In particular, this algorithm has the potential to plan shorter paths and achieve higher success rates compared to the original A* algorithm, as obtained from numerical simulations performed on the MATLAB platform. In [36], an important step is represented in the dynamic modeling and path planning of a wheeled lunar rover that can jump. This rover can move on flat surfaces using its wheels, while it is equipped with a unique ability to overcome obstacles in complex and obstacle-filled areas thanks to its inertial wheel. First, the researchers examined the kinematic and dynamic modeling of the rover in detail, documented the lunar surface at the time of the jump with a camera, and after converting this data into a grid map, the starting point of each step was determined using the Q-Learning method. In [37], we focus on the development of traversability analysis and path planning algorithms for extreme-terrain rappelling rovers to address the challenging terrains of planets. Although the rappelling rovers used in this study can safely traverse steep surfaces, navigating such terrains is complex. This study presents new traversability analyses and path planning algorithms for rovers to safely navigate such challenging terrains. It also addresses the unique stability and accessibility challenges of rappelling systems.

In [38], we present a new method based on Bi-RRT (Bidirectional Rapidly Exploring Random Tree) algorithm, APF (Artificial Potential Field) algorithm, and DOM (Digital Orthophoto Map) to address the path planning problem of rovers on the lunar surface. This study first constructs an occupancy grid map using DOM, then designs an obstacle-based repulsive potential field and a target-based attractive potential field using this map. Then, it uses a new potential area guidance, growth method and target biased sampling method with an improved Bi-RRT algorithm. [39] makes a significant contribution by addressing the path planning problems of lunar rovers. Generally, lunar rover path planning algorithms have problems such as slow convergence rate and falling to local optimal solution. This study proposes a comprehensive genetic algorithm based on virtual three-dimensional modeling to solve these problems. The terrain comprehensive cost function is used to determine the fitness function in adjusting the genetic factors. [40] considers the autonomous path planning technology of Mars Rover on the surface of Mars under the limited computing resources. This research aims to make the map model more suitable by combining the visibility-graph method and grid method and optimizing the grid partitioning method of the combined method. Basically, it provides a direction search method that can use the computational resources efficiently. In addition, the visibility-graph map direction search method shows that it has less complexity in terms of computational time and can achieve better results by performing a synchronous search in the path planning process. In [41], autonomous path planning and dynamic obstacle avoidance operations in dynamic environments with preliminary map information were combined using RRT* and dynamic window approach. During the implementation of this method, simulation and physical verification were carried out on the ROS development platform and the Jackal unmanned vehicle platform.

Up to this point, the studies have included rover applications of different path planning algorithms. The following studies include path planning applications with the help of different sensor systems. In this context, in [42], a CG-Space-based dynamic path planning and obstacle avoidance algorithm is presented for a 10 DOF wheeled rover moving on 3D rough terrain. The study proposes a dynamically reconfigurable tree structure using a customized RRT* algorithm based on CG-Space. In this way, the rover can replan its path in real time and reach its destination when it encounters fixed and randomly moving obstacles. As an example of a different sensor usage, in [43], it includes simultaneous location detection and map creation processes using sensors such as Lidar. The map of the lunar surface is defined as a two-dimensional grid and this grid map is processed using long-short-term memory (LSTM). Then, the rover is trained using multi-shot learning with a deep Q-network (duel DQN) for automatic path planning. The results obtained in this study show that this method is effective and adaptable for rover vehicles.

This article also investigates how experiences and achievements from previous space exploration missions are used in the design and implementation of differential drive space exploration rovers. It also examines how new generation sensors, image processing algorithms and artificial intelligence techniques contribute to the development of this field. This study stands out as an important step towards expanding the boundaries of space exploration technologies and gaining deeper insights for humanity. The fact that differential drive space exploration rovers provide access to previously unreached areas and that intelligent path planning strategies increase efficiency emerge as an exciting potential study in determining the future directions of space exploration.

4 Avionics System

In this section, the design details of the differential drive space exploration rover used for space exploration purposes by communicating in the S-Band (frequencies between 2 GHz and 4 GHz), HF (Ultra High Frequency) and VHF (Very High Frequency) frequency bands are given. The brain of the designed avionics system is provided by a computer on the robot. The control information sent to the robot by the ground operator is processed by the control software developed by us using the java programming language and running on this mini computer, and this processed information is arranged appropriately and sent to the serial communication interface card. At the same time, the information coming from the serial communication card and the image and sound information received from the camera and microphone are sent to the ground operator via this computer. In this study, EmETXe-i92U0 is used as a computer system specially designed for rovers that will take part in space exploration and other extraterrestrial missions. This compact size CPU module offers high performance and reliability even in limited space conditions. Type 6 architecture includes two high-performance connectors to provide powerful and stable data communication. The soldered 11th Generation Intel Core i7-1185G7E processor comes with integrated Intel Graphics chipset and offers extraordinary processing power. EmETXe-i92U0 also has a wide operating temperature range, can operate reliably from -40 to 85 degrees Celsius. This system can meet the communication requirements with Intel I219LM PCIe GbE PHY and iAMT support. It also has Dual Channels 24-bit LVDS, Analog RGB and 3 DDI ports to support four independent displays. Configurable with AMI UEFI BIOS, this system is an excellent choice for demanding industrial control and data communication applications. EmETXe-i92U0 offers the best computing solution for rovers that will take part in extraterrestrial missions with its low power consumption, robust structure and high processing capacity.

In addition, different communication methods are used to ensure communication throughout the mission. The first of these is the HackRF One, a Software Defined Radio (SDR) device that can operate between 1 MHz and 6 GHz. This extraordinary device offers a perfect tool for testing, developing and exploring modern and future radio technologies. The HackRF One stands out with its functionality and flexibility. Its half-duplex transceiver feature can both receive and transmit radio signals. The 20 million sample rate per second provides high-precision data collection and processing. Its compatibility with GNU Radio, SDR# and other software increases the flexibility and usability of this device. The software-configurable RX and TX gain and baseband filter of the HackRF One allow users to customize the device to suit various applications. In addition, the software-controlled antenna port power provides greater flexibility and sensitivity. In addition to its excellent performance, the HackRF One is easy to use with its high-speed USB 2.0 connection and USB power capability. However, for long range VHF communications the LimeSDR Mini 2.0 offers a great solution. This next generation software defined radio maintains the well-known features of the previous

LimeSDR Mini, but has been enhanced with a new FPGA. The LimeSDR Mini 2.0 has the same form factor, the same LMS7002 RF transceiver, but houses a more powerful Lattice ECP5 FPGA. The ECP5 FPGA provides the ground operator with on-board processing capability. The LimeSDR Mini 2.0 can operate at frequencies from 10 MHz to 3.5 GHz, and is also equipped with full duplex receive and transmit capability.

The space exploration vehicle is equipped with 3 separate camera systems to provide observation and reconnaissance purposes. The first of these; See3CAM_CU135 is a 13 MP fixed-focus main camera that provides high-resolution 4K video shooting and advanced image processing capabilities. This USB is based on a 1/3.2" AR1335 CMOS image sensor with 1.1µm pixel BSI technology. This 4K card camera incorporates a high-performance Image Signal Processor chip (ISP) specifically optimized for video streaming and image processing applications. The ISP and sensor settings are carefully tuned to produce superb video quality in both uncompressed UYVY and Compressed MJPEG formats. The camera can also achieve high frame rates of up to 816 fps in Region of Interest (ROI) resolutions, adapting to a variety of application scenarios. In addition to the main camera, the second camera, See3CAM_CU27, is a Full HD USB 3.1 Gen 1 camera based on the Sony® STARVIS™ IMX462 sensor, capable of capturing high-quality images in the near-infrared region. The IR sensitivity of the IMX462 sensor makes this camera an outstanding tool for low-light conditions, enabling high-quality and clear images to be captured. The camera is able to optimize color accuracy by including a high-performance (ISP) that performs important functions such as auto exposure and auto white balance. It offers high frame rates such as FHD @ 100 fps, making it easier to capture fast-moving objects. It provides video output in uncompressed UYVY and compressed MJPEG formats. In addition, Tara - USB 3.0 Stereo Camera with MT9V024 sensor is used to meet the depth perception and stereoscopic imaging requirements. The camera system uses Velodyne HDL-32E LIDAR with 32 LIDAR channels and a 360° horizontal field of view aligned from +10.67° to -30.67° for security enhancement and terrestrial mapping purposes. The HDL-32E generates point clouds of up to 695,000 points per second with a range of up to 100 meters and typical accuracy of ±2 cm. All these systems are carried on the rover by Pololu High-Power Motor HP 25D 12V power supply, which can control both rear wheels independently, and cylindrical brushed DC geared motors are used. This motor provides powerful use in high-performance tasks with 8.4 Nm torque and 400 RPM (revolutions per minute). Jrk G2 21v3 USB Motor Controller is used to meet both precise control and feedback requirements during operation of the motors and to control them via computer interface. LiFePO4 Solar Power Lithium Storage Battery is used for the operation of the systems.

5 Control System

5.1 Equations of Motion (EoM) for Differential Drive Space Exploration Rover (DDSER)

This section discusses the kinetics of a differential drive space exploration rover. The kinematics of differential drive rovers are called differential drive because their primary mode of motion is the difference in the speed of the two tires. In other words, if both wheels turn in the same direction and speed, the rover moves straight ahead. If they turn in the same speed but in reverse, the rover moves straight back. Also, if one side turns faster than the other but both are in the same direction, the rover turns in that direction. For example, if the right wheel of the rover turns faster, it is observed that the rover starts to turn counterclockwise when viewed from above. Similarly, if the left wheel of the rover turns faster, it is observed that it starts to turn clockwise.

As shown in Fig. 1, the right tire and the left tire ϕ defined by ϕ_1 (right) and ϕ_2 (left) have two degrees of freedom, while the body has three degrees of freedom, x, y and θ. Therefore, the five-variable generalized coordinate $q = [x_b, y_b, \theta, \phi_1, \phi_2]$ is used for the pose of this DDSER. Here, θ is expressed as the angle of the body or the angle of the x_b axis of the body frame relative to the x_e of the earth frame. In addition, x_b, y_b are expressed as the location of the center of the wheels, not the center of the robot. In addition, when two wheels are connected using an axis that is the length of the wheels, the body width defined as 2w is obtained. In this context, the centroid of the DDSER can be in front of or behind the body, and in this study, it is defined as d unit away from the center of mass center of the wheels and zero with the body axis. Therefore, the center of mass rover is expressed as perfectly symmetrical with respect to the center of the wheel.

It should be noted that, despite having five generalized coordinates, each variable is not independent and there are constraints between them. In the context of these constraints, it is assumed that the wheels do not slip, that is, they do not move horizontally or vertically, their speeds are zero and they are completely stationary, with c_1 (right) and c_2 (left) being the initial contact points. In addition, the center of each tire is defined as h_1 (right) and h_2 (left). From this, no no-slip condition imposes some constraints between these degrees of freedom. The first thing to note here is the speed of the center of each tire, expressed by Eq. 1.

$$\vec{V}_{h_1} = \begin{bmatrix} \dot{x} + w\dot{\theta}\cos\theta \\ \dot{y} + w\dot{\theta}\sin\theta \\ 0 \end{bmatrix}, \qquad \vec{V}_{c_1} = \vec{V}_{h_1} + \vec{\omega}_{w_1} \times \vec{R}_{h_1 c_1} = \begin{bmatrix} \dot{x} + w\dot{\theta}\cos\theta + \rho\dot{\phi}_1\cos\theta \\ \dot{y} + w\dot{\theta}\sin\theta + \rho\dot{\phi}_1\sin\theta \\ 0 \end{bmatrix} \tag{1.a}$$

$$\vec{V}_{h_2} = \begin{bmatrix} \dot{x} + w\dot{\theta}\cos\theta \\ \dot{y} + w\dot{\theta}\sin\theta \\ 0 \end{bmatrix}, \qquad \vec{V}_{c_2} = \vec{V}_{h_2} + \vec{\omega}_{w_2} \times \vec{R}_{h_2 c_2} = \begin{bmatrix} \dot{x} - w\dot{\theta}\cos\theta - \rho\dot{\phi}_2\cos\theta \\ \dot{y} - w\dot{\theta}\sin\theta - \rho\dot{\phi}_2\sin\theta \\ 0 \end{bmatrix} \tag{2.b}$$

Where $\vec{V}_{h_1}$ is the speed of the center of the right tire, $\vec{V}_{h_2}$ is the speed of the center of the left tire, $\vec{V}_{c_1}$ is the speed of the contact of the right tire, $\vec{V}_{c_2}$ is the speed of the contact of the left tire, $\vec{\omega}_{w_1}$ is the angular velocity of the right side of the vehicle, $\vec{\omega}_{w_2}$ is the angular velocity of the left side of the vehicle, $\vec{R}_{h_1}$ is the radius of the right wheel from the center h_1 and $\vec{R}_{h_2}$ is the radius of the left wheel from the center h_2.

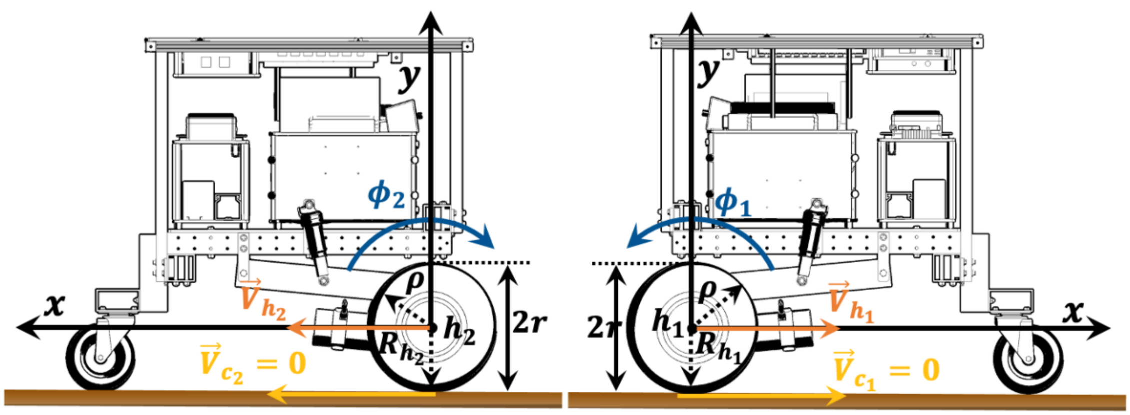

Fig. 1. 2D path planning for roadmaps with different obstacles.

The speed expressions defined here are defined with respect to the earth frame. DDSER rotates with an angular velocity ω, which is the time derivative of θ. In this context, the configuration of the obtained DDSER is shown using the 5 generalized coordinates and their time-dependent changes shown in Eq. 2.

$$\vec{q} = [x \quad y \quad \theta \quad \phi_1 \quad \phi_2]^T \tag{2.a}$$

$$\dot{\vec{q}} = [\dot{x} \quad \dot{y} \quad \dot{\theta} \quad \dot{\phi}_1 \quad \dot{\phi}_2]^T \tag{2.b}$$

As mentioned before, the fact that the contact points are completely stationary clearly shows that the constraints expressed in the four rows of the matrices in the equations should be equal to zero. The no-slip condition for the contact points is provided by 4 nonholonomic constraints, 1 of which is redundant, as shown in Eq. 3. Here, it is assumed that the DDSER moves on a flat surface (no climbing up or down). Therefore, the potential energy is zero and the Lagrangian can be considered equal to the kinetic energy of the system.

$$\dot{x} + w\dot{\theta}\cos\theta + \rho\dot{\phi}_1\cos\theta = 0 \tag{3.a}$$

$$\dot{y} + w\dot{\theta}\sin\theta + \rho\dot{\phi}_1\sin\theta = 0 \tag{3.b}$$

$$\dot{x} - w\dot{\theta}\cos\theta - \rho\dot{\phi}_2\cos\theta = 0 \tag{3.c}$$

$$\dot{y} - w\dot{\theta}\sin\theta - \rho\dot{\phi}_2\sin\theta = 0 \tag{3.d}$$

Here, since x, y, θ, ϕ_1 and ϕ_2 cannot be easily integrated with respect to time, four non-holonomic constraints are formed. The reason why 1 of the 4 constraints is unnecessary here is that one of these contact points cannot move in both x and y directions and the other cannot move only in the x direction, and it cannot move in the y direction. In other words, the forward speed of the wheel results in the speed being zero and the lateral speed being zero. Here, the expression that the entire DDSER cannot move laterally to the right is expressed as a non-holonomic constraint. Since point c does not move in the Earth frame, both axes of the body frame must be zero. Here, three non-holonomic constraints are used instead of four, since one can be obtained from the other by starting from the fact that $\dot{y}_r$ is zero for DDSER. If the constraints are included in the cost function and written as "faffian constraints", it is not difficult to minimize the integral of the action. As always in the Lagrangian, the difference between the kinetic and potential energies is needed. Then, when the Faffian constraints are written in the $C(q)\,\dot{q} = 0$ format, the friction under the tires emerges. Therefore, some algebraic operations are needed when writing equation Eq. 3 in the $C(q)\,\dot{q} = 0$ format. In order to do this, the $\dot{\theta}$ terms are eliminated by adding Eq. 3.a with Eq. 3.c and Eq. 3.b with Eq. 3.4, as shown in Eq. 4.

$$\dot{x} = \frac{\rho}{2}\cos\theta(\dot{\phi}_2 - \dot{\phi}_1) \tag{4.a}$$

$$\dot{y} = \frac{\rho}{2}\sin\theta(\dot{\phi}_2 - \dot{\phi}_1) \tag{4.b}$$

Here $\dot{\phi}_2 - \dot{\phi}_1$ represent the input signals that control the rotation speed of the tires of DDSER. In this way, the speed of $\dot{x}$, $\dot{y}$ and $\dot{\theta}$ movements of DDSER can be controlled. Similarly, if Eq. 3.a and Eq. A3.c are subtracted from each other as shown in Eq. 5, $\dot{\theta}$ can be obtained.

$$\dot{\theta} = \frac{\rho}{2w}(\dot{\phi}_2 - \dot{\phi}_1) \tag{5}$$

However, it should be noted that the result obtained by subtracting Eq. 2.b and Eq. 3.d is the same as Eq. 5, so there is no need to rewrite it. Based on this, Eq. 4 and Eq. 5 are brought together and written in the matrix form shown in Eq. 6.

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \frac{\rho}{2}\begin{bmatrix}(\dot{\phi}_2 - \dot{\phi}_1)\cos\theta\\ (\dot{\phi}_2 - \dot{\phi}_1)\sin\theta\\ (\dot{\phi}_2 - \dot{\phi}_1)/w\end{bmatrix} \tag{6}$$

This equation clearly shows that $C(q)$ must be expressed in matrix form. Therefore, the "simple linear algebra" shown in Eq. 7 is obtained.

$$C(q)\dot{q} = 0 \qquad \text{where} \qquad C(q) = \begin{bmatrix}1 & 0 & 0 & \frac{\rho}{2}\cos\theta & -\frac{\rho}{2}\cos\theta\\ 0 & 1 & 0 & \frac{\rho}{2}\sin\theta & -\frac{\rho}{2}\sin\theta\\ 0 & 0 & 1 & \frac{\rho}{2w} & \frac{\rho}{2w}\end{bmatrix} \tag{7}$$

Where is the format of my faffian constraints and the $C(q)$ matrix needed in the previous grant equation. In this context, as stated before, in order to facilitate the transition to the lagrangian, which is defined as the difference between the kinetic energy and the potential energy, it is assumed that the DDSER moves in a flat orbit. In this case, the potential energy goes to zero in the equation shown in Eq. 8 and the lagrangian is equal to the kinetic energy. In the case where the orbit is not flat, an extra term is added for the potential energy.

$$\frac{d}{dt}\left[\frac{\partial\mathcal{L}(\vec{q},\dot{\vec{q}})}{\partial\dot{\vec{q}}}\right] - \frac{\partial\mathcal{L}(\vec{q},\dot{\vec{q}})}{\partial\vec{q}} - C(q)^T\vec{\lambda} = \vec{T} \tag{8}$$

The kinetic energy consists of three components, the body and the two wheels. As shown in Eq. 9, the two wheels both translate and rotate to provide the kinetic energy of the body.

$$\mathrm{K.E.} = \frac{m_b}{2}\left\|\vec{V}_b\right\|^2 + \vec{\omega}_b^T I_b \vec{\omega}_b = \frac{m_b}{2}\left[\dot{x}^2 + \dot{y}^2 + d^2\dot{\theta}^2 + 2d\dot{\theta}(\dot{y}\cos\theta - \dot{x}\sin\theta)\right] + \frac{I_b}{2}\dot{\theta}^2 \tag{9}$$

Where $\frac{m_b}{2}\left\|\vec{V}_b\right\|^2$ is the translation term, $\vec{\omega}_b^T I_b \vec{\omega}_b$ is the rotation term. In addition, m_b is the rover's mass, $\vec{V}_b$ is the body center velocity, I_b is the moment of inertia matrix. The expression for $\frac{I_b}{2}\dot{\theta}^2$ depends on the geometry and shape of the rover's body. The components of the body's velocity are shown in Eq. 10.

$$\vec{V}_{b_1} = \vec{V}_r + \vec{\omega}_r \times \vec{r}_{h_1} = \begin{bmatrix}\dot{x}\\ \dot{y}\\ 0\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \dot{\theta}\end{bmatrix} \times \begin{bmatrix}d\cos\theta\\ d\sin\theta\\ 0\end{bmatrix} = \begin{bmatrix}\dot{x} - d\dot{\theta}\sin\theta\\ \dot{y} + d\dot{\theta}\cos\theta\\ 0\end{bmatrix} \quad , \quad \vec{\omega}_b = \begin{bmatrix}0 & 0 & \dot{\theta}\end{bmatrix}^T \tag{10}$$

Where $\vec{V}_r$ is the speed placed on the body reference frame, $\vec{\omega}_r$ is the angular speed placed on the body reference frame, $\vec{r}_{h_1}$ is the radius of right wheel center w.r.t center of mass, $\dot{x} - d\dot{\theta}\sin\theta$ x-velocity components of centroid and $\dot{y} + d\dot{\theta}\cos\theta$ y-velocity components of centroid The center of mass of the main vehicle body is located at a distance "d" along the x-axis of the body fixed reference frame. The speed expression in Eq. 10 is written in explicit form in Eq. 9, and the right side of the equation gives the kinetic energy of the body. Depending on the speeds of the centers needed for the wheels calculated in Eq. 1, the right wheel speed in Eq. 11.a is $\vec{\omega}_{b_1}$rotating around its own axis with $\dot{\phi}_1$, but the body is also rotating with $\dot{\theta}$. Then, the inertia matrix is shown in Eq. 11.b.

$$\vec{\omega}_{b_1} = \dot{\phi}_1\begin{bmatrix}\sin\theta\\ \cos\theta\\ 0\end{bmatrix} + \dot{\theta}\begin{bmatrix}0\\ 0\\ 1\end{bmatrix} = \begin{bmatrix}\dot{\phi}_1\sin\theta\\ \dot{\phi}_1\cos\theta\\ 0\end{bmatrix} \tag{11.a}$$

$$I_w^b = \begin{bmatrix}I_{xx}^b & 0 & 0\\ 0 & I_{yy}^b & 0\\ 0 & 0 & I_{zz}^b\end{bmatrix} = \begin{bmatrix}\frac{m_w}{12}(3\rho^2 + t^2) & 0 & 0\\ 0 & \frac{m_w}{2}\rho^2 & 0\\ 0 & 0 & \frac{m_w}{12}(3\rho^2 + t^2)\end{bmatrix} \tag{11.b}$$

Here, each wheel is assumed to be a disk of thickness t and radius ρ. This allows us to calculate the moment of inertia matrix for each wheel. All of these come from a disk shape. Therefore, each will be considered a disk model for simplification. The resulting inertia tensor is transformed as shown in Eq. 12.

$$I_w = R_{Z,\theta} I_w^b R_{Z,\theta}^T \begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}I_{xx}^b & 0 & 0\\ 0 & I_{yy}^b & 0\\ 0 & 0 & I_{zz}^b\end{bmatrix}\begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix} \tag{12}$$

Where $R_{Z,\theta}$ represents the rotation matrix of the body with respect to the earth frame and I_w^b represents the moment of inertia of the wheels. The reason for this is that the parameters of the kinetic energy are tried to be expressed on the same axis by applying the rotation matrix to the moment of inertia defined in the body frame and defining it in the earth frame. From here, the kinetic energy is obtained for two wheels as Eq. 13.a right and Eq. 13.b left.

$$\mathrm{K.E.} = \frac{m_w}{2}\left\|\vec{V}_{b_1}\right\|^2 + \frac{1}{2}\vec{\omega}_{b_1}^T I_b \vec{\omega}_{b_1} = \frac{m_w}{2}\left[\dot{x}^2 + \dot{y}^2 + W^2\dot{\theta}^2 + 2W\dot{\theta}(\dot{x}\cos\theta + \dot{y}\sin\theta)\right] + \frac{I_{zz}^b}{2}\dot{\theta}^2 + \frac{I_{yy}^b}{2}\dot{\phi}_1^2 \tag{13.a}$$

$$\mathrm{K.E.} = \frac{m_w}{2}\left\|\vec{V}_{b_2}\right\|^2 + \frac{1}{2}\vec{\omega}_{b_1}^T I_b \vec{\omega}_{b_1} = \frac{m_w}{2}\left[\dot{x}^2 + \dot{y}^2 + W^2\dot{\theta}^2 - 2W\dot{\theta}(\dot{x}\cos\theta + \dot{y}\sin\theta)\right] + \frac{I_{zz}^b}{2}\dot{\theta}^2 + \frac{I_{yy}^b}{2}\dot{\phi}_1^2 \tag{13.b}$$

When the kinetic energy obtained for the wheels is added to the kinetic energy of the body, the three components that produce kinetic energy are brought together. To achieve this, Eq. 13 is written in Eq. 9 and the total kinetic energy is obtained from Eq. 14.

$$\mathrm{K.E._{total}} = \mathrm{K.E._{body}} + \mathrm{K.E._{right\ w.}} + \mathrm{K.E._{left\ w.}}$$
$$\mathrm{K.E._{total}} = \partial\mathcal{L}(\vec{q},\dot{\vec{q}}) = \frac{1}{2}(m_b + 2m_w)(\dot{x}^2 + \dot{y}^2) + \frac{1}{2}\left(m_b d^2 + I_b + 2m_w W^2 + 2I_{zz}^b\right)\dot{\theta}^2 + m_b d\dot{\theta}(\dot{y}\cos\theta - \dot{x}\sin\theta) + \frac{I_{yy}^2}{2}\left(\dot{\phi}_1^2 + \dot{\phi}_2^2\right) \tag{14}$$

This is a simplified version of the total kinetic energy. Here $m_b + 2m_w$ is the total mass of the rover, $\frac{1}{2}(m_b + 2m_w)(\dot{x}^2 + \dot{y}^2)$ is the kinetic energy translation of the body, $m_b d^2 + I_b + 2m_w W^2 + 2I_{zz}^b$ is the total z-axis rotational inertial kinetic energy rotation of the chassis and $\frac{I_{yy}^2}{2}\left(\dot{\phi}_1^2 + \dot{\phi}_2^2\right)$ is the kinetic energy rotation of the tires about y-axis. Also $m_b d\dot{\theta}(\dot{y}\cos\theta - \dot{x}\sin\theta)$ is the difference between the center of gravity and the center of gravity of the wheels, with d being zero. Therefore, the distance d indicates that the rover's centroid is at the center of the wheels. In addition, since the potential energy is zero, the same result is obtained as the Lagrange equation. Then, by combining the constraints, the Euler-Lagrange equation is applied with Lagrange multipliers to find the equations of motion, shown in Eq. 15.

$$\frac{d}{dt}\left[\frac{\partial\mathcal{L}}{\partial\dot{\vec{q}}}\right] - \frac{\partial\mathcal{L}(\vec{q},\dot{\vec{q}})}{\partial\vec{q}} - C^T(\vec{q})\vec{\lambda} = \vec{T} = [0 \quad 0 \quad 0 \quad \tau_1 \quad \tau_2] \tag{15}$$

Where τ_1 represents the torque applied to the right wheel, τ_2 represents the torque applied to the left wheel. In addition, since there is no external force or moment applied to the motion of the system and contributing to the motion of x, y or $\dot{\theta}$, the basic external forces or moments that will affect the motion in this system consist only of τ_1 and τ_2. However, since the motion occurs on a horizontal plane, there is no contribution from gravity, normal force and static friction. Therefore, the only generalized forces or torques that directly affect the motion of $\dot{\phi}_1$ and $\dot{\phi}_2$ are τ_1 and τ_2. Therefore, although it is not difficult to find the generalized moments of force, as shown in Eq. 16, as many $\vec{\lambda}$ vectors as there are restrictions are needed.

$$\frac{d}{dt}\begin{bmatrix} m_t\dot{x} - m_b d\dot{\theta}\sin\theta \\ m_t\dot{y} + m_b d\dot{\theta}\cos\theta \\ I_t\dot{\theta} + m_b d(\dot{y}\cos\theta - \dot{x}\sin\theta) \\ I_{yy}^b\dot{\phi}_1 \\ I_{yy}^b\dot{\phi}_2 \end{bmatrix} + m_b d\dot{\theta}\begin{bmatrix} 0 \\ 0 \\ \dot{y}\sin\theta + \dot{x}\cos\theta \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \\ \frac{\rho}{2} & 0 & \frac{\rho}{2w} \\ -\frac{\rho}{2} & 0 & \frac{\rho}{2w} \end{bmatrix}\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \tau_1 \\ \tau_2 \end{bmatrix} \tag{16}$$

Where m_t represents the total mass of the wheels and the body, I_t represents the total moment of inertia matrix. In addition, three parameters are needed for the three constraints, λ_1, λ_2 and λ_3. Only by finding the partial constraints according to Lagrangian, the final form of the equation previously expressed as $C(q)\,\dot{q}$ is shown in Eq. 17.

$$M(\vec{q})\ddot{\vec{q}} + B(\vec{q},\dot{\vec{q}}) - C^T(\vec{q})\vec{\lambda} = \vec{T} \tag{17.a}$$

$$M(\vec{q}) = \begin{bmatrix} m_t & 0 & -m_b d\sin\theta & 0 & 0 \\ 0 & m_t & m_b d\cos\theta & 0 & 0 \\ -m_b d\sin\theta & m_b d\cos\theta & I_t & 0 & 0 \\ 0 & 0 & 0 & I_{yy}^b & 0 \\ 0 & 0 & 0 & 0 & I_{yy}^b \end{bmatrix};\ B(\vec{q},\dot{\vec{q}}) = -m_b d\dot{\theta}^2\begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \\ 0 \\ 0 \end{bmatrix};$$
$$C(\vec{q}) = \begin{bmatrix} \cos\theta & \sin\theta & 0 & \frac{\rho}{2} & -\frac{\rho}{2} \\ -\sin\theta & \cos\theta & 0 & 0 & 0 \\ 0 & 0 & 1 & \frac{\rho}{2w} & \frac{\rho}{2w} \end{bmatrix} \tag{17.b}$$

Where $B(\vec{q},\dot{\vec{q}})$ is used for the centripetal and Coriolis terms, $C(\vec{q})$ is used for the constraints and $\vec{T}$ is used as the torque. Thus, the equations of motion are obtained where the inputs are $\vec{T}$ and the outputs are $\ddot{\vec{q}}$. In the forward dynamics problem or inverse dynamics, the torque amount is calculated by knowing $\vec{q}$ versus time. Another point to note is that the number of unknowns is actually eight, namely $\ddot{x}$, $\ddot{y}$, $\ddot{\theta}, \ddot{\phi}_1$, $\ddot{\phi}_2, \lambda_1, \lambda_2$ and λ_3. However, the equation $M(\vec{q})\ddot{\vec{q}}$ gives only five equations of these constraints. In this context, the other three equations needed come from the constraint equations explained in Eq. 6. However, when we look at these constraint equations, we see that in order to find the Lagrangian multipliers, a time derivative must be taken from the constraint equations as shown in Eq. 18.

$$\frac{d}{dt}[C(\vec{q})\dot{\vec{q}}] = 0 \qquad \rightarrow \qquad C(\vec{q})\ddot{\vec{q}} + \dot{C}(\vec{q})\dot{\vec{q}} = 0 \tag{18.a}$$

$$(\vec{q})\ddot{\vec{q}} + B(\vec{q},\dot{\vec{q}}) - C^T(\vec{q})\vec{\lambda} = \vec{T} \qquad \rightarrow \qquad \ddot{\vec{q}} = M^{-1}(\vec{q})[\vec{T} - B(\vec{q},\dot{\vec{q}}) + C^T(\vec{q})\vec{\lambda}] \tag{18.b}$$

In this way, by finding the other unknowns with this new form, an equation that can give all eight unknowns is formed. If Eq. 18.a and Eq. 18.b are written together and the $\vec{\lambda}$ expression is drawn, the Langrange multiplier becomes Eq. 19.

$$\vec{\lambda} = -[C(\vec{q})M^{-1}(\vec{q})C^T(\vec{q})]^{-1}\left[C(\vec{q})M^{-1}(\vec{q})\left(\vec{T} - B(\vec{q},\dot{\vec{q}})\right) + \dot{C}(\vec{q})\dot{\vec{q}}\right] \tag{19}$$

Here, the equation $\vec{\lambda}$ obtained by eliminating the expression $\ddot{\vec{q}}$ gives the restraining forces or torques. An important point is that solving forward or reverse dynamics problems by integrating the equations into each other can only be done numerically and makes it difficult to see the effects of dynamic parameters on system performance. Therefore, the zero-distance d (the center of gravity of the chassis is on the axis of the wheels) constitutes a simplifying assumption as shown in Eq. 20.

$$d = 0 \qquad \rightarrow \qquad B(\vec{q},\dot{\vec{q}}) = 0 \qquad \text{and} \qquad M(\vec{q}) = \mathrm{diag}\left(m_t, m_t, I_t, I_{yy}^2, I_{yy}^2\right) \tag{20}$$

Although this is an assumption, it is still impossible to obtain equations that can be easily analyzed. Therefore, the solution is reached by resorting to numerical solutions as shown in Eq. 21, taking into account the beginning of the rover motion occurring from the stationary state ($\theta = 0, \dot{x} = \dot{y} = 0$).

$$\ddot{x} \cong -\left(\frac{1}{\rho}\right)\left(\frac{1}{m_w + m_t}\right)(\tau_1\tau_2) \tag{21.a}$$

$$\dot{y} = 0 \tag{21.b}$$

$$\ddot{\theta} \cong -\left(\frac{w}{\rho}\right)\left(\frac{\tau_1 + \tau_2}{m_w w^2 + I_t}\right) \tag{21.c}$$

Here the lateral motion constraint (non-holonomic) remains valid. Since the torques are in opposite directions, the negative difference in the equations actually means the total torque applied. Therefore, the more torque applied to each or both tires, the faster the system goes and the greater forward acceleration it achieves. If the system is made lighter or the radius of the wheels is made smaller, more acceleration occurs. Similarly, the torques must be greater to achieve faster turns or faster angular accelerations. The moments of inertia or the reduction of the mass of the blocks provide rotation with a greater angular acceleration. As a result, in order to maximize the forward acceleration of the DDSER, the mass of the chassis and wheels must be reduced. However, by increasing the gear reduction ratio, the torques must be maximized and the radius of the wheels must be reduced. In addition, in order to maximize the angular acceleration of the DDSER, the mass and inertia of the body and wheels must be reduced. Similarly, by increasing the gear reduction ratio, torques must be maximized, wheel radii must be reduced and width must be optimized. With this information, the optimum wheelbase distance is achieved as shown in Eq. 22.

$$2w^* = 2\sqrt{\frac{I_b + 2I_{yy}^b}{3m_w}} \tag{22}$$

Where $2w^*$ represents the optimum wheel to wheel distance. Since it is an inevitable fact that increasing the gear reduction ratio and decreasing the radius of the wheels will lead to a decrease in the maximum linear and angular speeds that can be obtained with DDSER, a trade-off must be made between speed and acceleration regarding these 2 parameters. In general, this section explains how to obtain the equations of motion while designing the different parameters of DDSER.

5.2 Multivariable Central Rover Control: Feedback Linearization Approach

This section deals with the implementation of Inverse Dynamics Control (Feedback Linearization) in Joint Space. In other words, the aim of this section is to consider a system where each joint can move independently of the others and the effects of the others create a small disturbance, and instead of any limiting assumption, the general case is considered. This allows DDSER to move very fast or very slowly or to have any kind of movement. Here, a system emerges where the movement of each joint depends on the other joints, all of them affect each other and the couplings are not small. The aim of this is to provide a control input u(t) that cancels the nonlinear terms due to gravity, Coriolis and centripetal terms and to ensure that it follows any random qu(t) input. These terms can be estimated with a high degree of accuracy (if these parameters cannot be estimated correctly, Adaptive Control and/or Robust Control Algorithms must be used) and to make the resulting system linear, which can be decoupled and easily controlled by means of a PD compensator. In this context, firstly the equation of motion of a space exploration rover is shown in Eq. 23.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u \tag{23}$$

Here, $C(q,\dot{q})$ is assumed to be due to centripetal and corialis accelerations, and friction terms are neglected, including $C(q,\dot{q})$ and $\dot{q}$. However, if there are friction terms, they can be added to the friction terms. In addition, u represents the control signal in the equation. In this method, the nonlinearity of the system is canceled by using feedback linearization with two control layers, inner and outer loop. Then the outer layer is simply obtained with a PD compensator as a forward term, and the feedback is linear. Here, differently, this term calculated for the inner loop is not based on the desired values, but only on the desired and some feedbacks. Therefore, a feedback linearization really occurs here. Then, if the friction and gravity terms were not present, my equation would not exactly resemble a second-order linear differential equation, but it would not be continuously linear. The reason for this is that the matrix $M(q)$ is a variable, not a constant. Another point that needs to be addressed is that if the friction and gravity terms are eliminated, a second-order system is formed as $a_q = \ddot{q}$ $(M(q)\ddot{q} = a_q)$. If the terms in aq are linear, a linear second-order system is obtained from a nonlinear second-order system, which allows easy use of controllers such as the PD controller or PD+feed forward used in the linear system. For this reason, the u signal is selected as the feed forward controller. In this way, all nonlinear terms are canceled and feedback linearization or linearization of the system is realized through this feedback term. From this point on, during the estimation of $C(q,\dot{q})$ and $g(q)$, the controller needs encoders by using q and $\ddot{q}$. However, in order to determine $C(q,\dot{q})$ and $g(q)$, commentators need to give position and velocity feedback. Assuming that $C(q,\dot{q})$ and $g(q)$ are well known and that these two terms cancel each other out and also that the matrix m is invertible, and $a_q = \ddot{q}$ is obtained, the “Inner Loop Control Law” shown in Eq. 24 cancels it out.

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = u \tag{24}$$

The goal of the outer loop is to define a_q in such a way that the error shown in Eq. 26 can be equal to zero. Therefore, if a_q can be chosen, the zero steady state error or asymptotically approaches zero. With the provision of these conditions, the “Outer Loop Control Law” shown in Eq. 25 becomes operational. In order to ensure the operational state, it is necessary to determine what a_q will be.

$$a_q = K_d e(t) + K_d \dot{e}(t) + \ddot{q}_d(t) \tag{25}$$

$$e(t) = q_d - q \tag{26}$$

Here $\ddot{q}_d(t)$ is the acceleration of the joint desired, i.e. feed forward term, e(t) is the error term, q_d is the desire control signal, q control signal is the proportional term coming from the K_p PD controller and K_d is the derivative term coming from the K_p PD controller. From here, if $a_q = \ddot{q}_d$ and $\ddot{e}(t) = \ddot{q}_d - \ddot{q}$, $\ddot{e}(t)$ can be expressed as in Eq. 27.

$$\ddot{e}(t) = K_d e(t) + K_p e(t) = 0 \tag{27}$$

Since it is known that $e(t) = e_0 e^{-\omega t}$, it is placed as $K_d = 2\omega$ and $K_p = \omega^2$ based on the characteristic equation. However, this situation is valid for a single degree of freedom. Since the system has many degrees of freedom in DDSER, although it appears as $K_d = 2\omega$ and $K_p = \omega^2$, the equation takes the form shown in Eq. 28. Therefore, if K_d and K_p are chosen in a way that determines the speed at which ω converges to zero, the error converges to zero as quickly as it depends on the speed of ω, thus ensuring the acceleration of the system. However, choosing ω very, very large here does not mean that it will be beneficial to obtain large gains and large gains in the real world.

$$K_p = \mathrm{diaq}\{\omega_1^2, \ldots\ldots \omega_n^2\} \tag{28.a}$$

$$K_d = \mathrm{diaq}\{2\omega_1, \ldots\ldots 2\omega_n\} \tag{28.b}$$

$e(t)$ provides the tracking error to converge to zero in a critically damped manner. As shown in Fig. 2, the outer loop controller takes q_d and then produces $\ddot{q}_d$ within itself. In addition, when it takes q and q_d, it subtracts here and produces the thing that creates the error and can produce $K_p e(t)$ and $K_d e(t)$ when it has the error. The PD controller is in the double derivative of the feedforward term with the outer loop controller. Then, if a_q is passed to the inner loop and the inner loop $M(q)$ takes a_q and adds the Coriolis centripetal and gravity terms to it, the control signal is produced. The multivariable central motion algorithm using feedback linearization is a very powerful method because it eliminates many of the limiting assumptions. The system can be controlled by adding the damping term and dry friction terms to the other terms in terms of $u(t)$. This method is shown to work when the estimates of the robot parameters are correct and the calculations of the control loops can be done quickly. In addition, by changing only the outer loop as shown in Eq. 29, starting from Eq. 23, the robot can be controlled to achieve tracking in the task area.

$$K_p = \mathrm{diaq}\{\omega_1^2, \ldots\ldots \omega_n^2\} \tag{28.a}$$

$$K_d = \mathrm{diaq}\{2\omega_1, \ldots\ldots 2\omega_n\} \tag{28.b}$$

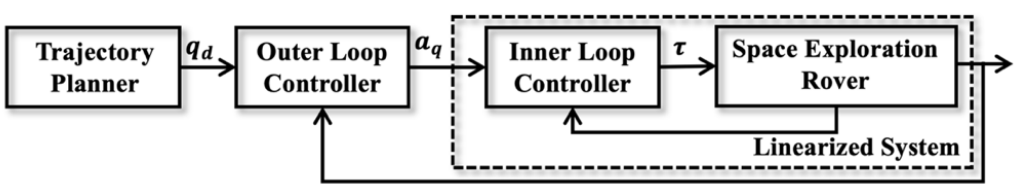

Fig. 2. Loop control schematic.

The multivariable central motion algorithm using feedback linearization is a very powerful method because it eliminates many of the limiting assumptions. The system can be controlled by adding the damping term and dry friction terms to the other terms in terms of $u(t)$. This method is shown to work when the estimates of the robot parameters are correct and the calculations of the control loops can be done quickly. In addition, by changing only the outer loop as shown in Eq. 29, starting from Eq. 23, the robot can be controlled to achieve tracking in the task area.

$$\dot{X} = J_a(q)\dot{q} \tag{29.a}$$

$$\ddot{X} = J_a(q)\ddot{q} + \dot{J}_a(q)\dot{q} \tag{29.b}$$

Where X is the end-effector pose using the minimum orientation representation and J is the analytic Jacobian. If X_d is known, q_d can be obtained from inverse kinematics. For the additional calculation of inverse kinematics, the time derivative of the Jacobian is calculated as shown in Eq. 30.

$$\mathrm{X} = \begin{bmatrix} \mathrm{d} \\ \alpha \end{bmatrix} \quad \rightarrow \quad \dot{\vec{\mathrm{X}}} = \begin{bmatrix} \mathrm{V_r} \\ \omega_\mathrm{r} \end{bmatrix} = \mathrm{J_a}\dot{\vec{\mathrm{q}}} \tag{30}$$

Here d is the end effector position, α is the end effector orientation (not the rotation matrix), $\dot{\vec{X}}$ is the end effector velocity, V_r is the velocity of the body, ω_r is the angular acceleration of the body and J_a is the analytic jacobian that represents the minimum representation of the rotation. In this case, the time derivative of the analytic jacobian is needed to relate the equations in the joint space to the task space. Therefore, when the inner loop control law expressed by Eq. 24 is combined with the time derivative of the analytic jacobian, $\ddot{q}$ is obtained from the relationship shown in Eq. 30.

$$\ddot{\vec{\mathrm{q}}} = \vec{\mathrm{a}}_\mathrm{q} = \mathrm{J_a^{-1}}\left(\ddot{\vec{\mathrm{X}}} - \dot{\mathrm{J}}_\mathrm{a}\dot{\vec{\mathrm{q}}}\right) \tag{31}$$

Where the term $\dot{J}_a\dot{\vec{q}}$ is not desired for a simple PD control. In this context, a_q is obtained as shown in Eq. 32.

$$\mathrm{a_q} = \mathrm{J^{-1}}\{\ddot{\mathrm{X}}_\mathrm{d} + \mathrm{K_p}(\mathrm{X_d} - \mathrm{X}) + \mathrm{K_d}(\dot{\mathrm{X}}_\mathrm{d} - \dot{\mathrm{X}}) - \dot{\mathrm{J}}\dot{\mathrm{q}}\} \tag{32.a}$$

$$\widetilde{\mathrm{X}} = \mathrm{X_d} - \mathrm{X} \tag{32.b}$$

Where $\widetilde{X}$ represents the error. From here, a_q takes the form in Eq. 33.

$$\ddot{\widetilde{\mathrm{X}}} + \mathrm{K_d}\dot{\widetilde{\mathrm{X}}} + \mathrm{K_p}\widetilde{\mathrm{X}} = 0 \tag{33}$$

The terms here are obtained from the feedback of the task area. Here q is obtained from the encoder sensor, while q for the motor is obtained from the tachometer sensor. In order to obtain feedback for angular positions and speeds, the end effector position in the task area must be adjusted by the sensor used or the end effector speeds in the work area. In addition, range-sensor is used during X measurement and computer vision is used during $\dot{X}$ acquisition. As a result, the difficulties of the real world can be facilitated by converting the model into a task area.

5.3 Trajectory Path Traversal

The equation of motion for the differential drive space exploration rover, as shown in Eq. 34, is expressed as follows.

$$\mathrm{M}\ddot{\mathrm{q}} + \mathrm{B} + \mathrm{C^T}\lambda = \tau \tag{34}$$

In the governing equation of this space rover, M acceleration, B any centripetal or coriolis accelerations $C^T\lambda$ term due to the constraints of the rover and τ input to the system. The constraint differential drive here is due to the fact that the space rover cannot move sideways. Therefore, when this y_r is aligned with the y_r direction on the body axis, $\dot{y}_r$ is equal to zero. As a result of the non-holonomic constraint, the robot is prevented from moving sideways or sideways, and only forward and left or right movements can be made along the circle. Another important point to note here is that the $C^T\lambda$ constraint term in order to correct the friction direction is expressed as the negative of the expression in the equation of motion and provides the correction of the direction of motion with a positive sign. In the presence of constraints, the $C^T\lambda$ lagrangian term in the equation of motion is expressed in Eq. 3 with the λ lagrangian multiplier and the C terms coming from the faffian constraint. These are the forces of the constraint. This traction force represents the friction under the tire. The C matrix, written as Faffian constraints, is the matrix that describes all the non-slip conditions for the slipping conditions that arise from the fact that the tires must not slip on both the right and left sides. Then, if we start from the equation of motion, we can find λ as a result of combining the equations of motion with the constraint equations as in Eq. 35.

$$\lambda = -[\mathrm{C}\mathrm{M^{-1}}\mathrm{C^T}]^{-1}[\mathrm{C}\mathrm{M^{-1}}(\tau - \mathrm{B}) + \dot{\mathrm{C}}\dot{\mathrm{q}}] \tag{35}$$

With this λ equation, the friction force and tracking force under each tire can be found. In addition, λ is a function of the input torque, τ. The excessive torque that the tire is exposed to while driving is clearly revealed by the amount of friction that occurs under the tires when the gas pedal is pressed too hard or too lightly, due to the difference in torque on the replaced tires. As expressed in this equation, λ is a function of the inputs, rather than being independent of the inputs. As shown in Fig. 3, while the space exploration rover maintains a constant speed of V_d (speed of the robot along the path), it is required to use a control law for the wheel torques, τ, while following a desired trajectory determined as $y_d = x_d^2$ as an application from different types of trajectories. First of all, in order to simplify the function $y_d = x_d^2$, if a parabola starting from the origin or close to the origin is desired to be followed, a torque is applied to the DC motors that turn the wheels to the right or left. In addition, by only rotating the torques, it is ensured that the desired trajectory is followed and the desired speed V_d is kept constant. Therefore, instead of going at a variable speed along the path, the space exploration rover is aimed to pass the path at the desired constant speed. Since there is nonlinearity in B, M and C, it is clearly shown that the governing equations are not linear. Here, the most logical solution to provide motion control is the "Multivariate Central Rover Control through Feedback Linearization" discussed in the previous section. The purpose of this non-linear control feedback linearization is to get rid of non-linear terms with the provided torque. In this way, all joints are controlled simultaneously and basically, they are provided to follow the desired input signals. In order to provide this, asymptotic zero-going error is made by using a PD or PID controller. At the basis of this, there are two control layers: an inner loop that creates the torque based on PID or PD control and an outer loop that produces the desire acceleration, a_q.

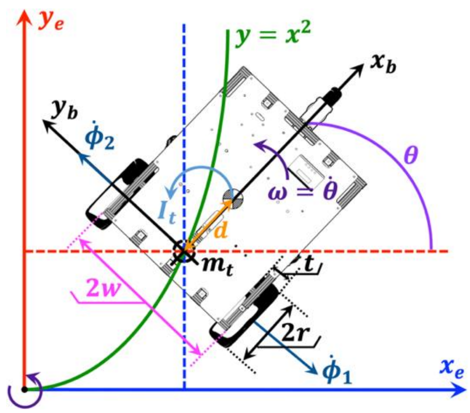

Fig. 3. Loop control schematic.

As shown in Eq. 35 and Eq. 36, due to the presence of nonlinear terms $(C(q.\dot{q})\dot{q} + g(q))$ in the inner loop, u, and outer loop, a_q, when the torque,τ, corresponding to u, is substituted into Eq. 35, on the right-hand side of Eq. 34, the nonlinear terms $(C(q.\dot{q})\dot{q} + g(q))$ simplify to only $M(q)\ddot{q} = M(q)a_q$. Therefore, the invertibility of the M term gives the equation $\ddot{q} = a_q$. Then, the a_q term is selected based on the PD control and fit forward so that the error converges asymptotically to zero. This clearly demonstrates the purpose of the control law used. Unlike in the previous section, the left-hand side is expressed as a function of the right-hand side and independent of everything else. However, it is seen that the C^T term explained in this section depends on the τ expression on the right side. In order to use the results obtained here in the control strategy, Eq. 34 and Eq. 35 must be combined and Eq. 36, which is the governing equation that does not include the τ term on the left side, must be established.

$$M\ddot{q} + B - C^T[CM^{-1}C^T]^{-1}\dot{C}\dot{q} + C^T[CM^{-1}C^T]^{-1}CM^{-1}B = \{C^T[CM^{-1}C^T]^{-1}CM^{-1} + I\}\tau \tag{36}$$

There is only τ on the right side, only this is the appropriate motion equation for i to apply the controller. When the movements from this equation are returned to feedback linearization for inner-outer loop control application, it is necessary to get rid of all the terms added while making the selection for τ. From here, in order to go down to the $M\ddot{q}$ expression on the left side of Eq. 36, the remaining terms on the left side must be included in the control loop control law. In addition, the inverses of the terms multiplied by τ on the right side must be taken. In this way, when multiplied by τ, they can cancel each other with their inverses. In that case, Eq. 37 expresses "Inner Loop Control".

$$\tau = \{C^T[CM^{-1}C^T]^{-1}CM^{-1} + I\}^{-1}[Ma_q + B - C^T[CM^{-1}C^T]^{-1}\dot{C}\dot{q} + C^T[CM^{-1}C^T]^{-1}CM^{-1}B] \tag{37}$$

If Eq. 37 is put into Eq. 36, the multiplier in curly brackets in Eq. 37 cancels each other with Eq. 36. From here, the entire left side of Eq. 36 is equal to the terms in the second closed bracket in Eq. 37 and cancels each other. The only result to be obtained from here is $M(q)\,\ddot{q} = M(q)a_q$. In addition, if the inverse of the term $M(q)$ exists and both sides can be multiplied by the inverse of $M(q)$, it is simplified to the expression $\ddot{q} = a_q$ and converted into a linear system. If the system of the space exploration rover expressed with the equations is not linear, the nonlinear system is converted into a linear system by selecting the control and canceling the nonlinear terms, and feedback linearization is performed when it needs to receive feedback from the q and $\dot{q}$ terms to create matrices and derivative expressions. The equation can be created using the feedbacks from the q and $\dot{q}$ terms. With this, the control signal can be created and all nonlinear terms can be eliminated. It should be known that this elimination can only be achieved if the system model is almost perfect. The B matrix, C matrix, and M matrix must be known exactly. In cases where these are not known, an adaptive control or a strong control must be used or a system definition must be made to define these expressions first. Finally, a feedback linearization idea is implemented. When the information from the M, B and C matrices is considered almost perfect or at least very close, the maximum cancellation $\ddot{q} = a_q$ can be obtained. After this, the control part occurs where the outer control loop comes into play. In order for the error in the queue to go to zero, the aim of the outer control law is for the error q (q_e) to go to zero as the limit goes to infinity in time, as shown in Eq. 38.

$$q_e = q_d - q, \qquad \lim_{t\to\infty} q_e \to 0 \tag{38}$$

In the case where the errors asymptotically converge to zero, the combination of PD control and feedforward is selected for a_q. Therefore, the expression a_q, as shown in Eq. 39, contains a combination of a PD control and a feedforward component, taking into account the dynamics of the q_e error.

$$a_q = \ddot{q}_d + K_p q_e + K_d \dot{q}_e \tag{39}$$

Where $\ddot{q}_d$ represents the feed reward term that does not depend on the feedback from the system. This term does not depend on q, $\dot{q}$ or $\ddot{q}$. This predefined $\ddot{q}_d$ signal can be obtained in advance and its second derivative can be transmitted to the system. In other words, this is called the feed forward term. In addition, the P (Proportional) controller in the PD term is again expressed as $K_p \dot{q}_e$ and the D (Derivative) controller term is expressed as $K_d \dot{q}_e$. Here $\dot{q}_e$ represents the error and $\dot{q}_e$ represents the derivative of the error. When Eq. 38 and Eq. 39 are combined in Eq. 40, the second order homogeneous differential equation is obtained.

$$\ddot{q}_e + K_d \dot{q}_e + K_p q_e = 0 \tag{40}$$

If the values of K_d and K_p are chosen appropriately, it becomes certain that $\lim_{t\to\infty} q_e \to 0$ is obtained. Here, the appropriate choice for K_p and K_d is a group ω_i squared ($K_p = \{\omega_i^2\}$) for K_p, while the same group ω_i but doubled instead of squared ($K_d = \{2\omega_i\}$) for K_dK_d. This ensures that the q_e solution shown in Eq. 41 will be obtained. Since all of the ω_i terms expressed here are positive, it shows that all these negative exponential values go to zero as time passes and the magnitude of ω_i controls the convergence rate, bringing larger gains.

$$q_e = Ae^{-\omega_i t} + Bte^{-\omega_i t} \tag{41}$$

However, this causes convergence to occur faster. Based on Eq. 39, if the system shows a steady state error, it is converted to PID control by bringing it to the form of Eq. 42 with the help of the I (Integral) controller.

$$a_q = \ddot{q}_d + K_p q_e + K_d \dot{q}_e + K_I \int q_e dt \tag{42}$$

In case of noise in the signal, the derivative term can be added to prevent the derivative from growing too much by adding a PID plus filter. On the other hand, PD, PI, PID PLUS filter can vary depending on whether there is a steady state error or noise. Although there is no need for noise filtering in the simulation case, in real life, since all signals received from q given in Eq. 38 definitely have noise, the noise must be filtered. However, if q_d is received from some sensors, it may not necessarily have noise since it will be a predefined signal. Because something is being followed in real time. If this is not predefined, there may be noise.

Therefore, the PID plus filter must be used. Similarly, it may depend on whether the system shows a steady state error or not. Although no filter is needed during the simulation, a little bit of steady state error occurs while the system is being tested. Therefore, a very small amount of integral controller is added after making sure that the gain is very small in order to avoid stability in the system. As a result, with inner loop control expressed by Eq. 36 and outer loop control as shown in Eq. 37 system control can be provided and the desired trajectory can be followed. In addition, it was stated that the q term in the equation of motion of a differential drive space exploration rover has five elements as $x, y, z, \theta, \phi_1, \phi_2$. The position of q is chosen to be x and y of the body center. Then, ϕ_1 (right) and ϕ_2 (left) represent the ϕ of the right tire and the left tire. When the q expression is needed, it is necessary to have q_d or $\dot{q}_d$ for the control law. Then, the x and y elements in the desire q_d having 5 elements ($q_d = [x_d, y_d, \theta_d, \phi_{1,d}, \phi_{2,d}]$)) can be taken from the x point on the curve of any road. However, θ represents the slope of the tangent line, which is the tangent inverse of y . Thus, it can desire find the corresponding angle of two tires. Therefore, the expression desire q must be found. In order to find desire $\dot{q}_d$, its derivative can be found independently in 5 elements ($\dot{q}_d = [\dot{x}_d,\ \dot{y}_d,\ \dot{\theta}_d,\ \dot{\phi}_{1,d},\ \dot{\phi}_{2,d}]$). Then, the expression x can be integrated in a way that it changes with respect to time instead of changing with respect to x. In addition, here, $\ddot{q}_d$ and q signals are needed in addition to $\dot{q}_d$.

When $\dot{q}_d$ is obtained, it provides to obtain q_d with integration on Simulink and $\dot{q}_d$ with derivative. Then the target here depends on obtaining $\dot{q}_d$ expression. During its finding, the slope of y according to x should be selected in such a way that traversing the curve speed (V_d) can be provided. Therefore, in order to find $\dot{x}_d, \dot{y}_d$ etc. expressions, these two criteria should be taken into consideration at the same time. In order to produce these signals, 5 elements should be examined separately. Since it is the first term, it is possible to start from the speed relation during the obtaining of $\dot{x}_d$. The position of the vehicle expressed with x and y axes can be expressed as rectangular coordinates Eq. 43, where V_d is the constant speed of the robot.

$$V_d = \sqrt{(V_x)^2 + \left(V_y\right)^2} \tag{43}$$

Where V_x is the derivative of x with respect to time and V_y is the derivative of y with respect to time, and since these expressions are what is desired, this desire becomes V_x and this desire becomes V_y. Then these terms take the form of the derivative of $\dot{x}_d$ and $\dot{y}_d$. By taking the square of both sides, Eq. 44 is obtained and it is accepted as 1 unit during the testing of the system. Although the vehicle's passing speed on the road is selected as one unit, i.e. 1 meter per second, different values can be set for different control design conditions according to how fast the vehicle is desired to move.

$$V_d^2 = 1 = \dot{x}_d^2 + \dot{y}_d^2 \tag{44}$$

Since the desired trajectory is considered as $y_d = x_d^2$ is a function of $\dot{x}_d$. If the derivative of both sides with respect to time is taken as shown in Eq. 44, Eq 45 is obtained.

$$\dot{y}_d = 2x_d\dot{x}_d \tag{45}$$

Here, instead of the derivative of the right side with respect to x_d, the derivative with respect to x_d is first taken and this expression is multiplied by the derivative of x_d with respect to time, thus the chain rule is used. In addition, the speed expression determined as 1 unit is formed into the Eq. 47 and the equation is arranged according to its factored form.

$$1 = \dot{x}_d^2 + (2x_d)^2\dot{x}_d^2 = \dot{x}_d^2[1 + (2x_d)^2] \tag{46}$$

The $\dot{x}_d$ term is removed from this rearranged expression to form Eq. 47.

$$\dot{x}_d = \frac{1}{\sqrt{1 + (2x_d)^2}} \tag{47}$$

It has been found that $\dot{y}_d$ can be expressed with Eq. 45 while $\dot{x}_d$ is found. When the $\dot{x}_d$ expression expressed with Eq. 47 is combined with the $\dot{y}_d$ expression expressed in Eq. 45, the final $\dot{y}_d$ expression is formed with Eq. 48.

$$\dot{y}_d = \frac{2x_d V_d}{\sqrt{1+(2x_d)^2}} \tag{48}$$

After finding the first two elements of $\dot{q}_d$, $\theta_d{}'$ must be found and $\dot{\theta}_d$ must be expressed by taking its derivative with respect to time. The slope of the tangent line θ is equal to the arctangent of y'. However, since the expression y' is the derivative of y_d with respect to x_d ($d(y_d)/d(x_d)$), it is expressed with y'=2x_d. After finding θ_d' by taking the derivative of θ_d with respect to time (d/dt), the expression $\dot{\theta}_d$ is obtained by taking the time derivative of the arctangent, Eq. 49.

$$\theta_d = \operatorname{atan}(y_d') = \operatorname{atan}(2x_d) \;\; \text{then} \;\; \dot{\theta}_d = \frac{2\dot{x}_d}{1+(2x_d)^2} \tag{49}$$

It is clearly seen that the equation expressed here is the derivative of the arctangent of $\dot{x}_d$. From here, when Eq. 45 is combined with Eq. 47, the final $\dot{\theta}_d$ equation is obtained with Eq. 50.

$$\dot{\theta}_d = \frac{2V_d}{\left(\sqrt{1+(2x_d)^2}\right)^3} \tag{50}$$

Finally, by finding the rate of change or angular velocity of the two tires in the desired condition, $\dot{\phi}_1$ and $\dot{\phi}_2$, all elements for implementing control will be known. As shown in Fig. 2, the relationships between the angular velocities of the wheels and the forward speed of the body center and the angular rotation of the body (radius of the two tires, r, and radius width of the body, W, are known. Since $\dot{\phi}_1$ and $\dot{\phi}_2$ are in the same direction in the kinematic equations, the counter clockwise signs for $\dot{\phi}_1$ and the clockwise signs for $\dot{\phi}_2$ are expressed as opposite. Looking at the two linear equations expressed in Eq. 51, V_d is a controllable parameter and after deriving $\dot{\theta}_d$, $\dot{\phi}_1$ and $\dot{\phi}_2$ are obtained by the known right-hand-side relation.

$$\begin{cases} \frac{r}{2}(\dot{\phi}_2 - \dot{\phi}_1) = V_d \\ \frac{-r}{2W}(\dot{\phi}_2 + \dot{\phi}_1) = \dot{\theta}_d \end{cases} \;\; \text{then} \;\; \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} = \begin{bmatrix} -\frac{r}{2} & \frac{r}{2} \\ \frac{-r}{2W} & \frac{-r}{2W} \end{bmatrix}^{-1} \begin{bmatrix} V_d \\ \dot{\theta}_d \end{bmatrix} \tag{51}$$

Here, $\dot{\theta}_d$ and V_d used in the equation make the angular velocities desired ($\dot{\phi}_{d,1}$ and $\dot{\phi}_{d,2}$) since they are used in the creation of $\dot{\phi}_1$ and $\dot{\phi}_2$. Therefore, the desired angular velocities ($\dot{\phi}_{d,1}$ and $\dot{\phi}_{d,2}$) can also be found with Eq. 51. Thus, $\dot{q}_d$ can be obtained, and $\ddot{q}_d$ and q_d expressions can be obtained by taking the integral and derivative, respectively. In addition, it is possible to create the outer loop control shown in Eq. 39. After the controller application is implemented, a Matlab script with defined parameters is written to basically provide the simulation of the equations of motion and it is provided to call the equations of motion in the Simulink model. As shown in Fig. 4, the inner loop that provides the rover movement is the outer loop control. In addition, it provides the $\dot{q}_d$ generator points containing the path traversal equations. From here, the initial conditions of the rover are double integrated and the resulting $\ddot{q}_d$ generator is obtained.

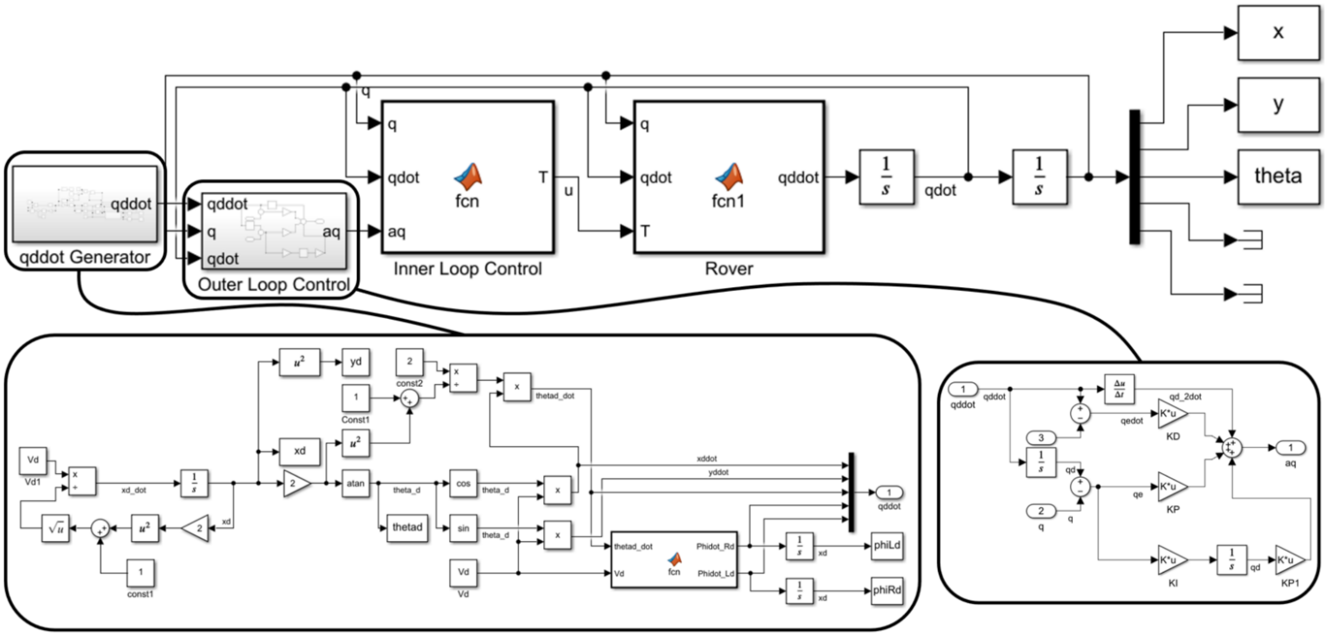

Fig. 4. Simulink model of the system.

All positions and velocities expressed here belong to a second-order system. When the rover is used, recognition allows the vehicle to pass through all desired paths from a certain start to a certain end. However, since this creates the problem of which path the vehicle should take, it is necessary to define either directly based on the path planning algorithm or mathematically the path it wants to take. As a result, the control design of a differential drive robot has been realized to follow a certain path autonomously. The design under this heading is basically based on Eq. 34. Thus, the application of multi-variable central control inner and outer loop control to the differential drive space exploration rover has been explained. The numbers used here and the conditions are presented and simulated to show that the initial system works. In this context, the mass of the rover's chassis is 30 kilograms, the mass of each wheel is 1 kilogram, the width of the body (2w) is 150 cm, the radius of each tire is 15 cm, the thickness of each wheel or tire is 2.5 cm, the offset between the center of gravity of the body and the center of the wheels (d) is 30 cm and the mass moment of inertia for the body is 15.625 kilogram-meter-square. In the simulation model established in the Simulink environment, 3 different orbits were used in order to have different dynamics of the orbits: sinusoidal orbit, infinite orbit and square orbit.

Fig. 5.a shows the X and Y axis errors with the rover's sinusoidal orbit tracking. The obtained results show that the rover converges to the reference orbit in a short time. As seen in the figure, the rover managed to settle into the reference orbit in a short time with the selected parameters. The position error was eliminated in about 5 seconds. It took about 1 second for the estimated parameters to become stable after the rover settled into the orbit and became stable. Similarly, Fig. 5.b shows the X and Y axis errors with the rover's infinite orbit tracking. As seen in the graph, the rover managed to settle into the reference orbit in a short time with the selected parameters. Although the position error was eliminated in about 7 seconds, adopting a dynamic structure causes an instant increase in the X axis error at the 30th second. It took about 0.4 seconds for the estimated parameters to become stable after the rover settled into the orbit and became stable. It was observed that all parameters were asymptotically stable. In addition, the rover's square orbit tracking and X and Y axis errors are shown in Fig. 5.c. As seen in the graph, the rover managed to settle into the reference orbit with the selected parameters, but due to the dynamic movements at the corner points, the rover made position errors in the axes. These errors are also very obvious in the graphs. These errors were resolved in a short time and the reference orbit tracking was successfully performed again. In the selected square orbit, there are only dynamic movements at certain positions, and in other positions there are only movements with linear speed. In places where the orbit is stationary, the estimation parameters also remained stationary.

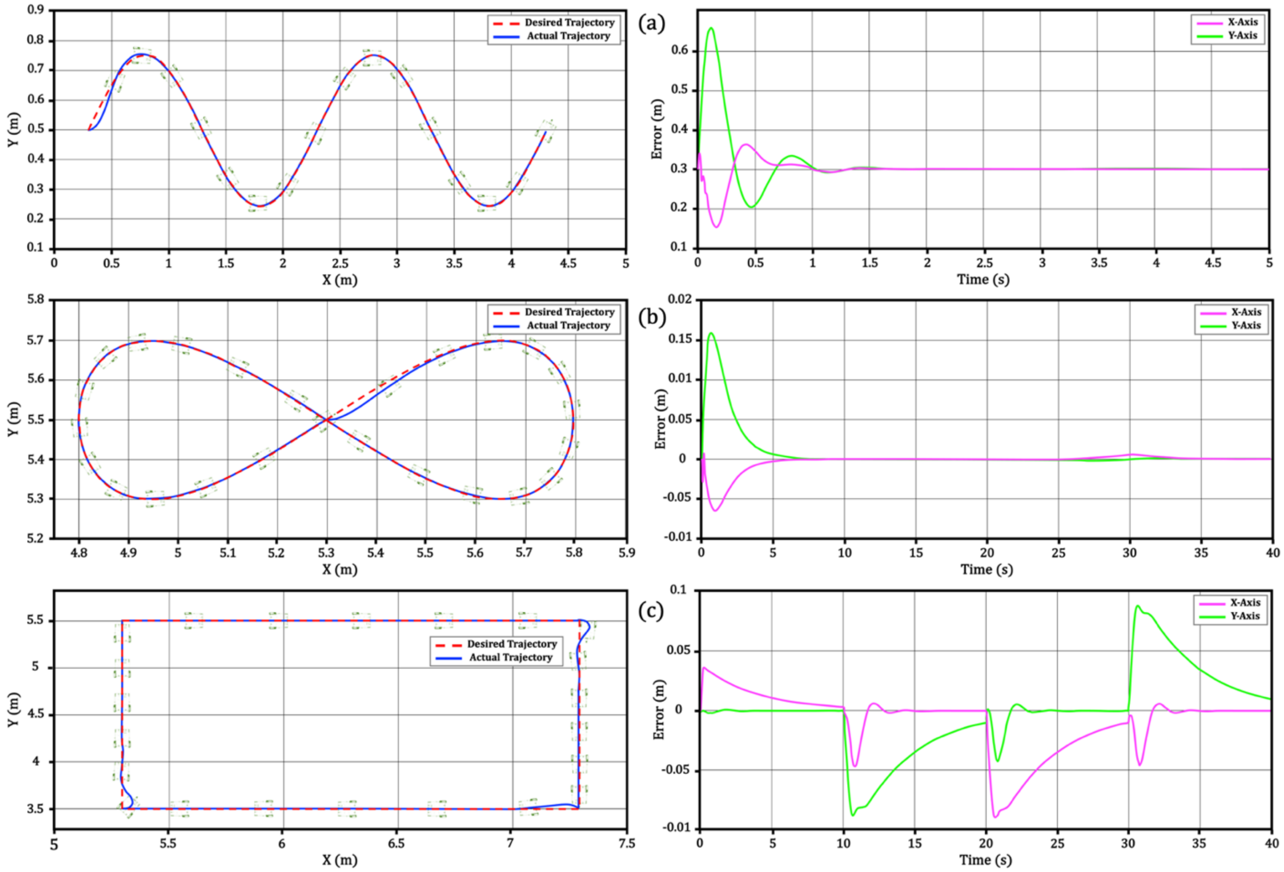

Fig. 5. Orbit tracking and axis errors.

As seen in Fig. 6.a, the instantaneous angular velocity for the sinusoidal orbit was able to settle at the targeted angular velocity in a very short time, while it took about 0.4 seconds for the linear velocity to stabilize. However, as seen in Fig. 6.b, the instantaneous angular velocity and linear velocity for the infinite orbit were able to settle at the targeted velocities in about 0.2 seconds. Similarly, as seen in Fig. 6.c, the instantaneous angular velocity and linear velocity for the square orbit were able to settle at the targeted velocities in about 0.2 seconds. Simulation studies were carried out in the Simulink environment and analyses were carried out in 3 different orbits, and it was observed that the rover moving in the infinite orbit minimized the position and speed errors in a shorter time compared to the sinusoidal orbit and became stable in a shorter time in parallel. In addition, it was observed that the angular velocity in the square orbit changed less over time. Therefore, it was realized that stable orbits such as square orbits were not suitable for applications where parameter estimations would be made. In this context, it was clearly seen that the estimation success of sinusoidal and infinite orbits was much higher.

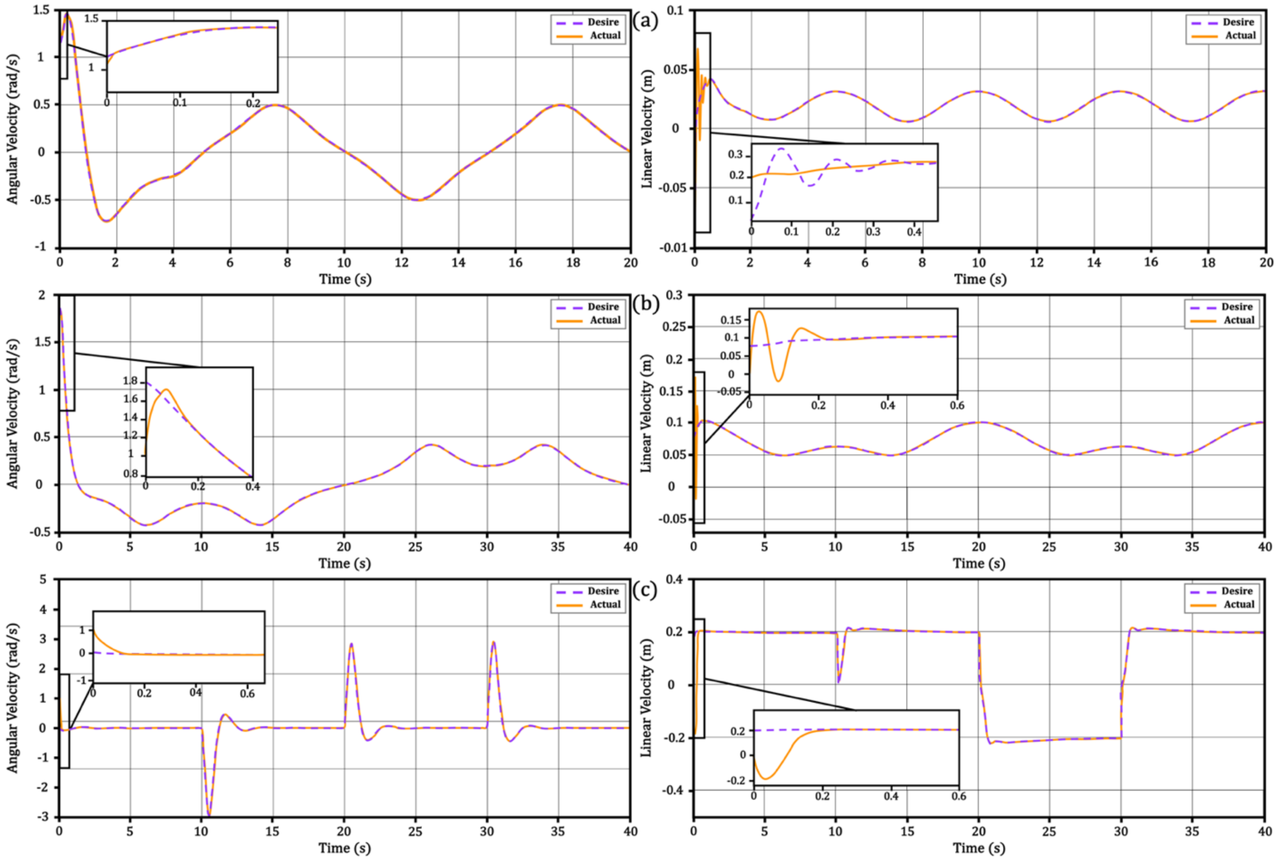

Fig. 6. 2D path planning for roadmaps with different obstacles.

6 Path Planning

The Rapidly-Exploring Random Tree (RRT) algorithm is a path planning method developed specifically for the purpose of addressing nonholonomic constraints. This approach rapidly expands the tree using control inputs. Unlike other traditional methods, RRT focuses on randomly selected points instead of directing the system from point to point. RRT offers a wide range of applications and can address holonomic, nonholonomic and kinodynamic planning problems. It determines the number and complexity of parameters in a system and is of great importance in engineering disciplines, especially mechanical engineering, aerospace engineering, robotics and structural engineering. The algorithm selects random points and branches towards these points while creating a path from the starting point to the target. However, it does not reprocess previously visited nodes and instead of considering a nearby node, it selects a random node on the map and starts from a different previously visited node to reach this node. The search process is completed when the point where the two trees meet is found and this point is the best path. In conclusion, the Rapidly-exploring Random Tree (RRT) algorithm provides a fast and efficient solution to nonholonomic planning problems. This algorithm plays a critical role in engineering applications by ensuring that systems successfully reach their goals through random selection and rapid expansion.

In this section, it is aimed to develop 3D path planning techniques based on 2D path planning algorithms designed for rovers and to make rapid progress in practice. Obstacles created in 2D environment contribute to obtaining the optimum path by preventing obstacles between the source and the target of the rover and using the recommended method. The RRT algorithm used for this purpose is one of the most widely used for autonomous rovers. The map contains three basic elements: obstacles, starting point and destination point, and is created using Matlab software. The algorithms include various steps for calculating the 3D path. First, information about obstacle heights is collected from a local starting point and transmitted to a target point. Then, the most suitable path for the rover to travel is determined based on obstacle heights and shortest path planning. Different alternative path plans are created using 2D path planning algorithms according to obstacle heights. In order to achieve these goals, 3 different areas are determined and the optimum path is selected as a result of running the RRT algorithm 10 times. As shown in Fig. 7, obstacles, each with a different length and width, are determined. The first and third obstacles in Fig. 7.a are 20 meters wide and 60 meters long, and the second obstacle is 40 meters wide and 40 meters long. The path from the starting point to the target point is calculated in a 2D environment. Similarly, as shown in Fig. 7.b, there are four different obstacles, each with a different length and width. The first obstacle is 20 meters wide and 20 meters long, the second obstacle is 20 meters wide and 23 meters long, the third obstacle is 20 meters wide and 40 meters long, and the fourth obstacle is 20 meters wide and 50 meters long. There is also a single obstacle, as shown in Fig. 7.c. The obstacle is 50 meters wide and 60 meters long.

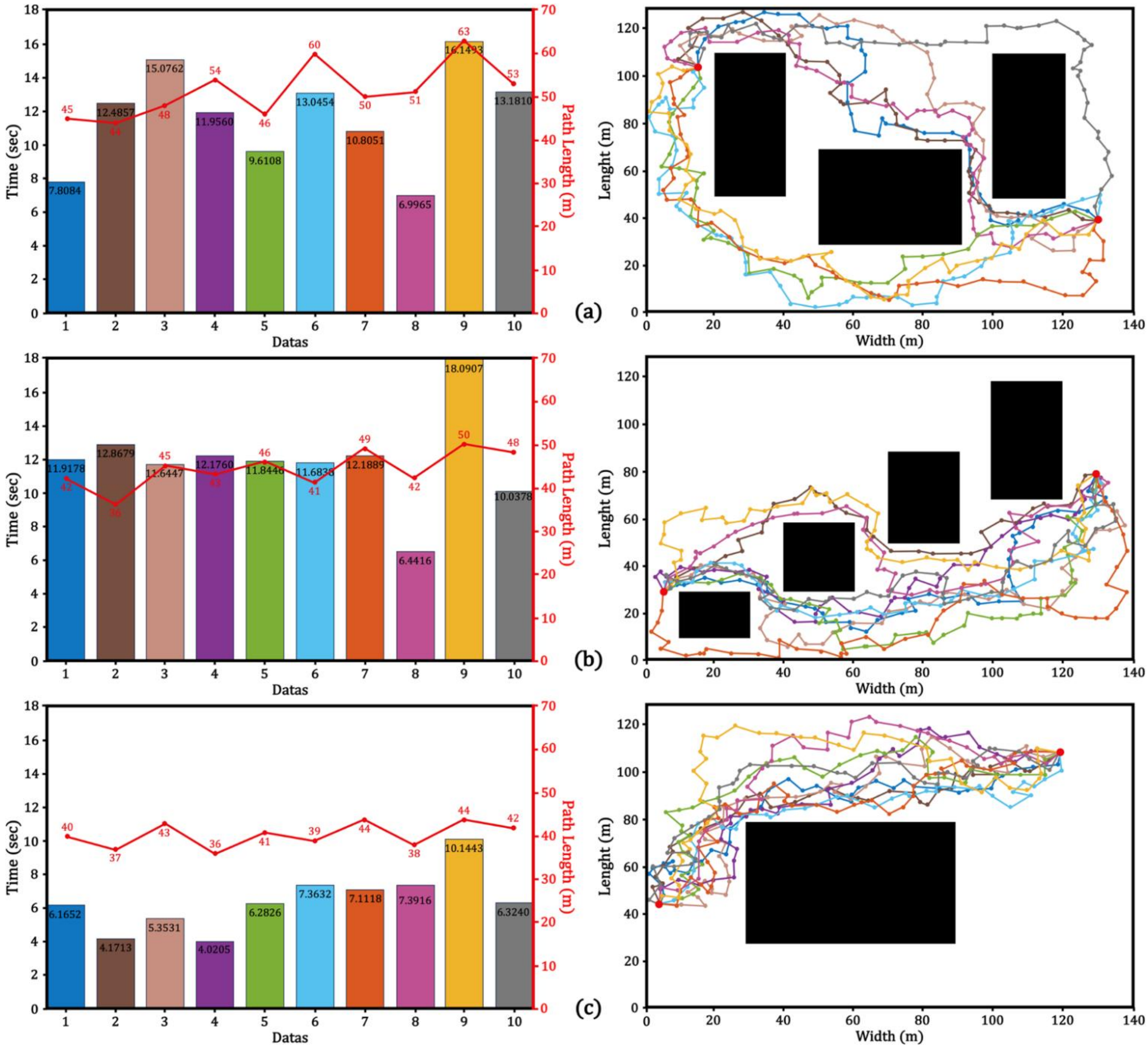

Fig. 7. 2D path planning for roadmaps with different obstacles.

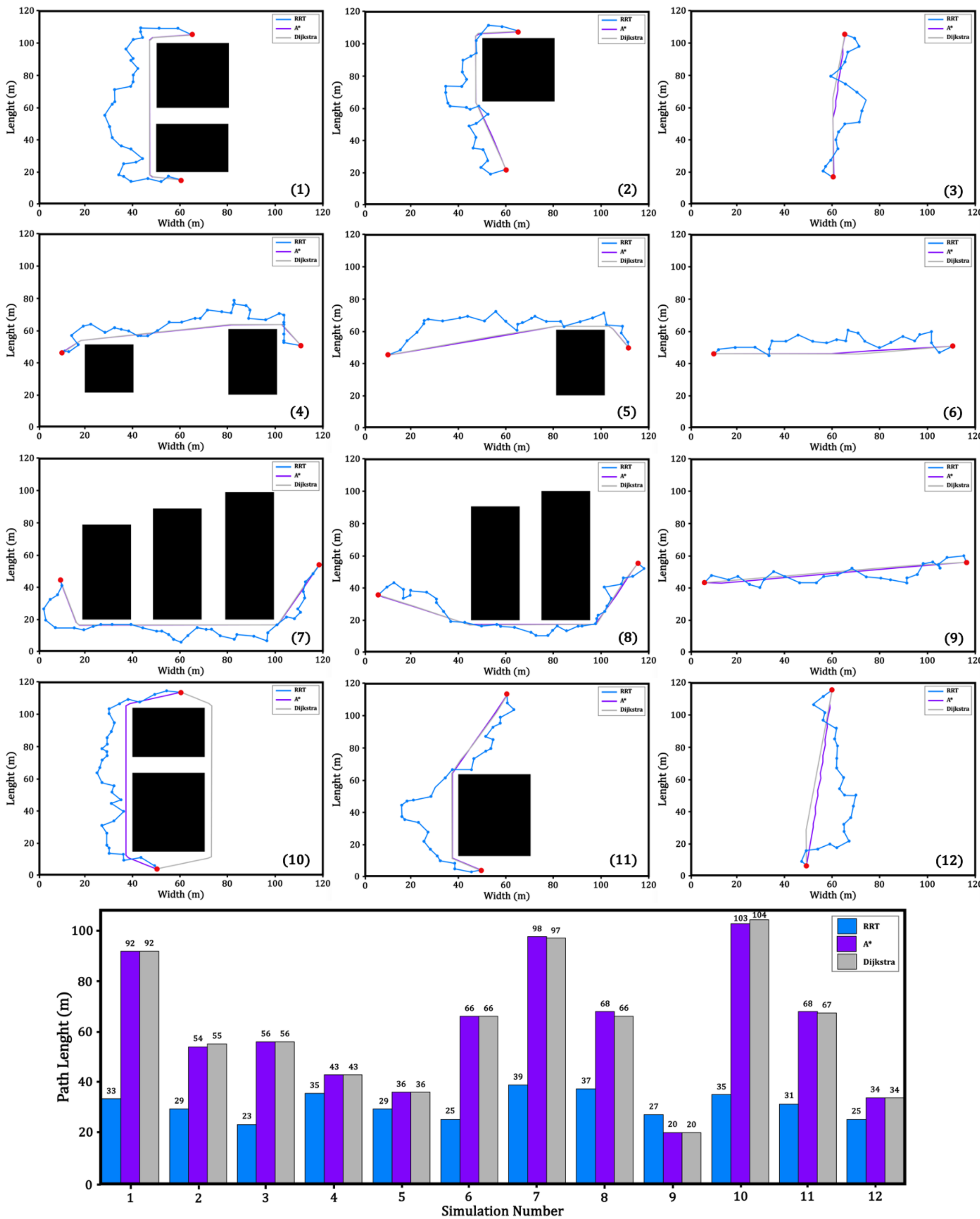

Fig. 8. Road performance of road maps with different obstacles compared to the algorithms in the literature.

For Fig. 7.a, it was seen that the path plan drawn by the first simulation among 10 iterations performed for the optimization of RRT was a more suitable path in terms of length and time. In a similar comparison, it was seen that the path plan drawn by the eighth simulation among 10 iterations in Fig. 7.b was a more suitable path in terms of length and time. Finally, it was seen that the path plan drawn by the fourth simulation among 10 iterations in Fig. 7.c was a more suitable path in terms of length and time. These situations showed that the RRT algorithm drew a path by determining different points towards the target in each simulation. This clearly showed with simulations that the RRT algorithm made branches by selecting random points and created the most suitable route to the target according to the branches.

After the RRT path planning control was performed on the differential drive rover, it was compared with the Dijkstra and A* algorithms, which have a place in the literature for finding the shortest distance between two nodes, in order to highlight the benefits of the algorithm used. Fig. 8 shows the success of determining the shortest path to the target with the application of RRT compared to other algorithms. The obtained data showed that the A* algorithm scans the area only towards the target, while the Dijkstra algorithm scans a much wider area and reaches the target. The RRT algorithm, on the other hand, draws a path by determining different points towards the target in each simulation. The main reason for this approach is that the RRT uses random samples from the search area while creating a rooted tree in the initial configuration. While drawing each sample, it tries to establish a connection between the tree and the nearest state. If this connection passes through a completely empty area and does not violate any constraints, the new state is added to the tree. However, if the random sample is further away from the nearest state in the tree, a new state at the maximum distance in the tree is used instead of the random sample. In this way, the random samples are allowed to control the tree growth and determine the growth direction.

7 Conclusions

This study emphasizes that autonomous path planning for differentially driven space exploration rovers can be successfully achieved using a controller with nonlinear governing equations and RRT (Rapidly-Exploring Random Tree) path planning. In order to test different dynamics in the simulation environment, the simulation model was studied on three different orbits: sinusoidal orbit, infinite orbit and square orbit. The space rover quickly adjusted to the reference orbit in the sinusoidal orbit and eliminated the position error in approximately 5 seconds. Also, the instantaneous angular velocity reached the target values in 0.1 seconds and the linear velocity in 0.4 seconds. In the infinite orbit, the reference orbit settlement time was approximately 7 seconds, while both angular and linear velocities were stabilized in 0.2 seconds. In the square orbit, although the reference orbit was generally settled, and dynamic movements, especially at the corner points, caused temporary position errors, these errors were corrected in a short time and the velocities reached the target values in 0.2 seconds. The findings show that the success of the parameter estimation depends on the stability in the orbit and speed tracking. In the infinite orbit in particular, the fact that the position and speed errors were minimized in a shorter time compared to the sinusoidal orbit allowed the estimation parameters to stabilize more quickly. These results are an important step for space exploration rovers to be used more efficiently in complex tasks. In this context, advanced interfaces and communication systems for conditions have been integrated with Software-Defined Radio (SDR) technology with communication bands such as S-band, VHF, UHF, and the communication capabilities of the rovers have been significantly increased. What's more, advanced imaging systems such as different cameras and LIDAR systems have helped the rovers to examine their surroundings in more detail and safely. Furthermore, for optimum path planning, it aimed to determine the shortest path to the target by avoiding obstacles using various two-dimensional methods that evaluate different conditions and requirements. Among the methods used for this purpose, Dijkstra algorithm determines the shortest distance between two nodes with a greedy approach, while A* algorithm determines this distance using a heuristic method. RRT algorithm reveals the most suitable route to the target by creating branches over randomly selected points. Simulation results showed that A* algorithm performs area scanning only focused on the target, RRT algorithm draws a path over different points in each trial, and Dijkstra algorithm reaches the target by scanning a wider area. As a result, RRT algorithm provided a shorter path to reach the target compared to A* method. This situation shows that RRT is more effective compared to previously widely used path planning algorithms such as A* and Dijkstra in the literature, and this algorithm has great potential in shaping the future strategies of autonomous space exploration.

In future studies, a cooperative approach based on swarms of mobile robots or rovers is proposed to enable space exploration rovers to gain more autonomous capabilities and efficiency. This could enable rovers to perform complex tasks more effectively. In this context, smaller swarm members orbiting a main rover could be used to create a more detailed map of the Earth's surface or to perform specific tasks. These swarm members could communicate with the main rover to share data and improve the rover's decision-making processes. Such a swarm-based cooperation could contribute to more efficient and successful future space exploration missions. Moreover, this approach could further expand the boundaries of space exploration technology by offering exciting opportunities to study the interactions between different types of rovers.

References

1. Azkarate, M., Gerdes, L., Wiese, T., Zwick, M., Pagnamenta, M., & et al. (2022). Design, testing, and evolution of mars rover testbeds: european space agency planetary exploration. *IEEE Robotics & Automation Magazine*, 29(3), 10-23, https://doi.org/10.1109/MRA.2021.3134875
2. Gage, J.L., Richards, E., Lepak, N., Kaczmar, N., Soman, C., & et al. (2019). In-field whole-plant maize architecture characterized by subcanopy rovers and latent space phenotyping. *Plant Phenome Journal*, 2(1), 1-11. https://doi.org/10.2135/tppj2019.07.0011
3. Xu, W., Liu, X., Yan, Z., Li, L., Zhang, Z., & et al. (2021). The MarSCoDe ınstrument suite on the mars rover of China's Tianwen-1 mission. *Space Science Reviews*, 217(5), 1-58. https://doi.org/10.1007/s11214-021-00836-5
4. Fries, M.D., Lee, C., Bhartia, R., Razzell Hollis, J., Beegle, L.W., & et al. (2022). The SHERLOC Calibration Target on the Mars 2020 perseverance rover: Design, operations, outreach, and future human exploration functions. *Space Science Reviews*, 218(6), 1-33. https://doi.org/10.1007/s11214-022-00907-1
5. Arreguin, A.L., Montenegro, S., & Dilger, E. (2021). Towards in-situ characterization of regolith strength by inverse terramechanics and machine learning: A survey and applications to planetary rovers. *Planetary and Space Science*, 204, 105271, 1-10. https://doi.org/10.1016/j.pss.2021.105271
6. Stack, K.M., Williams, N.R., Calef, F., Sun, V.Z., Williford, K.H., & et al. (2020). Photogeologic map of the perseverance rover field site in jezero crater constructed by the mars 2020 science team. *Space Science Reviews*, 216(8), 1-47. https://doi.org/10.1007/s11214-020-00739-x
7. Morse, Z.R., Harrington, E., Hill, P.J.A., Christoffersen, P., Newman, J., & et al. (2019). The use of GIS, mapping, and immersive technologies in the CanMars Mars Sample Return analogue mission; advantages for science interpretation and operational decisi on-making. *Planetary and Space Science*, 168, 15-26. https://doi.org/10.1016/j.pss.2019.01.001
8. Bell, J.F., Maki, J.N., Mehall, G.L., Ravine, M.A., Caplinger, M.A., & et al. (2021). The Mars 2020 perseverance rover mast camera zoom (mastcam-z) multispectral, stereoscopic ımaging ınvestigation. *Space Science Reviews*, 217(1), 1-40. https://doi.org/10.1007/s11214-020-00755-x
9. Creech, S., Guidi J., & Elburn, D. (2022). Artemis: An overview of nasa's activities to return humans to the moon. In *2022 IEEE Aerospace Conference (AERO),* Big Sky, MT, USA, (pp. 1-7). IEEE Press. https://doi.org/10.1109/AERO53065.2022.9843277
10. Sharma, M., Gupta, A., Gupta, S.K., Alsamhi, S.H., & Shvetsov, A.V. (2022). Survey on unmanned aerial vehicle for mars exploration: Deployment use case. *Drones,* 6(1):4, 1-23. https://doi.org/10.3390/drones6010004
11. Wiens, R.C., Maurice, S., Robinson, S.H., Nelson, A.E., Cais, P., & et al. (2020). The supercam ınstrument suite on the nasa mars 2020 rover: body unit and combined system tests. *Space Science Reviews*, 217(4), 1-87. https://doi.org/10.1007/s11214-020-00777-5
12. Liang, X., Chen, W., Cao, Z., Wu, F., Lyu, W., & et al. (2021). The Navigation and Terrain Cameras on the Tianwen-1 Mars Rover, *Space Science Reviews*, 217(3), 1-20. https://doi.org/10.1007/s11214-021-00813-y
13. Maki, J. N., Gruel, D., McKinney, C., Ravine, M. A., Morales, M., & et al. (2020). The mars 2020 engineering cameras and microphone on the perseverance rover: A next-generation ımaging system for mars exploration. *Space Science Reviews*, 216(8), 1-48. https://doi.org/10.1007/s11214-020-00765-9
14. Maurice, S. Wiens, R.C., Bernardi, P., Caïs, P., Robinson, S., & et al. (2021). The SuperCam Instrument Suite on the Mars 2020 Rover: Science Objectives and Mast-Unit Description. *Space Science Reviews*, 217(3), 1-108. https://doi.org/10.1007/s11214-021-00807-w
15. Farley, K.A., Williford, K.H., Stack, K.M., Bhartia, R., Chen, A., & et al. (2020). Mars 2020 mission overview. *Space Science Reviews*, 216(8), 1-41. https://doi.org/10.1007/s11214-020-00762-y
16. Rodriguez-Manfredi, J.A., de la Torre Juárez, M., Alonso, A., Apéstigue, V., Arruego I., & et al. (2021). The Mars Environmental Dynamics Analyzer, MEDA. A Suite of Environmental Sensors for the Mars 2020 Mission. *Space Science Reviews*, 217(48), 1-86. https://doi.org/10.1007/s11214-021-00816-9
17. Jaramillo-Morales, M. F., Dogru S., & Marques, L. (2020). Generation of energy optimal speed profiles for a differential drive mobile robot with payload on straight trajectories. In *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Abu Dhabi, United Arab Emirates, (pp. 136-141). IEEE Press. https://doi.org/10.1109/SSRR50563.2020.9292590
18. Khalil, A., Mukhopadhyay, S. & Rehman, H. (2022). Comparative performance analysis of PI controller and MRAC for a differential drive robot. In *5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Cairo, Egypt, (pp. 1-6). https://doi.org/10.1109/ICCSPA55860.2022.10019165
19. Iqbal, J., Xu R., Halloran H. & Li, C. (2020). Development of a multi-purpose autonomous differential drive mobile robot for plant phenotyping and soil sensing. *Electronics*, 9(9), 1-19. https://doi.org/10.3390/electronics9091550
20. Thoesen, A., & Marvi, H. (2021). Planetary surface mobility and exploration: A review. *Current Robotics Reports*, 2(3), 239-249. https://doi.org/10.1007/s43154-021-00056-3
21. Shivam, Kumar J.A., Subramaniam P., Mercy L., & Raj S.D. (2020). Design and fabrication of smart rover using rocker bogie mechanism. *AIP Conference Proceedings*, 2311(1), (2020). https://doi.org/10.1063/5.0034190
22. Pflueger, M., Agha A., & Sukhatme, G. S. (2019). Rover-IRL: Inverse reinforcement learning with soft value ıteration networks for planetary rover path planning. In *IEEE Robotics and Automation Letters*, 4(2), 1387-1394. https://doi.org/10.1109/LRA.2019.2895892
23. Bai, J.H., & Oh, Y. J. (2020). Global path planning of lunar rover under static and dynamic constraints. *International Journal of Aeronautical and Space Science*, 21(4), 1105–1113. https://doi.org/10.1007/s42405-020-00262-x
24. Xia, Y., & Shen, G. (2018). A novel learning-based global path planning algorithm for planetary rovers. *Neurocomputing*, 361, 69-76. https://doi.org/10.1016/j.neucom.2019.05.075
25. Yu, X., Wang, P., & Zhang, Z. (2021). Learning-based end-to-end path planning for lunar rovers with safety constraints. *Sensors*. 21(3):796, 1-16. https://doi.org/10.3390/s21030796

26. Jin, X., Lan, W., Wang, T., & Yu, P. (2021). Value iteration networks with double estimator for planetary rover path planning. *Sensors*. 21(24), 1-16. https://doi.org/10.3390/s21248418
27. Jung, M., Chuen Tan, K., & Dai R. (2022). Path planning for a jumping rover team with a charging station in multi-waypoints visiting missions. *Frontiers in Control. Enginnering*. Vol. 3, 1-15. https://doi.org/10.3389/fcteg.2022.803468
28. Hedrick, G., Ohi, N., & Gu, Y. (2020. Terrain-aware path planning and map update for mars sample return mission. *IEEE Robotics and Automation Letters*, 5(4), 5181-5188. https://doi.org/10.1109/LRA.2020.3005123
29. Sánchez-Ibánez, J.R., Pérez-del-Pulgar, C.J., Carlos J. Azkarate, M., & Gerdes, L. (2019). Dynamic path planning for reconfigurable rovers using a multi-layered grid. *Engineering Applications of Artificial Intelligence*, Vol. 86, 32-42. https://doi.org/10.1016/j.engappai.2019.08.011
30. Jiang, J., Zeng, X., Guzzetti, D., & You, Y. (2020). Path planning for asteroid hopping rovers with pre-trained deep reinforcement learning architectures. *Acta Astronautica*, Vol. 171, 265-279. https://doi.org/10.1016/j.actaastro.2020.03.007
31. Wang, Y., Wan, W., Gou, S., Peng, M., & Liu, Z., (2020). Vision-based decision support for rover path planning in the Chang'e-4 mission. *Remote Sensing*, 12(4) 1-16. https://doi.org/10.3390/rs12040624
32. Chen, G., You, H., Huang, Z., Fei, J., & Wang, Y. (2022). An efficient sampling-based path planning for the lunar rover with autonomous target seeking. *Aerospace*, 9(3), 1-27. https://doi.org/10.3390/aerospace9030148
33. Lou, Q., González F. & Kövecses, J. (2019). Kinematic modeling and state estimation of exploration rovers. *IEEE Robotics and Automation Letters*, 4(2), 1311-1318. https://doi.org/10.1109/LRA.2019.2895393
34. Cardoso, R. C., Kourtis, G., Dennis, L. A., Dixon, C., & Farrell, M. (2021). A review of verification and validation for space autonomous systems. *Current Robotics Reports*, 2(3), 273-283. https://doi.org/10.1007/s43154-021-00058-1
35. Wu, W., Xie, X., Wei, M., Liu, N., & Chen, X. (2019). Planetary rover path planning based on improved A* algorithm. In *Intelligent Robotics and Applications: 12th International Conference, ICIRA 2019*, Shenyang, China, August 8–11, 2019, Proceedings, Part IV 12 (pp. 341-353). Springer International Publishing. https://doi.org/10.1007/978-3-030-27538-9_29
36. Wang, X., Liu, Y., Zhou, R., She, J., & Qi, N. (2021, May). Dynamic modeling and path planning of jump-wheeled lunar rover. In *2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Nanchang, China, (pp. 529-534). IEEE. https://doi.org/10.1109/YAC53711.2021.9486667
37. Paton, M., Strub, M. P., Brown, T., Greene, R. J., & Lizewski, J. (2020, October). Navigation on the line: Traversability analysis and path planning for extreme-terrain rappelling rovers. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Las Vegas, NV, USA, (pp. 7034-7041). IEEE. https://doi.org/10.1109/IROS45743.2020.9341409
38. Wang, Y., Yuan, Q., Guo, Y., & Han, W. (2021, July). Path planning for lunar rover based on bi-rrt algorithm. In *2021 40th Chinese Control Conference (CCC),* Shanghai, China, (pp. 4211-4216). IEEE. https://doi.org/10.23919/CCC52363.2021.9549340
39. Lanfeng, Z., Lina, Y., & Hua, F. (2019). Lunar rover path planning based on comprehensive genetic algorithm based on slip prediction. *Journal of Physics: Conference Series, IOP Publishing*, 1267(1), 1-6. https://doi.org/10.1088/1742-6596/1267/1/012097
40. Tao, Z., Zhang, W., Jia, Y., & Chen, B. (2022, November). Path Planning Technology of Mars Rover Based on Griding of Visibility-graph Map Direction Search Method. In *2022 China Automation Congress (CAC)*, Xiamen, China, (pp. 6703-6707). IEEE. https://doi.org/10.1109/CAC57257.2022.10055733
41. Wenyuan, Z., Jifeng, G., & Chengchao, B. (2019, October). Adaptive Path Planning for Unmanned Planetary Rover with Dynamic Obstacle. In *2019 IEEE International Conference on Unmanned Systems (ICUS),* Beijing, China, (pp. 730-735). IEEE. https://doi.org/10.1109/ICUS48101.2019.8996050
42. Katiyar, S., & Dutta, A. (2022). Dynamic path planning over CG-Space of 10DOF rover with static and randomly moving obstacles using RRT* rewiring. *Robotica*, 40(8), 2610-2629. https://doi.org/10.1017/S0263574721001843
43. Hu, T., Cao, T., Zheng, B., Zhang, H., & Ni, M. (2021, October). Large-scale Autonomous Navigation and Path Planning of Lunar Rover via Deep Reinforcement Learning. In *2021 China Automation Congress (CAC)*, Beijing, China, (pp. 2050-2055). IEEE. https://doi.org/10.1109/CAC53003.2021.9728075