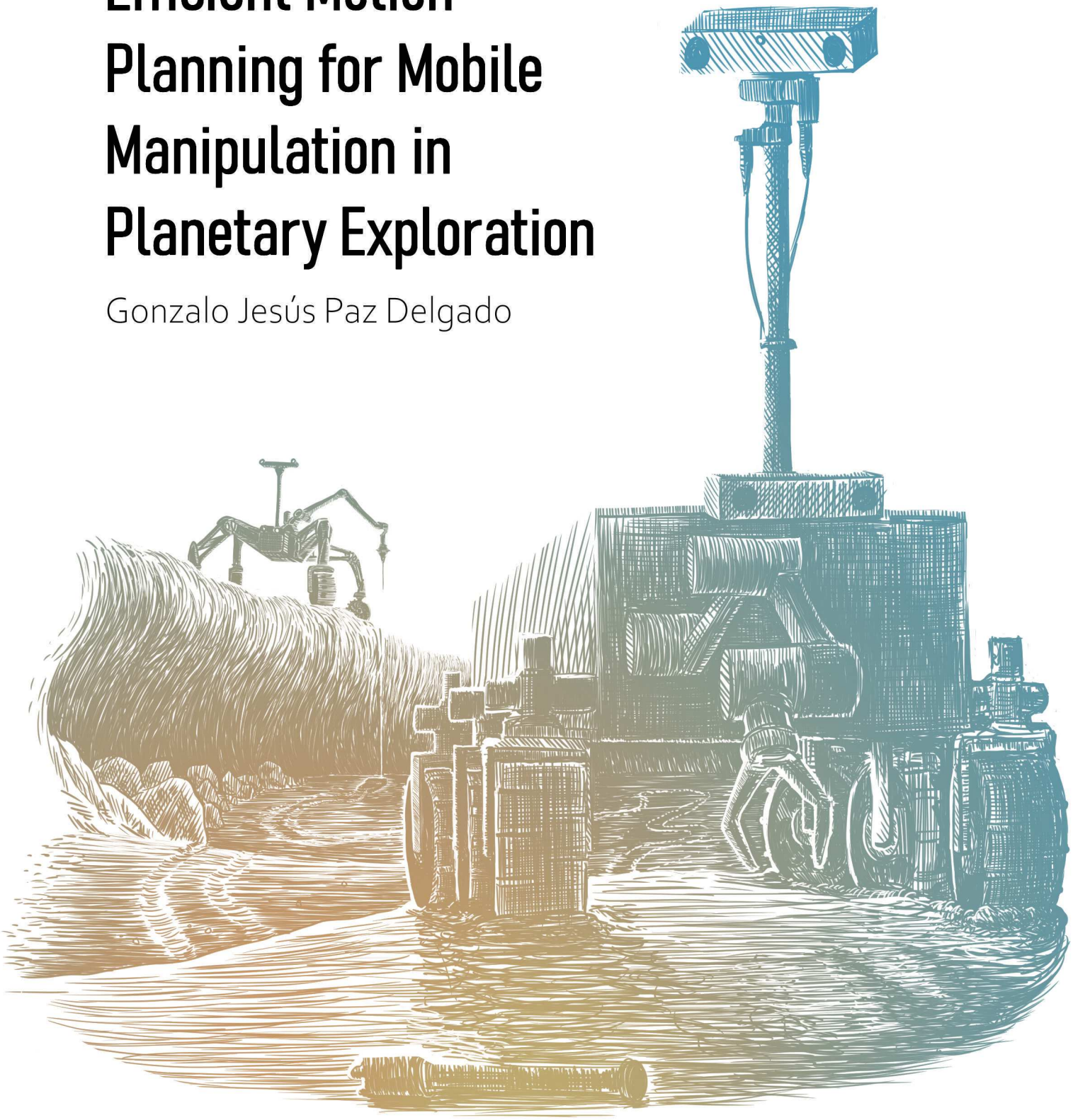
PHD THESIS

Efficient Motion Planning for Mobile Manipulation in Planetary Exploration

Gonzalo Jesús Paz Delgado



SPACE ROBOTICS LAB
UNIVERSIDAD DE MÁLAGA
dfki ai

DOCTORAL THESIS

Efficient Motion Planning for Mobile Manipulation in Planetary Exploration

Author:
Gonzalo Jesús Paz Delgado

Málaga, Spain, April 3rd, 2024

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

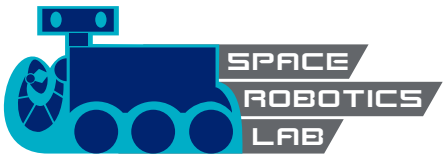
as an international cotutelle, with joint supervision

Prof. Dr. Carlos Jesús Pérez del Pulgar
Space Robotics Laboratory
ISA
Escuela de Ingenierías Industriales
Universidad de Málaga

Doctorate in Mechatronics Engineering

Prof. Dr. Dr. h.c. Frank Kirchner
Robotics Innovation Center
DFKI
Fachbereich 3 - Mathematik und Informatik
Universität Bremen

Doctorate in AI-based Space Robotics



dfki
ai

To those who doubt, make mistakes, rectify, learn, and continue.

Para todo aquel que duda, se equivoca, rectifica, aprende, y continúa.

Acknowledgements

When I look back, I do not remember so bad the bad moments, nor that good the good ones. For better or for worse, I have always had a bad memory. I do not remember so dark the frustration when the goals seemed unfeasible, nor that brilliant the satisfaction after achieving them. However, I do remember clearly those moments I shared with each person who walked beside me on this journey. I do hold the mark from every little push, every kind smile, and every encouraging word, so I could keep advancing with a firm foot on this path that I chose. That is why, when I look back, I only think of showing my gratitude. I apologize since I have always been a bit intense and I have a lot to say. So here we go.

First of all, thank you, Carlos. I started this journey because of you, and I have come this far thanks to you. Every step I have taken on this path has been possible because of the opportunities you have gained and given to me. Thank you for your tireless ambition, for your always constructive criticism, and above all, for your constant, kind, and respectful support. What I have of an engineer or researcher, undoubtedly, I learned from you. Thank you, Martin. How easy it is to work by your side! You are the ideal leader, motivating and energizing anyone with just a few words, with that spontaneity and simplicity that characterize you. Thanks to you I fulfilled the dream of working at ESA, and I overcame every obstacle during my stay there. I hope to work under your supervision again. Thank you, Raúl. If I went to Bremen and survived, it was thanks to you. When I arrived there, you welcomed me, accompanied me, and advised me. You integrated me into work and also into your social life, even though we had just met. If I grew so much there, it was thanks to you. I can never thank you enough. Thank you, Frank, for opening the door to this joint doctorate and that stay in Bremen, which I will always remember as one of the most beautiful stages of my life.

Thanks to my family. Everything I do is for and because of you. Thanks to my parents for always supporting me, without judgment and without restrictions. Thanks, Mom. Thank you for being my reference for love, affection, and kindness. Every time I doubt, I always remember what you told me once: "Never get tired of being good". It's not easy, but every day you show that it can be achieved. I will try to follow in your footsteps. Thanks, Dad. If I am organized and a perfectionist, it is because of you. If I can stay calm even in a storm, it is because of you. If I am curious and analytical, it is because of you. If I like to make people laugh by saying something outrageous from time to time, yes, it is also because of you. Thanks to my siblings for always being together. Thanks, Vichy. You will always be my little sister, even though you are much braver and stronger than me. Thanks for giving me so much love and attention. Thanks for trying to open up this tough mind that I have. Thanks for the sweet gift of your company. Thanks, Manolon. Your inner world is so immense that it overflows, and you can capture me in it. You take me back to those days when the only thing that mattered was dreaming, letting the imagination fly,

and enjoying. I rest best when I am with you: all problems disappear. Thanks to my grandparents and uncles. From you, I learned generosity and boundless kindness. Thanks to my cousins, who have been models for me to follow. Thanks, Ale. You are almost like a brother to me. You have always been the mirror with which to compare myself, to (positively) feedback my energies, goals, likes, and will to keep growing. We will get even further, together. Thanks, Álvaro and Laure. Since I was I child you were the example of good guy I wished to be: kind, strong-willed, competitive, independent, always happy. I still admire you today. Thanks also to the relatives with whom I do not share blood. Thanks, Antonio, and thanks, Mari Carmen. Although you probably do not know each other, you have one thing in common: you have brought happiness to my parents with your company, always being attentive, generous, and respectful to my siblings and me. Thanks for being part of my family. Thanks, Lourdes. From you, I learned never to give up, no matter how many blows I receive. I wish I could tell you about all the challenges I have overcome.

Thanks to my space robotics colleagues in Málaga, with whom I have achieved so many goals through sweat and tears. Thanks, Ricardo, for being a role model, and for leaving a trail in the lab to make the path easier for those who came after. Thanks, Laura. You spread your energy and enthusiasm to everybody around you, work is not work when you are nearby. I treasure all the moments with you in Málaga, Holland, Bremen, and Lanzarote. Thanks, Raúl. You came in so strong that I had no choice but to push myself not to fall behind. Thanks for being so critical and nonconformist; there is no other way to improve what surrounds us. Thanks, Nacho. You have that different spark in your way of being that makes me doubt and rethink who I am, and thus continue to grow. Thanks, Palomeque. Thanks for your support. Although I do not smoke, every stop for a "purillo" with you has allowed me to breathe and gather the strength to move forward. Thanks, JC. Thanks for that easy laughter that cheers everybody up even in the most complicated days. Thanks to my colleagues in the Robotics and Automation Section of ESA, particularly Levin and Tim. Only the best make it to ESA. But I do not say this because you have an incomparable talent with rovers (which you do), but because of your quality as humans. You make anyone feel at home with your friendliness and kindness. Thanks to my colleagues at DFKI RIC in Bremen, especially to Jonathan and Shubham. I have really enjoyed your company. Life changes are so easy if you find people as close, attentive, and cheerful as you. Thanks also to my colleagues in the CoRob-X project, especially to Alex. Thanks to your calm, kindness, and good management, I was able to enjoy in Lanzarote the three most incredible weeks of my career.

Thanks also to the family we do choose. Thanks, Rodri. Thanks for this beautiful friendship since we were two years old. Thanks for keeping the flame just as alive despite the distance. Thanks for teaching me to be strong. Thanks for lifting me when I fall and for celebrating with me every time I climb a new step. Today, it is

time to celebrate. Thanks, Álvaro, a.k.a. Chivo. Thanks for being so immensely kind and generous even with everything you have gone through. Thanks for that innocent humor that makes anyone laugh. Thanks for consoling me even when you were the one who needed consolation. Thanks, Alfre. You are that light that never goes out. You are the definition of someone good, simple, and happy. I do not remember hearing you speak badly of anyone, ever. I wish I could learn from you that beautiful way of living life. Thanks for always being there. Thanks, Alcaide. It is clear that we are different, we fight, bully each other, and get mad at each other. But that is because of the love and respect we have for each other. There is no one with whom I can have such deep conversations as with you, and at the same time, laugh out such loud. I hope you are happy, now you can officially call me a "mad scientist". Thanks, Boned, Lope, Ipe, Pepe, Raya, Mérida, Belén, Félix. I am grateful for having so many and such good friends. I apologize for not mentioning you all as you deserve. Thanks also to my friends from Bremen, Paula, Salva, and Marta, for bringing light to my stay there and turning it into one of the most beautiful stages of my life. I did not want to leave. Thanks, Carmen. Although our paths separated some time ago, if I have come this far, it is largely thanks to you. Your affection and warmth gave me the peace and tranquility that I needed to focus all my energies on keep growing, and on pushing my studies and career.

Finally, a special thanks to my grandfather Juan José. Abu, you were the instigator of my scientific curiosity, of my eagerness to know why things are as they are, and of my passion for understanding physics in nature. I hope my parents (who are both doctors) will not take this too bad, but how could my siblings and I be anything other than engineers, having a granddad like you? If only I could receive another email from you, with some geometry problem to break my head. If only you were here to see me defend this doctoral thesis.

Agradecimientos

Cuando miro atrás, no recuerdo tan malos los malos momentos, ni tan buenos los buenos momentos. Por suerte o por desgracia, siempre he tenido mala memoria. No recuerdo tan oscura la frustración cuando los objetivos parecían inalcanzables, ni tan brillante la satisfacción al verlos cumplidos. Sin embargo, sí que recuerdo con claridad los momentos que he compartido con cada persona que me ha acompañado en este recorrido. Sí que mantengo la huella que marcó cada pequeño empujón, cada amable sonrisa, cada palabra de ánimo, para que yo siguiera avanzando con paso firme en el camino que elegí. Por eso, cuando miro atrás, solo pienso en dar gracias. Disculpad, pues siempre he sido un poco intenso y tengo mucho que decir. Así que vamos allá.

En primer lugar, gracias Carlos. Empecé este camino gracias a ti, y he llegado hasta aquí gracias a ti. Cada paso que he dado en este camino ha sido posible por las oportunidades que tú has conseguido y me has otorgado. Gracias por tu incansable ambición, por tu siempre constructivo sentido crítico, pero sobretodo por tu apoyo constante, amable y respetuoso. Lo que tengo de ingeniero o investigador, sin duda, lo he aprendido de ti. Gracias Martin. Qué fácil es trabajar a tu lado. Eres el líder ideal: capaz de motivar y llenar de energía a cualquiera sólo con unas palabras, con esa naturalidad y sencillez que te caracterizan. Contigo cumplí el sueño de trabajar en la ESA, y gracias a ti superé allí cada obstáculo que se me presentaba. Ojalá volver a trabajar bajo tu supervisión. Gracias Raúl. Si pude ir y sobrevivir en Bremen, fue gracias a ti. Cuando llegué allí me acogiste, me acompañaste y me aconsejaste. Me integraste en el trabajo y también en tu vida social, y eso que nos acabábamos de conocer. Si crecí tanto allí, fue gracias a ti. Nunca podré agradecértelo lo suficiente. Gracias Frank, por abrirme la puerta a este doctorado en cotutela y a esa estancia en Bremen, que siempre voy a recordar como una de las etapas más bonitas de mi vida.

Gracias a mi familia. Todo lo que hago es por y para vosotros. Gracias a mis padres por apoyarme siempre, sin juicios y sin restricciones. Gracias Mamá. Gracias por ser mi referente de amor, cariño y amabilidad. Cada vez que dudo, siempre me acuerdo de ti, de eso que me dijiste una vez: "nunca te canses de ser bueno". No es nada fácil, pero cada día tú demuestras que se puede lograr. Intentaré seguir tus pasos. Gracias Papá. Si soy ordenado y perfeccionista, es por ti. Si soy capaz de mantener la calma incluso en una vorágine, es por ti. Si soy curioso y analítico, es por ti. Si me gusta hacer reír soltando alguna barbaridad de vez en cuando, sí, también es por ti. Gracias a mis hermanos, por mantenernos siempre juntos. Gracias Vichy. Siempre vas a ser mi hermanita pequeña, aunque seas mucho más valiente y fuerte que yo. Gracias por darme tanto cariño y atención. Gracias por intentar abrir un poco esta mente terca que tengo. Gracias por el dulce regalo de tu compañía. Gracias Manolon. Tienes un mundo interior tan inmenso que te desborda, y a mí eres capaz de atraparme en él. Me llevas de vuelta a esos días en los que lo único que importaba era soñar, dejar volar la imaginación, y disfrutar. Cuando mejor descanso es cuando

estoy contigo: todos los problemas desaparecen. Gracias a mis abuelos y a mis tíos. De vosotros aprendí la generosidad y la bondad sin límites. Gracias a mis primos, ellos que han sido mis modelos a seguir. Gracias Ale. Eres casi un hermano para mí. Siempre has sido el espejo con el que compararme para realimentar (positivamente) mis energías, mis objetivos, mis gustos, y mis ganas de seguir creciendo. Espero que lleguemos aún más lejos, juntos. Gracias Álvaro y Laure. Desde pequeño fuisteis el ejemplo de buena persona en quien me quería convertir: amables, luchadores, competitivos, independientes, siempre alegres. Os sigo admirando hoy día. Gracias también a los familiares con los que no comparto sangre. Gracias Antonio, y gracias Mari Carmen. Aunque probablemente no os conocéis, tenéis una cosa en común: habéis traído felicidad a mis padres con vuestra compañía, siempre siendo atentos, generosos y respetuosos con mis hermanos y conmigo. Gracias por formar parte de mi familia. Gracias Lourdes. De ti aprendí a no rendirme nunca, por muchos golpes que reciba. Ojalá pudiera contarte todos los retos que he superado.

Gracias a mis compañeros robóticos espaciales de Málaga, con los que he alcanzado tantos éxitos a base de sudor y lágrimas. Gracias Ricardo por ser un referente, por dejar una estela en el laboratorio para hacer más fácil el camino a los que veníamos detrás. Gracias Laura. Contagias tu energía y tu entusiasmo a todo el que te rodea, trabajar no es trabajar si tú estás cerca. Atesoro todos los momentos contigo en Málaga, Holanda, Bremen, y Lanzarote. Gracias Raúl. Llegaste pisando tan fuerte que no tuve otra que apretar para no quedarme atrás. Gracias por ser tan crítico e inconformista, no hay otra forma para mejorar lo que nos rodea. Gracias Nacho. Tienes esa chispa diferente en tu forma de ser que me hace dudar y replantearme lo que soy, y así seguir creciendo. Gracias Palomeque. Gracias por tu apoyo. Aunque yo no fume, cada parada para un purillo contigo me ha permitido respirar y coger fuerzas para seguir adelante. Gracias JC. Gracias por esa risa fácil que alegra el ambiente hasta en los días más complicados. Gracias a mis compañeros de la Robotics and Automation Section de la ESA, particularmente a Levin y a Tim. Se nota que a la ESA solo llegan los mejores. Pero no lo digo porque tengáis un talento incomparable con los rovers (que lo tenéis), sino por vuestra calidad humana. Hacéis sentir a cualquiera como en casa, con vuestra simpatía y sencillez. Gracias a mis compañeros del DFKI RIC en Bremen, en concreto a Jonathan y Shubham. Cómo he disfrutado vuestra compañía. Qué fácil son los cambios en la vida si al moverte encuentras personas tan cercanas, atentas y amables como vosotros. Gracias también a mis compañeros del proyecto CoRob-X, y en especial a Alex. Gracias a tu calma, paciencia, amabilidad y buena gestión, pude disfrutar en Lanzarote de las tres semanas más increíbles de mi carrera.

Gracias también a la familia que elegimos. Gracias Rodri. Gracias por esta amistad tan bonita desde que teníamos dos años. Gracias por mantener la llama igual de viva a pesar de la distancia. Gracias por enseñarme a ser fuerte. Gracias por levantarme cuando caigo, y por celebrar conmigo cada vez que subo un nuevo escalón. Hoy nos toca celebrar. Gracias Álvaro, a.k.a. Chivo. Gracias por ser tan

inmensamente bondadoso y generoso aun con todo lo que has tenido que pasar. Gracias por ese humor inocente que hace reír a cualquiera. Gracias por consolarme incluso cuando eras tú el que necesitaba consuelo. Gracias Alfre. Eres esa luz que nunca se apaga. Eres la definición de alguien bueno, sencillo, feliz. No recuerdo oírte hablar mal de nadie, nunca. Ojalá aprender de ti esa forma tan bonita de vivir la vida. Gracias por estar siempre ahí. Gracias Alcaide. Está claro que somos diferentes, chocamos, nos metemos el uno con el otro, y nos enfadamos. Pero eso es por el cariño y respeto que nos guardamos. No hay nadie con quien pueda mantener conversaciones tan profundas como contigo, y al mismo tiempo reírme a carcajadas. Espero que estés contento: ahora ya puedes llamarme científico loco oficialmente. Gracias Boned, Lope, Ipe, Pepe, Raya, Mérida, Belén, Félix. Tengo que dar gracias por tener tantos y tan buenos amigos. Disculpad por no mencionaros a todos como os merecéis. Gracias también a los Bremenses, Paula, Salva y Marta, por darle luz a mi estancia en Bremen y convertirla en una de las etapas más bonitas de mi vida. Realmente no quería irme de allí. Gracias Carmen. Aunque nuestros caminos se separaran hace un tiempo, si he llegado hasta aquí es en gran parte gracias a ti. Tu cariño y compañía me daban la tranquilidad y la paz necesarias para poder centrarme con todas mis energías en seguir creciendo, en seguir avanzando con mis estudios y mi trabajo.

Por último, un gracias especial a mi abuelo Juan José. Abu, tu fuiste el instigador de mi curiosidad científica, de mis ansias de saber el por qué de las cosas, de mi pasión por los fenómenos físicos de la naturaleza. Que me perdonen mis padres (que son los dos médicos), pero, ¿cómo íbamos a ser mis hermanos y yo otra cosa que ingenieros, teniendo un referente como tú? Ojalá recibir otro correo tuyo, con algún problema de geometría con el que romperme la cabeza. Ojalá pudieras estar aquí para verme defender esta tesis doctoral.

UNIVERSIDAD DE MÁLAGA
Escuela de Ingenierías Industriales
Space Robotics Laboratory, ISA
Doctorate on Mechatronics Engineering

UNIVERSITÄT BREMEN
Fachbereich 3 - Mathematik und Informatik
Robotics Innovation Center, DFKI
Doctorate on AI-based Space Robotics

Abstract

Efficient Motion Planning for Mobile Manipulation in Planetary Exploration

by Gonzalo Jesús Paz Delgado

Space is the main frontier to further expand our human civilization and our knowledge of the universe. Exploring space is challenging and expensive since humans are not biologically prepared to survive in the harsh conditions of space. Robots are a suitable alternative though, since they can reach and explore space in a much cheaper way and without risking human lives. In the last decades the so-called planetary exploration vehicles, or rovers, have been able to gather very interesting information about the Moon and Mars. They do so through scientific instruments, which are generally held by a robotic arm that places them in scientifically interesting targets on the surface. Rovers find a main drawback in remote teleoperation from Earth, given the huge challenges of transmitting information in space (delays, limited communication windows). Therefore, the scientific return of planetary missions is greatly increased if the rover can perform the task autonomously, i.e. without the need for human intervention.

Considering the mobility provided by the robot's locomotion system and the manipulation capabilities of the robotic arm, planetary rovers can be viewed as mobile manipulators. Since a great amount of the scientific tasks that a rover performs include mobile manipulation movements, autonomously executing them would raise the scientific return of the planetary exploration mission. Though autonomous navigation on Mars is already quite advanced, demonstrated for instance with the Perseverance rover, autonomously performing mobile manipulation tasks is still troublesome. Such a challenge is mainly caused by the complexity of planning and controlling the movements of the mobile platform and the robotic arm together, which cannot be easily achieved in space due to the limited computational resources available out of the Earth.

Hereby, this doctoral thesis presents the research conducted by the author to achieve autonomous mobile manipulation in planetary exploration vehicles. The goal is to plan and control the motion of the robot efficiently, i.e. with low computational cost, to perform the mobile manipulation scientific tasks. To do so, on-Earth technology has been analyzed, modified, and applied to the space exploration

field, with four main contributions. First, the global path planner methodology to reach the area of scientific interest is enhanced, feeding back local information from the rover to increase its awareness of the scenario. Second, a computationally lightweight and collision-free motion planner for mobile manipulation is presented, which ensures the safety of the robot while performing the mobile manipulation scientific tasks. Third, an optimal-control-based motion planner for mobile manipulators is proposed, which can maximize the motion efficiency and comply with the system constraints, at the same time it keeps a low computational cost thanks to a series of warm start stages, which makes it suitable for space. Fourth, a model predictive controller for mobile manipulation is used to increase the safety of multi-robot missions, in the particular use case of rappelling into lava tubes. All of these contributions have been thoroughly analyzed and validated by means of experimental campaigns, which include simulation, laboratory, and field tests with rover prototypes in analogue scenarios to Mars or the Moon.

UNIVERSIDAD DE MÁLAGA
Escuela de Ingenierías Industriales
Space Robotics Laboratory, ISA
Doctorate on Mechatronics Engineering

UNIVERSITÄT BREMEN
Fachbereich 3 - Mathematik und Informatik
Robotics Innovation Center, DFKI
Doctorate on AI-based Space Robotics

Sumario

Planificación Eficiente de Movimientos para Manipulación Móvil en Exploración Planetaria

by Gonzalo Jesús Paz Delgado

El espacio es la principal frontera para seguir expandiendo la civilización humana y nuestro conocimiento del universo. Explorar el espacio es difícil y costoso, ya que los humanos no estamos preparados biológicamente para sobrevivir en las duras condiciones que hay fuera de nuestro planeta. Sin embargo, los robots son una alternativa, dado que pueden moverse y explorar el espacio de una forma mucho más barata y sin arriesgar vidas humanas. En las últimas décadas los llamados vehículos de exploración planetaria, o rovers, han sido capaces de recoger información muy interesante sobre la Luna y Marte. Utilizan para ello múltiples instrumentos científicos, que generalmente están instalados en un brazo robótico. Este brazo coloca los instrumentos en objetivos de interés científico en la superficie extraterrestre. No obstante, los rovers tienen un gran inconveniente a la hora de ser teleoperados desde la Tierra, teniendo en cuenta lo difícil que es transmitir información en el espacio (retardos, ventanas de comunicación limitadas). Por ello, el retorno científico de las misiones planetarias se incrementa notablemente cuando el rover es capaz de realizar las tareas de manera autónoma, es decir, sin la necesidad de intervención humana.

Teniendo en cuenta la movilidad que proporciona el sistema de locomoción del rover y las capacidades de manipulación del brazo robótico, los rovers planetarios se pueden categorizar como manipuladores móviles. Ya que una gran cantidad de las tareas científicas que realizan los rovers incluyen manipulación móvil, ejecutarlas de una forma autónoma mejoraría notablemente el retorno científico de las misiones de exploración planetaria. Aunque la navegación autónoma en Marte está bastante avanzada como ha demostrado el rover Perseverance, realizar tareas de manipulación móvil de forma autónoma es aún un desafío. Esto se debe principalmente a la complejidad de planificar y controlar de manera conjunta los movimientos de la plataforma móvil y del brazo robótico, lo cual no se puede lograr de forma sencilla en el espacio debido a los limitados recursos computacionales disponibles fuera de la Tierra.

Con todo esto, esta tesis doctoral presenta la investigación realizada por el autor para alcanzar un mayor nivel de autonomía para manipulación móvil en vehículos de exploración planetaria. El objetivo es planificar y controlar los movimientos del robot de forma eficiente, es decir, con un coste computacional bajo, para llevar a cabo las tareas científicas de manipulación móvil. Para ello, la tecnología terrestre disponible ha sido analizada, modificada, y aplicada al campo de la exploración espacial, con cuatro contribuciones principales. Primero, se ha mejorado la metodología de planificación global de trayectorias para alcanzar el área de interés científico, realimentando información local del rover para incrementar su nivel de conocimiento sobre el escenario. Segundo, se presenta un planificador de movimientos seguro y eficiente para manipulación móvil, el cual garantiza que el robot no va a colisionar mientras realiza la tarea científica de manipulación móvil. Tercero, se propone un planificador de movimientos para manipuladores móviles basado en control óptimo, el cual es capaz de maximizar la eficiencia en los movimientos mientras cumple con las limitaciones del sistema, al mismo tiempo que mantiene un coste computacional bajo gracias a varias etapas de inicialización de la optimización. Este bajo coste computacional permite que el algoritmo sea usado en el espacio. Cuarto, un controlador predictivo por modelo es utilizado para incrementar la seguridad durante una operación de rappelling con manipulación móvil, en la cual un equipo de robots colabora para explorar un túnel Lunar de lava. Todas estas contribuciones se han analizado y validado en profundidad mediante campañas experimentales, incluyendo simulaciones, experimentos de laboratorio y experimentos de campo, utilizando varios prototipos de rovers en escenarios análogos a la Luna o Marte.

"Per aspera ad astra" -
"Through hardship to the stars"

Lucius Annaeus Seneca
Hercules furens, 1st century A.D.

Contents

List of Figures

List of Tables

List of Acronyms

ACE Approximate Clearance Evaluation. 32

ADE Autonomous Decision making for very long traverses. 12–15, 17, 19, 22, 102, 181, 183, 185, 189

AEGIS Autonomous Exploration for Gathering Increased Science. 5, 29, 174

AI Artificial Intelligence. 18, 20, 28, 29

API Application Programming Interface. 98, 105, 106, 114

APXS Alpha Proton X-Ray Spectrometer. 6, 7, 175

ARES Autonomous Routing on Extreme Surfaces. 13, 182

ASAN Advanced Small Analyzer for Neutrals. 8, 176

ASTRA Advanced Space Technologies in Robotics and Automation. 19, 20, 186–188

AutoNav Autonomous Navigation. 2, 4, 5, 32, 34, 35, 171–174

BGMR Bayesian Gaussian Mixture Regressor. 41

CBF Control Barrier Function. 46

ChemCam Chemistry and Camera complex. 29

CHIMRA Collection and Handling for In situ Martian Rock Analysis. 6, 175

CLF Control Lyapunov Function. 46

CNN Convolutional Neural Network. 122

CNSA Chinese National Space Agency. 8, 174, 176

CoRob-X Cooperative Robots for Extreme Environments. 13, 15, 16, 18, 20, 22, 128, 129, 138, 140, 181, 183–185, 187–189

CPU Central Processing Unit. 106, 108, 112, 114, 125, 140, 152

CSLQ Constrained Sequential Linear Quadratic regulator. 87–91, 123–125, 146

DART Dynamic Animation and Robotics Toolkit. 72, 73

DEM Digital Elevation Map. 15, 26, 28, 52–54, 67, 97, 98, 100, 103, 106, 107, 114, 115, 121, 183

DFKI German Research Center for Artificial Intelligence. 14, 15, 17, 96, 97, 102–104, 106, 130, 136, 137, 142, 145, 150, 151, 183–185, 193, 195

DoF Degrees of Freedom. 6–9, 11, 35, 38, 40, 41, 74, 80, 81, 84, 103, 104, 116–119, 129, 150, 175, 176, 178, 180
DP Dynamic Programming. 52
DRT Dust Removal Tool. 7, 175
DyMu Dynamic Multilayered path planner. 51–53, 63, 66, 67, 89, 97, 149, 192

EC European Commission. 12–14, 181, 183
ESA European Space Agency. 5, 7, 9, 12–14, 17, 19, 22, 96, 97, 122, 174, 176, 177, 181–183, 189
ESA-PRL Planetary Robotics Laboratory, Robotics and Automation Section, European Space Agency. 13, 14, 17, 19, 22, 96, 116, 117, 121–123, 144, 146, 149, 150, 182–185, 187, 189, 192, 194
ESTEC European Space Research and Technology Center. 13, 14, 17, 19, 96, 117, 122, 146, 182–184, 187
ETH Swiss Federal Institute of Technology. 9, 177
ExoTeR Exomars Testing Rover. 13, 14, 17–19, 22, 96, 116–119, 121, 123, 125, 127, 146, 150, 182, 185–187, 189, 194

FACE Frontal Approach Cost Editing. 68–70
FMM Fast Marching Method. 13, 19, 33, 51–53, 55–58, 66–70, 72, 74, 76, 86–89, 91, 93, 112, 113, 153, 182, 187
FoV Field of View. 44, 125
FPGA Field Programmable Gate Array. 5, 28, 30, 152, 153, 174
FSM Fast Sweeping Method. 34, 66, 113

GDM Gradient Descent Method. 55, 68
GNC Guidance, Navigation and Control. 9, 13, 18, 23–26, 97, 178, 182, 186
GNSS Global Navigation Satellite System. 29, 100, 104
GP Gaussian Process. 40
GPR Gaussian Process Regressor. 41
GPS Global Positioning System. 29, 104, 107, 114, 115

HAMP-BUA Hierarchical and Adaptive Mobile Manipulator Planner with Base pose Uncertainty and its propagation to Arm motions. 37
HDPR Heavy Duty Planetary Rover. 17, 22, 96, 97, 100–102, 144, 149, 184, 189, 192
HiRISE High Resolution Imaging Science Experiment. 26, 27, 29
HRSC High Resolution Stereo Camera. 26

IK Inverse Kinematics. 19, 72–74, 77, 79, 105

iLQR Iterative Linear Quadratic Regulator. 39

IMU Intertial Measurement Unit. 30, 104, 117, 122

IROS International Conference on Intelligent Robots and Systems. 17, 184

ISA Department of Systems Engineering and Automation. 12, 181

JIST Journal on Intelligent Service Robotics. 17, 18, 185

JPL Jet Propulsion Laboratory. 2, 3, 6, 7, 28, 34, 50, 172

k-NN k-Nearest Neighbor. 41

KUKA Keller und Knappich Augsburg. 44

LIDAR Light Detection and Ranging. 104, 114, 137, 145

LocCam Localization Camera. 104, 114, 117, 122, 125, 152

LQR Linear Quadratic Regulator. 39, 91, 122

LSC Last Section Control. 68, 70, 71, 79, 115

MAHLI Mars Hand Lens Imager. 7, 175

MARS Machina Arte Robotum Simulans. 106, 107, 112, 113, 136, 137

MCS Motion Control Subsystem. 105

MED Mediterranean Conference on Control and Automation. 19, 186

MER Mars Exploration Rover. 3, 4, 6, 30, 32, 172, 173, 175

MI Microscopic Imager. 6, 175

MIT Massachusetts Institute of Technology. 106, 116

MM Mobile Manipulation. 1, 5–12, 15, 17, 18, 21–25, 35–48, 51, 63–68, 71, 79, 80, 83, 94, 96, 102, 103, 105, 107, 109, 111, 113, 115, 129, 133, 136, 138–141, 143–153, 171, 175–181, 183, 185, 188, 189, 191–195

MPC Model Predictive Control. 12, 18, 22, 39, 42–46, 48, 88, 94, 96, 129, 133–136, 138–141, 143, 144, 146, 148, 151, 153, 181, 185, 189, 194, 195

MSL Mars Science Laboratory. 4, 6, 7, 173, 175

MSR Mars Sample Return. 5, 7, 18, 174–176, 186

MWMP Multi-staged Warm started optimization Motion Planner. 83, 86, 88, 90, 93, 94, 116, 118, 122, 123, 125, 127, 134, 135, 145, 146, 150, 151, 153, 193, 194

NASA National Aeronautics and Space Administration. 2–4, 6–8, 23, 28, 33, 34, 50, 171–173, 176
NavCam Navigation Camera. 104, 114, 117
NMPH Nonlinear Model Predictive Horizon. 45
NOAH-H Novelty or Anomaly Hunter-HiRISE. 27

ODE Open Dynamics Engine. 107, 136
OHMP Optimized Hierarchical Mobile Manipulator Planner. 38, 65
OpenCV Open Source Computer Vision Library. 67, 74
OUM Ordered Upwind Method. 34

PDE Partial Derivative Equation. 52
PM-PE Path and Motion planning in Planetary Exploration. 13, 14, 182, 183
PM-SFR Path and Motion planning for a Sample Fetching Rover. 13, 14, 182
PPWS Path Planning Warm Start. 86–88, 91
PRM Probabilistic RoadMaps. 38

RAT Rock Abrasion Tool. 6, 175
REU Robotic Explorer Units. 15, 20, 183, 187
RIC Robotics Innovation Center. 15, 17, 97, 103, 104, 106, 130, 136, 142, 145, 150, 151, 183–185, 193, 195
RL Reinforcement Learning. 46
ROCK Robot Construction Kit. 97, 105, 136, 140
RoI Region of Interest. 29
Roskosmos Russian Federal Space Agency. 5, 174
RP Rover Planner. 4, 173
RRC Robot Remote Control library. 105
RRT Rapidly exploring Random Trees. 37, 38, 112
RTK Real-Time Kinematic. 100, 104, 114

SA/SPaH Sample Acquisition, Processing, and Handling. 6, 175
SAS Space Application Services. 15, 183
SCS Sampling and Caching Subsystem. 7, 175, 176
SFR Sample Fetch Rover. 7, 14, 176, 182
SLAM Simultaneous Localization and Mapping. 13, 30, 31, 182
SLQ Sequential Linear Quadratic Regulator. 39, 44, 87, 88, 90–94, 146, 150, 153, 193

SOWG Science Operations Working Group. 4, 173
SPOC Soil Property and Object Classification. 27
SRC Strategic Research Cluster. 15, 183
SRL Sample Retrieval Lander. 7, 176
STOMP Stochastic Trajectory Optimization for Motion Planning. 40
SVM Support Vector Machine. 28

TMDS Tether Management and Docking System. 128–130, 135, 136, 141–143, 146
TrajOpt Trajectory Optimization. 40
TRL Technology Readiness Level. 152
TSL Taylor Series Linearization. 84, 119

UMA University of Málaga. 13, 14, 19, 182, 183, 187
UMA-SRL Space Robotics Laboratory from the University of Málaga. 12–15, 20, 97, 181–184, 187
URDF Unified Robot Description Format. 72, 73, 105, 106, 112, 136
USA United States of America. 17, 184
USLQ Unconstrained Sequential Linear Quadratic regulator. 87, 89–91, 123, 125

VIPER Volatiles Investigating Polar Exploration Rover. 8, 176
VKC Virtual Kinematic Chain. 41

List of Symbols

i, j, n, m, s	Indexes
p	Position vector, $p = [x\, y\, z]$ in 3D
ϕ	Orientation vector, $\phi = [\varphi\, \vartheta\, \psi]$ in 3D
$\tilde{p}_j$	Discrete node, the closest one to position p_j
$\tilde{p}_0$	Starting node
$\tilde{p}_g$	Goal node
$\tilde{p}_j^G$	Global node
Ω	Cost map
$\Omega(\tilde{p}_j)$	Individual cost associated to node $\tilde{p}_j$
Υ	Cost to go, the result of expanding the Fast Marching wave
$\Upsilon(\tilde{p}_j, \tilde{p}_g)$	Cost to go from $\tilde{p}_j$ to $\tilde{p}_g$
$\|\|\nabla\Upsilon(\tilde{p}_j, \tilde{p}_g)\|\|$	Rate of propagation of the Fast Marching wave at $\tilde{p}_j$
Γ	Optimal path
$\Gamma(\tilde{p}_0, \tilde{p}_g)$	Optimal path to get to $\tilde{p}_g$ from $\tilde{p}_0$
$\Gamma(\tilde{p}_0, \tilde{p}_g, l)$	Function returning a p given a particular path length l
l	Input path length from the starting node $\tilde{p}_0$
l_g	Total length of Γ
$\mathcal{T}$	Traversability map
ε_j^T	Number of subdivided local nodes from a global node $\tilde{p}_j^G$
ε_j^o	Number of local nodes occupied by obstacles within $\tilde{p}_j^G$
η	Hazard density
ξ	Trafficability
d_r	Traverse distance on the repaired path
d_o	Original traverse distance, without obstacles
a_m	Area differentiating a type of terrain in a segmentation map
$\mathbb{A}$	Set of areas in the scenario
$\mathbb{A}_U \subseteq \mathbb{A}$	Subset of terrain areas with the costs updated
M	Number of areas in the scenario
f_n	Terrain feature, e.g. slip ratio
$\mathbb{F}$	Set of terrain features
N_f	Number of terrain features
$\bar{c}_j^G$	Original cost of the global node $\tilde{p}_j^G$
c_j^G	Updated cost of the global node $\tilde{p}_j^G$
c_m	Cost of area a_m

$\mathbb{C}$	Set of estimated costs for each terrain patch
$\mathbb{C}_U \subseteq \mathbb{C}$	Subset of updated costs
$\hat{c}_m$	Cost ratio between terrain a_0 and terrain a_m
$\hat{c}_{min}$	Minimum cost ratio of the traversed terrains $\mathbb{A}_U$
c_{min}	Minimum cost of the traversed terrains $\mathbb{A}_U$
ς_m	Terrain hardness of area a_m
ϱ_n	Weight of terrain feature f_n in the final cost
$\mathbb{W}$	Set of weights of each terrain feature
σ	Slip ratio
$\overline{\sigma_m}$	Mean of all the samples of slip ratio in the area a_m
$\overline{\xi_m}$	Mean of all the samples of trafficability in the area a_m
$\overline{f_{m,n}}$	Estimated value of feature f_n in terrain area a_m
v_w	Surface speed of a wheel
v_r	Ground-truth speed of the platform
${}^{j}T_{\kappa}$	Transformation matrix from j to κ
Ω_B	Cost map of the mobile base
Ω_M	Cost volume for the manipulator
Ω_R	Reachability volume
Π	Slice of the reachability volume $\Pi \subset \Omega_R$
$\mathcal{D}_B$	Distance to obstacles map
$\mathcal{D}_B(\tilde{p}_j)$	Euclidean distance from the node $\tilde{p}_j$ to the closest non-traversable node in $\mathcal{T}$
$\mathcal{D}_M$	Distance to collisions volume
$\mathcal{D}_M(\tilde{p}_j)$	Euclidean distance from the node $\tilde{p}_j$ to the closest non-traversable node in W_R
$\dot{c}_B$	Cost sharpness for the mobile base
$\dot{c}_M$	Cost sharpness for the manipulator
Υ_B	Cost to go of the mobile base
Γ_B	Optimal path of the mobile base
$\Gamma_B(n)$	Waypoint of the mobile base path
N_B	Total number of waypoints of the mobile base path Γ_B
Γ_M	Manipulator path
$\Gamma_M(s)$	Waypoint of the manipulator path
S	Total number of waypoints of the manipulator path Γ_M
ι	Index of the base path Γ_B for the arm to start moving
$\Gamma_B(\iota)$	Waypoint where the arm should start moving
W	Manipulator workspace
W_R	Safe manipulator workspace considering self-collisions
Λ	Manipulator motion profile
q	Manipulator joints configuration, $q = [q^p\ q^o]$
q_n	Manipulator configuration at waypoint $\Gamma_B(n)$
q^p	Manipulator position joints

q^o	Manipulator orientation joints
$\mathbb{Q}^o$	Full range of the arm orientation joints
d	Deployment distance
d_e	Small deployment distance, i.e. end deployment
d_b	Big deployment distance, i.e. beginning deployment
$\mathbb{T}$	Set of time steps, a.k.a. time horizon
t_n	Time step
Δt	Time step size
N	Number of time steps
$x(n)$, $u(n)$	State and actuation vectors
$x'(n)$, $u'(n)$	Unconstrained state and actuation
$x''(n)$, $u''(n)$	Constrained state and actuation
x_0, x_N	Current and goal states of the platform
u_0	Initial control plan
$Q(n)$, $R(n)$	State and input quadratic costs
$A(n)$, $B(n)$	State space model matrices
$C(n)$, $D(n)$, $r(n)$	State-input constraint matrices
$G(n)$, $h(n)$	Pure state constraint matrices
κ	Kinematic chain
K	Number of kinematic chains
${}^jP_\kappa$, ${}^j\dot{P}_\kappa$	Pose, speed of κ w.r.t j
${}^j\mathcal{R}_\kappa$	Rotation matrix of κ w.r.t j
w	World reference frame
${}^wP_1(0)$, ${}^wP_1(N)$	Mobile platform start and goal poses
${}^wp_1(0)$, ${}^wp_1(N)$	Mobile platform start and goal positions
q_κ, $\dot{q}_\kappa$, $\ddot{q}_\kappa$	Position, speed and acc. of the joints in κ
$\|\|q_\kappa\|\|$	Number of Degrees of Freedom of kinematic chain κ
$\mathbb{I}$	Identity matrix
$\mathbb{I}_j$	Identity matrix with size jxj
${}^j\mathcal{J}_\kappa$	Jacobian matrix of κ w.r.t j
I_κ, V_κ	Inertia and Coriolis matrices of κ
e_κ	Actuation effort vector of κ
δ_z	External perturbations
Z	Number of external perturbations
γ_z^κ	Matrix representing the effect of δ_z into κ
β_z^κ	Auxiliary variable, $\beta_z^\kappa = -I_\kappa^{-1}\gamma_z^\kappa$
J	Total cost to go optimal control
$\Phi(x(N))$	Terminal cost optimal control
$L(x(n), u(n), n)$	Intermediate cost optimal control
$x^0(n)$, $u^0(n)$	State and input references or targets
$C_c(n)$, $D_c(n)$, $G_c(n)$	Active constraints
$\bar{x}_0(n)$, $\bar{u}_0(n)$	References for current iteration

$\hat{D}(n)$, $\hat{r}(n)$	Aux. constraints predefinition SLQ
$\hat{A}(n)$	Aux. state model predefinition SLQ
$\hat{Q}(n)$, $\hat{R}(n)$	Aux. costs predefinition SLQ
$\hat{x}^0(n)$, $\hat{u}^0(n)$	Aux. state input predefinition SLQ
$\hat{P}(n)$, $\hat{s}(n)$, $\hat{M}(n)$	Aux. backward pass variables SLQ
ζ	Pure state constraint
$\mathcal{C}_x$	Set of pure state constraints
$\Psi_c(n)$, $y_c(n)$, $h_c(n)$	Aux. matrices pure state constraints SLQ
$\Psi(\zeta)$, $y(\zeta)$, $\mathcal{H}(\zeta)$	Aux. matrices pure state constraints SLQ
$\mathcal{F}_{\zeta,j}(n)$, $\mathcal{F}(\zeta,j)$, $\nu(\zeta)$	Aux. matrices pure state constraints SLQ
$\bar{v}(n)$, $\mu(n)$	Lagrangian multiplier vectors
$\lambda(n)$	Lagrangian co-state vector
$\bar{x}(n)$, $\bar{u}(n)$	Step plan for current iteration
α	Line search appliance step
θ_d, θ_s, θ_m	Driving, steering, arm rotational joints
τ_d, τ_s, τ_m	Driving, steering, arm actuators torques
g	Gravity acceleration
ρ	Rolling resistance
m	Vehicle mass
d_w	Wheels diameter
N_w	Number of wheels
F_t	Tether tension
$F_e = [F_{ee,x}\ F_{ee,y}\ F_{ee,z}]$	Force acting into the tip of the arm
$F_w = [F_{w,x}\ F_{w,y}\ F_{w,z}]$	Force acting into the spherical wrist
$F_s = [F_{s,x}\ F_{s,y}\ F_{s,z}]$	Force acting into the force/torque sensor
$\tau_w = [\tau_{w,x}\ \tau_{w,y}\ \tau_{w,z}]$	Torque acting into the force/torque sensor
$\tau_s = [\tau_{s,x}\ \tau_{s,y}\ \tau_{s,z}]$	Torque acting into the force/torque sensor
$\tau_{e,y}$	Torque endured by the elbow joint
d_{ee-s}	Distance from the tip of the arm to the force/torque sensor
d_{s-w}	Distance from the force/torque sensor to the spherical wrist
d_{w-e}	Distance from the spherical wrist to the elbow
Δx, Δy	Relative displacements of the end effector, x and y axes
$\Delta\varphi$, $\Delta\vartheta$	Relative rotations of the end effector, x and y axes
k_x, k_y	Spring constants for the displacements of the end effector in the x and y axes
k_φ, k_ϑ	Spring constants for the rotations of the end effector in the x and y axes
v_u	Unwinding velocity of the tether
v_c	Linear speed of the rappelling rover
$\mathcal{P}$	Proportional gain of the tension controller

F_r	Reference tension for the tether
k	Spring constant of the simulated tether

Chapter 1

Introduction

"Sic itur ad astra" -
"Thus one journeys to the stars"

Virgil
Aeneis, 1st century B.C.

1.1 Robotics duty in space science

Space is the main unknown that bounds human understanding of the universe and technological progression. Exploring space is difficult, costly, and dangerous for humans. Hence, robots have arisen in the last decades as the most suitable alternative, reaching further, with fewer resources and without risking human lives. In particular, planetary exploration vehicles, also called rovers, have become the best option to carry the scientific instruments that gather information on extraterrestrial surfaces like Mars (Gao and Chien, 2017) or the Moon. Generally held by a robotic arm, these instruments allow the rover to perform Mobile Manipulation (MM) tasks such as performing scientific measurements on the surface (Lehner et al., 2018), extracting Mars soil samples (Williford et al., 2018), collecting those sample tubes (Merlo, Larranaga, and Falkner, 2013) or even collaborating between rovers (Brinkmann et al., 2018). This kind of robot is called, therefore, a mobile manipulator, i.e. a manipulator with no limitation in its workspace thanks to the flexibility given by the mobile platform.

Nevertheless, Martian rovers have a main drawback on remote teleoperation from Earth, due to the small duration of the communication window (16 h per day, delaying 40 min each day) and the vast latency (20 min on average) between Earth and Mars (Lester and Thronson, 2011). Teleoperating a rover on Mars is, thus, an incredibly challenging endeavor, with high risk and low efficiency. Consequently, the alternative to teleoperation is to include some autonomous decision-making onboard the rover, permitting the system to react to unexpected events, or even take mid- to high-level decisions on its own. This capability is generally called autonomous navigation. Although with low relevancy in the first planetary missions, autonomous navigation has gradually taken significance during the last decades. In

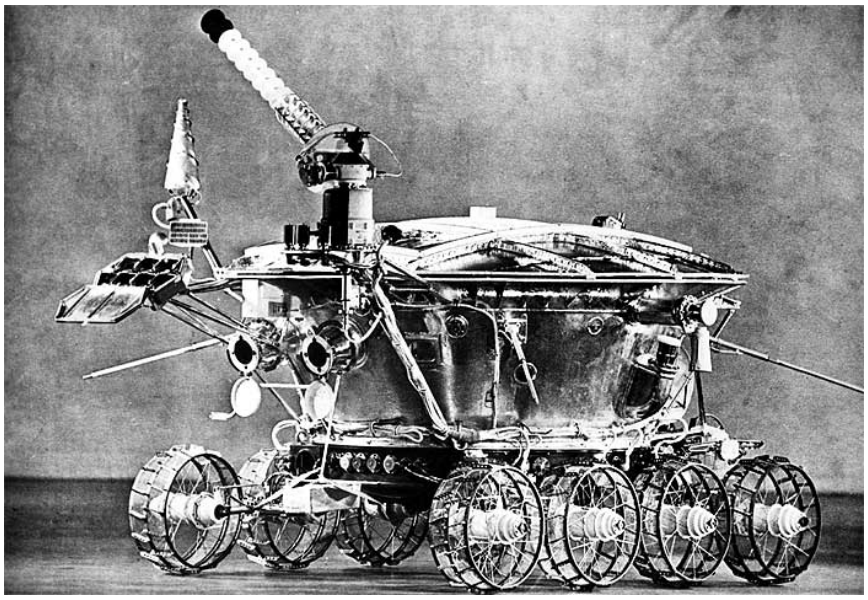


FIGURE 1.1: Lunokhod 1 of the Soviet Union, the first unmanned rover to ever traverse on an extraterrestrial surface (NASA/Goddard Space Flight Center/Arizona State University).

fact, on the last National Aeronautics and Space Administration (NASA) Martian rover, Perseverance, the so-called Autonomous Navigation (AutoNav) is achieving an astonishing 78 % self-driving distance on Mars during its first Martian year (Verma et al., 2023). Being such an autonomous rover, the driving distance has increased by 144 m on average per Martian sol (a Martian sol is a solar day on Mars, lasting 24 h and 39 min approximately) in comparison to previous missions, which usually couldn't achieve more than 30 m/sol. Although now impressive, these achievements are built on gradual research and testing of autonomous navigation capabilities thanks to the previous rovers. Hence, let us revisit the evolution of rovers and their autonomy throughout the history of planetary exploration.

Autonomous navigation on extraterrestrial surfaces

The concept of exploring extraterrestrial surfaces with robots was initially proposed with the Lunokhod (Moonwalker) program of the Soviet Union, being Lunokhod 1, shown in Figure 1.1, the first unmanned vehicle traversing any surface outside of the Earth (Kassel, 1970). This rover landed on the Moon in 1970 and traversed 10.54 km for eleven Lunar days, i.e. 321 Earth days. Its successor was the Lunokhod 2 rover (Severny, Terez, and Zvereva, 1975), which landed on the Moon in 1973 and covered approximately 39 km. Both rovers were completely teleoperated thanks to the small latency to the Moon (approx. 3 s). They managed to return thousands of images of the Lunar surface and also performed multiple soil analyses with spectrometers, penetrometers, and laser telemeters.

It was not until 25 years later that another robot was successfully deployed in an extraterrestrial body, with the NASA/Jet Propulsion Laboratory (JPL) mission

FIGURE 1.2: Selfie of Curiosity on Gale Crater (NASA/JPL-Caltech/Malin Space Science Systems).

Mars Pathfinder. This was a proof of concept about Martian exploration with robots, which was confirmed when the Carl Sagan Memorial Station landed on Mars in 1997 with the Sojourner rover (Matijevic et al., 1997). Although it could not cover more than a hundred meters, Sojourner included basic autonomous navigation functionalities, which established the first in-flight demonstration of this kind of technology. In particular, it had a "Go to Waypoint" method, which enabled the drivers to command a set of waypoints to be reached by the rover with straight lines. During the execution, the rover used wheel odometry to localize itself, and a series of sensors to detect proximity hazards and avoid them (Matijevic, 1998).

Later on, the overwhelming achievements of the NASA Mars Exploration Rover (MER)s (Maimone, Cheng, and Matthies, 2007), Spirit and Opportunity, demonstrated that planetary exploration vehicles guarantee a broad scientific return. Spirit and Opportunity initially had a planned 90-day mission lifetime when they landed on Mars in 2004, which was greatly overcome with the final duration of their journeys, six years for Spirit and 15 for Opportunity. These twin rovers had significantly improved capabilities w.r.t. Sojourner, with a ten times faster processor and locomotion (Rankin et al., 2023). They also had better autonomous navigation methods with

the so-called AutoNav, including three main features. First, visual odometry for localization, which provided 95 % successful convergence of the pose updates (Maimone, Cheng, and Matthies, 2007). Second, terrain assessment, which used stereo cameras to detect and estimate the position of nearby hazards. Third, a path selector, which could choose the best path to reach a goal among a series of Gaussian distributed paths (Biesiadecki and Maimone, 2006). Spirit and Opportunity used this AutoNav to autonomously cover 1825 m and 2433 m respectively, i.e. 23.60 % and 5.40 % of their total distance driven on Mars.

Performing more ambitious scientific tasks on the Martian surface became, after the great success of the MERs, the new goal. Particularly, the NASA Mars Science Laboratory (MSL) mission was launched in 2012 with the rover Curiosity (Anderson et al., 2012). This rover, shown in Figure 1.2, reached the Martian surface on the Gale Crater, where it keeps exploring nowadays. Curiosity includes a similar AutoNav to the MERs but with a six times faster processor. Nevertheless, the autonomous capabilities of Curiosity are not outstanding, covering autonomously only the 3.77 % of its total distance driven on Mars. This is a lower distance than the MERs but with much better capabilities, which can only be understood considering the strict scientific requirements for Curiosity.

Let us summarize how the drivers plan and command Curiosity each sol (Manning and Simon, 2014). First of all, there is a team that takes the high-level decisions called the Science Operations Working Group (SOWG). This team is composed of scientists, who propose different tasks to be carried out by the rover, and the rover engineers, who estimate which tasks are feasible considering the required time and energy. The SOWG members iterate several times, analyzing all the options until they decide on the exploration plan for the next Martian sol. Nevertheless, the SOWG has to finish with sufficient time before the Martian midnight. Otherwise, the Rover Planner (RP)s will not be able to take the SOWG-designed plan and translate it into a detailed series of low-level actions for the rover. These actions include, first, driving commands to get to some specific waypoints in the scenario, which are set by the RPs after looking at orbital and local imagery from the rover. Second, the driving commands give the lead to the scientific task, with specific commands for the robotic arm and the scientific instruments. The complete command sequence is, eventually, checked several times in a dedicated simulator. As a result, for each Martian sol, the RPs require approximately six hours to translate the plan from the SOWG to the detailed command list. Last of all, once Curiosity is woken up and working nominally, the set of instructions is sent in the early Martian morning, and Curiosity starts to follow the commands without any more communication until the end of the Martian sol. Remark that the rover does not know how to behave if anything goes wrong. In such a case, it would stop following the instructions until a new set of commands is received, which implies that the rover operation time for that sol would be lost.

Conclusively, even though Curiosity includes quite an advanced AutoNav for

hazard avoidance and reaching waypoints, the autonomous distance that it is able to traverse is highly constrained. On the one hand, due to the required direct teleoperation for task planning, scientific goals selection, and scientific task performance. But also, on the other hand, due to the available computational capacity onboard, which employs most of the processing time to keep all the engineering and scientific systems alive.

The autonomy drawbacks of Curiosity were later solved with the amazing autonomous navigation capabilities of the Perseverance rover, which landed on Mars in 2021. Perseverance is the first step of Mars Sample Return (MSR), probably the most ambitious planetary exploration program to date. MSR is formed by several missions with the final goal of bringing back to the Earth some samples of the Martian soil, which is very interesting from a scientific point of view, considering the highly advanced laboratories on our planet that could deeply analyze those samples. Perseverance is quite similar to Curiosity, inheriting most part of its developed technology, but with two computers and a Field Programmable Gate Array (FPGA) used for computer vision tasks. Hence, in Perseverance the autonomous navigation algorithms are not limited anymore by the computing time, being AutoNav able to sense the scenario and make decisions fast enough to move the rover autonomously at its maximum speed of 4.2 cm/s (Rankin et al., 2023). Additionally, Perseverance includes an Autonomous Exploration for Gathering Increased Science (AEGIS) algorithm, able to autonomously select and acquire images of scientifically interesting spots. Using this algorithm partially bypasses the need for Earth-driven science, enhancing noticeably the autonomy of the rover. In fact, Perseverance has driven 13 717 m in a completely autonomous manner, which is the 77.58 % of the total distance that the rover has traversed on Mars. This is 12 times the autonomous driven distance of Curiosity, and Perseverance has covered it in only two years in contrast to Curiosity, which has been exploring Mars for 11 years already.

Remark also the autonomy of some Martian rovers from other space agencies. First, the Chinese rover Zhurong has driven autonomously approximately the 40 % of the 450 m that this rover has traversed on Mars (Ding et al., 2022). Second, the Rosalind Franklin ExoMars rover (Vago et al., 2017). The European Space Agency (ESA) will launch this rover to Mars in 2028 in spite of the recent unfortunate events of the war in Ukraine, which led to interrupting the initial collaboration between ESA and the Russian Federal Space Agency (Roskosmos). ExoMars will include a full autonomy capability to get to commanded goals, by analyzing the terrain and planning a feasible path to reach the target (Winter et al., 2017).

It is clear, conclusively, that the autonomous navigation capabilities of rovers have evolved until reaching astonishing results with Perseverance. Nevertheless, apart from navigation, planetary rovers usually perform science through manipulation tasks, using their robotic arms to place a scientific instrument in an interesting spot of the scenario. Performing such MM tasks is still not autonomous, even for Perseverance. Besides, autonomous MM could enable new interesting tasks for

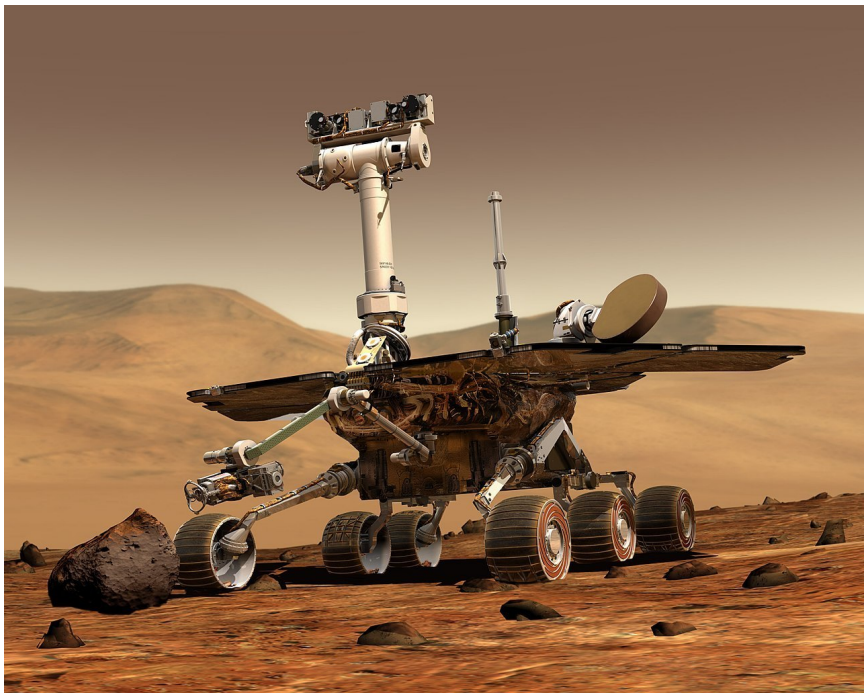


FIGURE 1.3: Representation of the twin rovers Spirit and Opportunity of the NASA mission MERs, including their robotic arm (NASA/JPL-Caltech).

planetary rovers, such as assembly or multi-robot collaboration. Let us hereby recap the past, present, and future of MM in planetary exploration and how autonomous MM could benefit the outcomes of planetary missions.

Planetary science and engineering through Mobile Manipulation

The first Martian rover, Sojourner, already had onboard a scientific tool. It was an Alpha Proton X-Ray Spectrometer (APXS) that was used to analyze the composition of some rocks and the Martian soil (Matijevic et al., 1997). Nevertheless, the first rovers including MM capabilities were the MERs, Spirit and Opportunity. As can be observed in Figure 1.3, the MERs had multiple scientific instruments on the end effector of their robotic arms, such as the Microscopic Imager (MI), an APXS like the one on Sojourner, or the Rock Abrasion Tool (RAT), able to clean rocks or the Martian surface to expose the materials of the underlying layers. Thanks to these tools, the MERs obtained evidence that the Martian climate was hotter in the past, with a dense atmosphere and an active water cycle, characteristics that made it possible for microbial life to exist.

The scientific achievements of the MERs increased the interest of the community in bringing more science to the next Mars program, MSL. As a result, Curiosity was designed with cutting-edge MM capabilities (Anderson et al., 2012). This rover is equipped with a five Degrees of Freedom (DoF) manipulator, called Sample Acquisition, Processing, and Handling (SA/SPaH). This subsystem can obtain samples from the interior of rocks or the soil with an instrument called Collection and Handling for In situ Martian Rock Analysis (CHIMRA), and process them with several

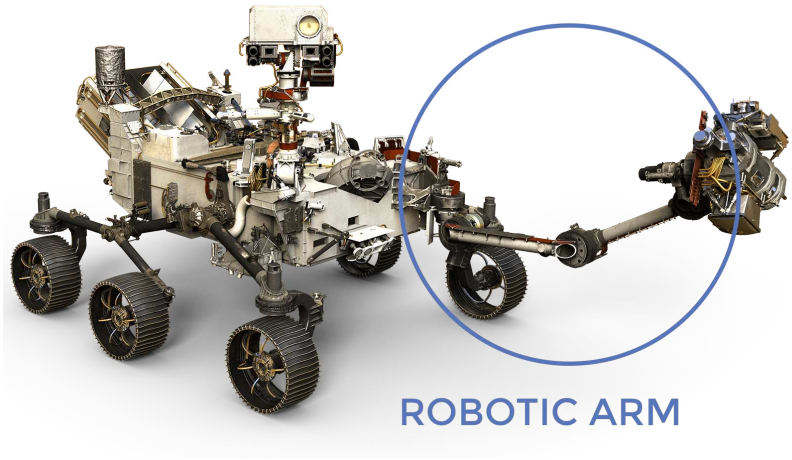


FIGURE 1.4: Perseverance Mars 2020 rover with its robotic arm (NASA/JPL-Caltech).

onboard science instruments. Additionally, the Dust Removal Tool (DRT) cleans the surfaces to ease the posterior scientific analysis with several contact instruments: an APXS and the Mars Hand Lens Imager (MAHLI), which obtains microscopic images to analyze the minerals and textures on the surface of Martian rocks and soil. The scientific discoveries of Curiosity are overwhelming thanks to these instruments. Curiosity found carbon organic molecules in some soil samples, detected methane in the Martian atmosphere, and confirmed the presence of permanent water in the past as rivers and lakes.

Finally, MSL made a way to MSR, with Perseverance. This rover is similar to Curiosity, including the so-called Sampling and Caching Subsystem (SCS) (Moeller et al., 2021), which is shown in Figure 1.4. The SCS is a five DoF robotic arm, able to collect, seal, and cache Mars soil, rock, and regolith samples. This whole procedure has hard control requirements, to avoid contamination of the samples since they will be later returned to Earth. Like the Curiosity rover, Perseverance's SCS also includes a rock abrasion and dust removal tool, which prepares the surfaces for further analysis with other scientific instruments. To date, Perseverance has discovered the presence of volcanic rocks in the Jezero Crater, where it landed. This was an unexpected discovery since it was believed that the origin of Jezero Crater rocks was purely sedimentary, from an ancient lake. Besides, Perseverance has already gathered 23 soil and rock samples, depositing ten of them on the Martian surface for a future mission to gather the sample tubes and bring them back to Earth.

Although now discontinued, the initial plan to solve the sample retrieval problem of MSR was the Sample Retrieval Lander (SRL). It was the second part of MSR, where ESA would provide the Sample Fetch Rover (SFR), a lightweight and fast MM rover designed to collect the sample tubes and bring them back to the landing site, to eventually launch them to Earth (Merlo, Larranaga, and Falkner, 2013). Now, NASA is planning to carry out the sample fetching task with MM helicopters or with the Perseverance rover itself. Anyhow, ESA has also pushed technology

regarding planetary exploration MM with the Rosalind Franklin ExoMars rover. ExoMars will obtain samples from the Martian subsurface since it is equipped with a 2 DoF drill which can reach a depth of 2 m, in order to obtain well-preserved samples of the bedrock. These samples are free from the damages of radiation and oxidation present on the Martian surface, which, in case of existence, favors the survival of chemical biosignatures.

Mars has been, undoubtedly, the lead actor in planetary exploration. However, space agencies from all over the world are lately rushing again to explore the Moon, since it will probably be used as a long-term station to sustain and ease human exploration of space. For instance, the Chinese National Space Agency (CNSA) has launched two Lunar rovers, Yutu (Xiao, 2014) and Yutu 2 (Lai et al., 2020), that landed successfully on the Lunar surface in 2013 and 2018 respectively. Equipped with robotic arms, spectrometers, ground-penetrating radars, or even an Advanced Small Analyzer for Neutrals (ASAN), the Chinese rovers were able to find evidence of at least nine different rock layers on the Moon, which implies a more complex geological composition of the Lunar surface than expected. Besides, these rovers provided high-resolution images of a Lunar ejecta. Also, NASA is planning to launch the Volatiles Investigating Polar Exploration Rover (VIPER) to the Moon in 2024 (Colaprete et al., 2019), in order to estimate the amount of water ice present on the Lunar poles, which could be used by future humans working on an eventual Lunar interplanetary station. But many other space agencies, private companies, or even universities, are planning and carrying out their own missions to the Moon involving planetary rovers. To name a few, let us list the Pragyan Lunar rover from the Indian Space Research Organisation that landed in August 2023 (Goswami, 2020), the Yaoki rover, which is supposed to realize the world's first commercial Lunar exploration in 2023 (*YAOKI - Japanese Lunar Rover* 2023), or the sad landing failure of the Rashid Lunar rover of the United Arab Emirates in April 2023 (Almaeeni et al., 2021).

As Perseverance has demonstrated, autonomy enlarges the scope of the missions and boosts the scientific return. It is foreseeable, therefore, that autonomous MM will ease the scientific exploration of extraterrestrial surfaces, while also opening a new great variety of tasks for planetary rovers. For instance, it will enable autonomous operation and maintenance of landers in the long term, using the MM capabilities to open/close valves or connect/disconnect plugs (Klamt et al., 2020). It will also permit the autonomous pick-up of materials or parts to transport and assemble them in situ (Schuster et al., 2016). Getting even further, a team of autonomous MM rovers could collaborate for in situ resource utilization to construct infrastructures on extraterrestrial surfaces (Govindaraj et al., 2019). This concept of multi-robot teams exploring collaboratively is, in fact, one of the main topics on planetary research projects nowadays, usually including a bigger rover with autonomous MM capabilities able to collect geological samples, deploy scientific instruments like a radio antenna (Schuster et al., 2020), provide power through portable batteries, or even

carrying the smaller rovers (Brinkmann et al., 2018). Highlight the use of a heterogeneous team of legged robots for space exploration, including legged mobile manipulators, as proposed during ESA's Space Resources Challenge by the team of the Swiss Federal Institute of Technology (ETH) Zurich (Arm et al., 2023).

It is clear, thus, that huge technological and scientific advances have been achieved thanks to robotics and MM in space, and this tendency is going to keep growing as space agencies and private companies from all over the world start contributing with their own rovers to explore the great variety of extraterrestrial surfaces in our Solar System. Nevertheless, autonomy in space is certainly a challenge that still requires a big research effort, even more, if we consider MM as the main approach for rovers to perform scientific and engineering tasks. Developing an advanced technology such as autonomous MM is still a challenge on Earth, and, therefore, even harder on Mars or the Moon.

Let us quickly depict how an autonomous robot works using a general robotics concept: the Sense, Think, Act paradigm (Siegel, 2003). Sense encompasses the subsystems that perceive the surroundings of the robot or measure its internal state. Think is where the trendy artificial intelligence would be placed, using the sensed information to decide what to do next and generate a plan. Lastly, Act takes care of the actuators that interact with the real world, commanding them in accordance with the plan. These three components are intrinsically required for a robot to be autonomous. Now, focusing on autonomous space robots, a Guidance, Navigation and Control (GNC) architecture is the most usual approach (Gerdes et al., 2020). In this case, Guidance (Think) is the subsystem that plans the motion of the rover to reach the goal and perform the task, Navigation (Sense) localizes the rover and maps the scenario with the onboard exteroceptive sensors, and Control (Act) dictates the low-level actuators to properly follow the commands, considering its current state and filtering perturbations.

The complexity of a GNC system is already high for an autonomous mobile robot, even worse for an autonomous MM rover. First, the rover has to reach the area of scientific interest, with a GNC autonomous architecture for the mobile base. Then, the rover has to get to the particular scientific target and carry out the task by performing an autonomous MM motion. Finally, the rover has to move to the next scientific target, commonly undoing its own steps, i.e. navigating through the same scenarios again. Such an autonomous MM process hinders particularly the Guidance since the motion planner has to consider a great number of DoF, which means there exist almost infinite ways of solving the problem. Besides, for extraterrestrial use cases, the planner has to be computationally lightweight to run on space-qualified processors, robust and predictable to ensure no fatal events could happen, and efficient enough to improve what teleoperation can do. It is hereby clear why planetary rovers do not perform MM tasks autonomously yet.

1.2 Contributions

A completely autonomous MM rover requires as a first step an efficient autonomous navigation system for the mobile base, to get to the area where to perform the MM task. Afterward, it requires a MM motion planner for both the mobile base and the manipulator, which is to date not viable in space systems, mainly due to its considerable computational cost. As a result, this thesis tries to push the boundaries of autonomous space exploration by answering how to efficiently plan the motion of a planetary rover to perform MM tasks, from the global approach of the rover to the area of scientific interest until the local execution of the MM task. This is an ambitious question, whose answer requires solving first a series of minor challenges:

- When approaching the area of scientific interest, and since the rover will probably cover the same areas again in the future, how to increase the rover's awareness of the characteristics of the surrounding scenario to improve future plans?
- When performing the scientific task, how to consider collision avoidance robustly and efficiently in the motion planning stage of MM, to ensure the system's safety during the operation?
- Given the efficiency requirements of planetary exploration vehicles, can optimal-control algorithms be applied to planetary MM motion planning, to minimize the energy spent with admissible computational costs?
- Considering the latest tendency of multi-robot space exploration, can optimal MM motion planning and control methods increase the safety of multi-robot missions in space?

These four questions are explored throughout this doctoral dissertation with a series of contributions to the space robotics field, which are the following:

1) **A dynamic cost map update procedure to improve autonomous navigation in round-trip missions**.

   Autonomous navigation for planetary rovers is commonly divided into two layers: the global layer and the local layer. Usually independent, the global layer is used to plan long- and mid-term motions by means of orbital imagery, and the local layer is used to replan in case of newly found obstacles with the rover's onboard sensors. However, if the rover is required to undo its own steps covering once again the same areas, it will find the same local obstacles as in the first traverse, unless the global layer is updated with the local information.

   Hence, the first contribution of this doctoral thesis is a dynamic update procedure for the global cost map, which increases the efficiency of the rover traverses after moving on similar scenarios, by feeding back the local information

gathered by the system during the mission. Two main sources of information are used: first, the position of newly found obstacles, that were not initially considered in the global map of the scenario; second, the characteristics of the different types of terrains in the scenario, that are continuously recomputed to improve the initial estimation. More details in Chapter 3.

2) **A robust coupled motion planner for workspace restricted MM in planetary exploration**.

Planning a safe motion of a mobile manipulator is hindered by the presence of obstacles in the scenario, but also by the onboard equipment of the rover. This equipment is constituted by scientific and engineering tools that limit the available workspace of the manipulator. Although critical, avoiding self-collisions online requires a significant computational effort, which has been classically handled by simplifying the solution with decoupled MM motions (first, moving the mobile base, second, moving the manipulator). Performing a coupled motion is more efficient in energy and time terms, but more expensive computationally too. Coupled motions would require, hence, a less demanding method for collision avoidance, such as computing offline the manipulator workspace to later consider self-collisions.

Therefore, the second contribution of this doctoral research is a coupled motion planner for workspace-restricted mobile manipulators, which ensures mission safety by considering the safe workspace of the manipulator before planning the motion. Besides, the efficiency is enhanced since the planner generates coordinated movements for the rover base and the arm, avoiding stale times by moving both systems at the same time. More details in Chapter 4.

3) **A computationally lightweight optimal motion planner for over-actuated mobile platforms applied to MM in planetary exploration**.

Over-actuated systems like mobile manipulators, i.e. platforms that have more DoF than required to move in their Cartesian space, have a considerable movement complexity. Although they are more versatile, the great variety of possibilities implies big computational efforts when planning their motion. Optimization is one of the most common methods to solve the over-actuated motion planning problem, nevertheless, their computational efficiency is severely reduced after considering dynamics and constraints. This issue can be partially mitigated with a warm start: initializing the optimization with a draft, quickly computed solution, which indicates the solver where to start looking for the optimal one.

A multi-staged warm start motion planner is presented, then, as the third contribution of this doctoral thesis. This optimal motion planner finds efficient

motions in energy and time terms, meanwhile also considering the kinematics and dynamics of the over-actuated robot, the system constraints, and nonlinearities. Besides, the warm start sequence reduces the convergence time, making the planner suitable for planetary exploration. More details in Chapter 5.

4) **Model Predictive Control (MPC) for a mobile manipulator to support a rappelling rover into Lunar lava tubes**.

Lunar lava tubes are of great scientific interest to study the origins of our solar system, and also to be used as a shelter for future space missions to increase the human reach into space. Since it is nowadays costly and dangerous to explore them with astronauts, a team of robots would be a safer and cheaper alternative. In this regard, a bigger robot could act as a mobile anchor, using a robotic arm to support the rappelling of a second smaller one, which would descend through a skylight into the cave to explore it. Clearly, this mission is significantly challenging for the systems, which may endure notable efforts during the rappelling operation. To increase the robustness of the mission, it would be key to control those efforts, particularly the ones affecting the robotic arm. Such a reactive, high-frequency control can be achieved using modern optimal control techniques, such as MPC.

Making the most of the third contribution of this doctoral thesis (the optimal motion planner), an optimal controller was developed and integrated into a tension compensation algorithm. This algorithm is able to cope with the efforts of the robotic arm during the rappelling support by performing MM motions. This tension controller is, thus, the fourth contribution of this doctoral dissertation. More details in Chapter 6.

1.3 Context and motivation

Ongoing and future planetary exploration missions are demanding improved self-reliant navigation capabilities to keep increasing their scientific return, as aforesaid. As a result, ESA and the European Commission (EC) have lately funded projects related to the boost of the autonomy level for planetary rovers. This includes research on path and motion planning, MM scenarios, or collaboration between rovers. The Space Robotics Laboratory from the University of Málaga (UMA-SRL), Department of Systems Engineering and Automation (ISA), has been actively participating in several of these projects, thanks to its wide experience in the field of autonomous navigation for mobile robots. This is the context where the research for this doctoral thesis was conducted. In particular, the author of this thesis worked intensively on three of these projects, which impacted significantly the results of his research: first, a collaboration between ESA and UMA-SRL with three consecutive research activities; second, the EC-funded H2020 project called Autonomous Decision making for very

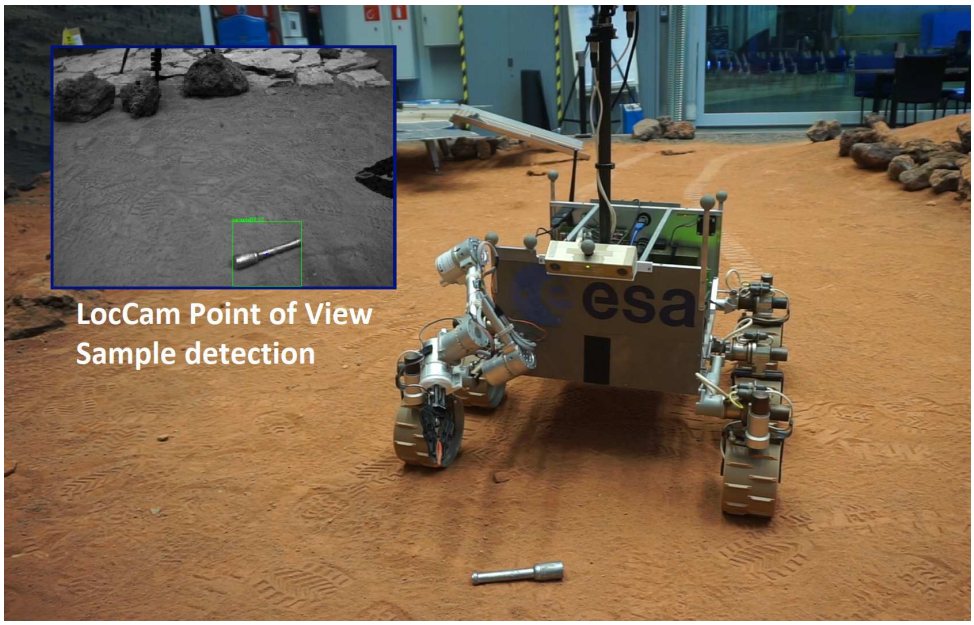


FIGURE 1.5: ExoTeR detecting a sample tube at the final tests of the ESA-UMA activity called PM-SFR.

long traverses (ADE); third, the last EC H2020 project on space robotics called Cooperative Robots for Extreme Environments (CoRob-X). Let us present these projects in the following.

UMA-SRL has been collaborating since 2016 with the Planetary Robotics Laboratory, Robotics and Automation Section, European Space Agency (ESA-PRL), located at the European Space Research and Technology Center (ESTEC), through three consecutive research and development activities. The first one, contract no. 4000118072/16/NL/LvH/gp, was called Autonomous Routing on Extreme Surfaces (ARES) and the objective was to develop an efficient path planner for reconfigurable planetary rovers, considering different locomotion modes, which are favorable to traverse different types of soil (Pérez-Del-Pulgar et al., 2017). The final tests carried out at the ESA-PRL with the Exomars Testing Rover (ExoTeR) were very promising, and further research led to the publication of several high-impact international journal papers. First, the Fast Marching Method (FMM), a method for path planning that generates optimal, smooth trajectories, was extended for multi-layered grids and dynamic replanning (Sánchez-Ibáñez et al., 2019b). Second, an efficient autonomous navigation architecture for GNC was proposed in (Gerdes et al., 2020). Finally, the use of Simultaneous Localization and Mapping (SLAM) and map matching for relative and absolute localization was presented in (Geromichalos et al., 2020). Exploiting the developed path planner in the framework of ARES, the second activity aimed at developing a complete motion planner for a rover equipped with a manipulator, which has to perform retrieving tasks (Sánchez-Ibáñez et al., 2019a). Called Path and Motion planning in Planetary Exploration (PM-PE), the resulting tests of this activity allowed ExoTeR to retrieve samples with an efficient and robust motion plan. Nevertheless, a sweeping movement with the end effector was needed to consider

FIGURE 1.6: SherpaTT rover of DFKI during the H2020 ADE project.

localization errors in the rover and sample poses. To perform a complete SFR-like mission, the rover should be able to detect and identify the sample before retrieving it, which was the main drawback of the PM-PE project. Thus, the third ESA-UMA activity, called Path and Motion planning for a Sample Fetching Rover (PM-SFR), aimed at integrating an intelligent vision system into the autonomous navigation architecture of the rover (Castilla-Arquillo et al., 2022), in order to detect the samples tubes online and perform a fully autonomous sample fetching mission (Mantoani et al., 2022). The results of this third activity were outstanding, with ExoTeR being capable of reaching, detecting, and retrieving a sample tube in a completely autonomous manner. This can be observed in Figure 1.5, which shows the moment of the sample identification in one of the laboratory tests at the ESA-PRL.

It was in the context of these collaboration activities between ESA and UMA-SRL that this thesis author had the privilege to stay twice at ESTEC, in Noordwijk, The Netherlands, under the supervision of Dr. Martin Azkarate, with a total duration of six months. During these stays, the author performed research and development, as well as laboratory and field test campaigns, that supported the thesis contributions. In particular, the algorithms and field trials of Contribution 1 were thoroughly accomplished at ESTEC during the first research stay in late 2019; the algorithm proposed in Contribution 2 is based on the motion planner developed for the second activity, PM-PE, and the laboratory tests for Contribution 3 were completely conducted at the ESA-PRL in ESTEC during the second research stay.

The acquired experience and developed algorithms within the ESA-UMA collaborations benefited the UMA-SRL to participate in the project *Autonomous DEcision making in very long traverses (ADE)* 2021. This was an EC-funded project of the H2020

call, belonging to the Strategic Research Cluster (SRC) on Space Robotics Technologies. The objective of ADE was to integrate most of the developed technologies in previous H2020 projects to create a fully autonomous navigation architecture, allowing a rover to navigate long distances on its own, between 1 to 3 km. Coordinated by GMV Aerospace and Defence S.A., the consortium was made up of 14 partners, including the Robotics Innovation Center (RIC) of the German Research Center for Artificial Intelligence (DFKI). DFKI was in charge of the rover SherpaTT, which can be observed in Figure 1.6, and the simulator.

In ADE, UMA-SRL was responsible for the MM component, which generates a motion plan for the rover-manipulator system in order to reach a certain goal and perform a scientific task with the robotic arm (Sánchez-Ibáñez et al., 2020). In this case, it was particularly challenging to generate safe motion plans due to the highly restricted workspace of the robotic arm, in view of the presence of several instruments for localization, hazard detection, or Digital Elevation Map (DEM) generation. Therefore, in the context of ADE, this thesis author developed a robust motion planner algorithm, prioritizing the security of the manipulator with self-collision avoidance methods, which is the second contribution. The simulations of the MM component for ADE, and the field trials that took place in 2021 at Bremen, Germany, originated the results that support Contribution 2.

In third place, the UMA-SRL participated in *CoRob-X - Cooperative Robots for Extreme Environments* 2023, which started in March 2021, being the most challenging H2020-funded project. It was the last one of the H2020 SRC projects on Space Robotics Technologies, and hence the developed technologies of all the previous H2020 projects were integrated into several Robotic Explorer Units (REU)s, which had to collaborate on exploring planetary surfaces, specifically focusing on Lunar caves and lava tubes. CoRob-X was led by DFKI, and UMA-SRL was in charge of the autonomous navigation of three of the REUs: the DFKI rovers SherpaTT and CoyoteIII, and the Space Application Services (SAS) rover LUVMI-X. Besides, a very risky MM task was performed: the SherpaTT robotic arm supporting the CoyoteIII rover with a tether meanwhile the latter is rappelling down a lava tube skylight.

The rappelling required a highly efficient and fast motion planner and controller, in order to place the robotic arm in the optimal poses to let CoyoteIII descend safely into the cave. This meant maintaining the torques applied to the SherpaTT robotic arm to the minimum, distributing them to the strongest joints, and ensuring that the arm was able to move smoothly in any direction while CoyoteIII was descending. The development of such challenging algorithms motivated the fourth contribution, as well as the stay of this thesis author at DFKI RIC in Bremen, Germany, for the whole year of 2022. This research stay also led to a cotutelle agreement for the doctoral studies, finalizing as a joint supervision with Dr. Carlos Jesús Pérez del Pulgar from UMA-SRL and Dr. Frank Kirchner from Universität Bremen. Finally, the 2023 CoRob-X field trials of Lanzarote, Canary Islands, Spain, resulted in multiple successful (and impressive) rappelling operations, as depicted in Figure 1.7, where both

FIGURE 1.7: SherpaTT and CoyoteIII before an attempt to rappel down the lava tube skylight within the field trials of the H2020 CoRob-X project in Lanzarote, Canary Islands, Spain.

rovers SherpaTT and CoyoteIII are shown instants before attempting to rappel down a lava tube skylight.

1.4 Publications

The contributions stated in this doctoral dissertation were published in relevant conferences and journals related to the space robotics field. In particular, there are four main publications, which are the following:

- **Improving Autonomous Rover Guidance in Round-Trip Missions Using a Dynamic Cost Map**. Gonzalo J. Paz-Delgado, Martin Azkarate, J. Ricardo

Sánchez-Ibáñez, Carlos J. Pérez-del-Pulgar, Levin Gerdes, and Alfonso García-Cerezo. *International Conference on Intelligent Robots and Systems (IROS)* Las Vegas, United States of America (USA), 25-29 10 2020 (Paz-Delgado et al., 2020).

Contribution 1 of this doctoral research is summarized in this conference paper, related to a procedure to improve the planned motions dynamically in planetary exploration. This is done by updating the path planner cost map to make the rover conscious of the gathered information of the scenario during the traverse and using it to generate better paths successively during the mission. The paper also shows a simulation and field test campaign carried out with the ESA-PRL's Heavy Duty Planetary Rover (HDPR) at Decos, close to ESTEC in Noordwijk, The Netherlands.

- **Combined path and motion planning for workspace restricted mobile manipulators in planetary exploration**. Paz-Delgado, J. Ricardo Sánchez-Ibáñez, Raúl Domínguez, Carlos J. Pérez-del-Pulgar, Frank Kirchner, and Alfonso García-Cerezo. *IEEE Access* (Paz-Delgado et al., 2023b).

  This journal paper is directly related to the ADE project and Contribution 2. It presents a coupled path and motion planner considering a highly restricted workspace of the robotic arm in planetary exploration MM. To ensure mission safety, the workspace of the manipulator is computed offline and used during the motion planning stage to avoid self-collisions. Besides, the efficiency of the motion plan is enhanced by coordinating the rover and manipulator movements. The paper concludes with a simulation and field test campaign with the DFKI RIC rover SherpaTT, performed in the context of the ADE project.

- **Multi-stage warm started optimal motion planning for over-actuated mobile platforms**. Gonzalo J. Paz-Delgado, Carlos J. Pérez-del-Pulgar, Martin Azkarate, Frank Kirchner, and Alfonso García-Cerezo. *Springer Journal on Intelligent Service Robotics (JIST)* (Paz-Delgado et al., 2023a).

  The research and results of Contribution 3, i.e. an optimal motion planning algorithm for over-actuated mobile platforms, are presented in this journal paper. In particular, it proposes a general approach to model over-actuated mobile systems, like mobile manipulators, and a novel warm-starting pipeline for optimal control-based motion planners. Finally, it includes a laboratory test campaign with ExoTeR at the ESA-PRL in ESA-ESTEC, that validates the presented algorithms. This paper was the journal cover of *Springer JIST*, Volume 16, Number 3, July 2023[1], as can be observed in Figure 1.8.

- **Cooperative robotic exploration of a planetary skylight surface and lava tube**. Raúl Domínguez, Carlos J. Pérez-del-Pulgar, Iulia Dragomir, Valérie Ciarletti, Gonzalo J. Paz-Delgado, Anne-Claire Berthet, Fabio Polisano, Leon Cedric Danter, Jonathan Babel, and Frank Kirchner. In preparation.

[1] `https://link.springer.com/journal/11370/volumes-and-issues/16-3`

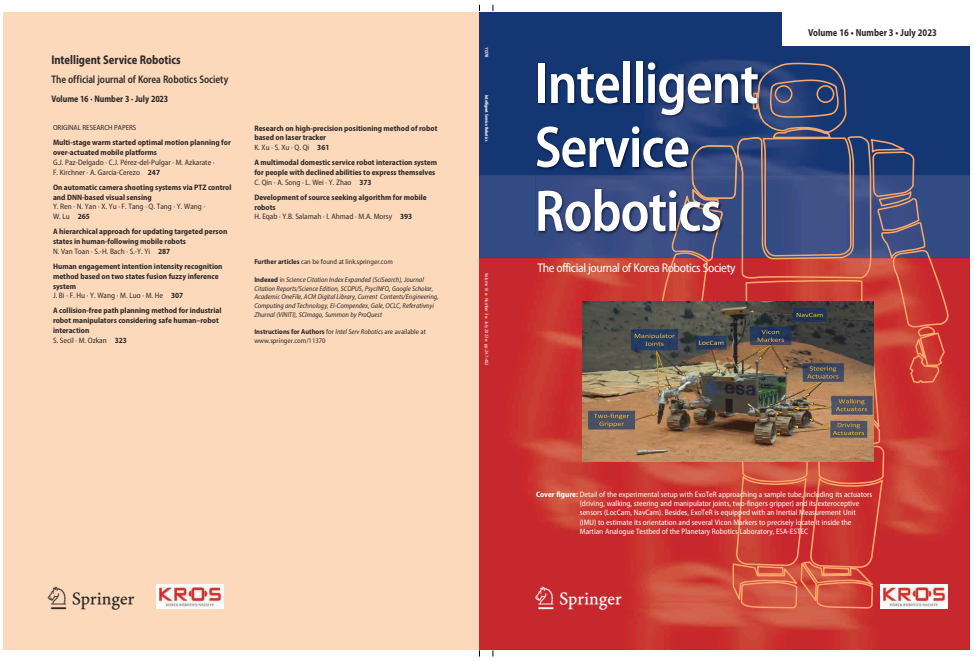


FIGURE 1.8: The paper "Multi-staged warm started optimal motion planning for over actuated mobile platforms", which presents Contribution 3, as the journal cover of *Springer JIST*, Volume 16, Number 3, July 2023.

The experiments conducted during the CoRob-X Lanzarote field trials of 2023 are reported in this journal paper, including the multi-robot rappelling into Lunar skylights, which are directly related to Contribution 4. It details the developed autonomous rappelling architecture, including the MPC for rappelling support through MM, the multi-robot team, and the rappelling experiments.

Additionally, during his doctoral studies, the author collaborated in multiple research activities and projects related to the space exploration topic, which led to several more publications. The main ones are listed in the following:

- **Hardware-Accelerated Mars Sample Localization Via Deep Transfer Learning from Photorealistic Simulations**. Raúl Castilla-Arquillo, Carlos J. Pérez-del-Pulgar, Gonzalo J. Paz-Delgado, and Levin Gerdes. *IEEE Robotics and Automation Letters*, pp 99:1-8, 2022 (Castilla-Arquillo et al., 2022).

  This journal paper proposes an Artificial Intelligence (AI) approach for autonomous detection and pose estimation of MSR sample tubes, using a Deep Neural Network and transfer learning. The used dataset is obtained from a photorealistic 3D simulator of Mars using Unreal Engine 4. This sample detector and pose estimator was integrated within the GNC architecture of ExoTeR, and used during the laboratory tests of Contribution 3.

- **Choosing the Best Locomotion Mode in Reconfigurable Rovers**. Carlos J. Pérez-del-Pulgar, Pablo Romeo-Manrique, Gonzalo J. Paz-Delgado, J. Ricardo

Sánchez-Ibáñez, and Martin Azkarate. *Electronics*, 8(7):818, 7 2019 (Pérez-Del-Pulgar et al., 2019).

Rovers like ExoTeR present different locomotion modes thanks to the reconfigurability of their legs and wheels. In this paper, a deep analysis of two locomotion modes is carried out in simulations with Vortex Studio. These locomotion modes are Normal Driving, the standard rolling locomotion mode, and Wheel-walking, where the legs are used to perform small sequential steps. In function of the characteristics of the soil, the best locomotion mode can be chosen based on the results of this paper. Additionally, a novel slip estimation method when using Wheel-walking is presented.

- **Experimental analysis of slip ratio using the wheel walking locomotion mode in reconfigurable rovers**. Salvador Domínguez-Durante, Carlos J. Pérez-del-Pulgar, Gonzalo J. Paz Delgado, and Martin Azkarate. *30th Mediterranean Conference on Control and Automation (MED)*, 2022 (Dominguez-Durante et al., 2022).

  Continuing with previous research about locomotion modes in reconfigurable rovers, this conference paper presents the experimental validation of the aforementioned slip ratio estimation method when using the Wheel-walking locomotion mode. To this end, an experimental setup to induce a given slip to the rover is proposed, which consists of a flat terrain and a tether mechanism.

- **Coupled Path and Motion Planning for a Rover-Manipulator System**. J. Ricardo Sánchez-Ibáñez, Gonzalo J. Paz-Delgado, Pablo Romeo-Manrique, Carlos J. Pérez-del-Pulgar, and Martin Azkarate. *15th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2019 (Sánchez-Ibáñez et al., 2019a).

  In this conference paper, the authors present the starting point of the later motion planner of Contribution 2 in ADE. This original motion planner generates a tunnel surrounding the mobile base path, where FMM in a 3D version is used to draw the end effector path. Afterward, the Inverse Kinematics (IK) model of the manipulator is used to obtain the manipulator configurations. The motion planner is finally tested in simulations with ExoTeR in Vortex Studio.

- **Samples detection and retrieval for a Sample Fetching Rover**. Laura M. Mantoani, Raúl Castilla-Arquillo, Gonzalo J. Paz-Delgado, Carlos J. Pérez-del-Pulgar, and Martin Azkarate. *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2022 (Mantoani et al., 2022).

  The complete autonomous sample retrieval architecture integrated into ExoTeR is presented in this conference paper, including the Guidance algorithms (path planning and trajectory following) and the sample detection and pose estimator. Two experiments are presented, a field test in the Search and Rescue experimental terrain at UMA, and a lab test in the ESA-PRL at ESA-ESTEC.

This autonomous navigation architecture is later benchmarked with the proposed algorithms of Contribution 3.

- **CoRob-X: A cooperative robot team for the exploration of Lunar skylights**. Alexander Dettmann, Thomas Vögele, Jorge Ocón, Iulia Dragomir, Shashank Govindaraj, Matteo De Benedetti, Valerie Ciarletti, Rafik Hassen-Khodja, Thierry Germa, Raphael Viards, Gonzalo J. Paz-Delgado, and Laura M. Mantoani. *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2022 (Dettmann et al., 2022).

  This conference paper gives a general overview of the CoRob-X project, which is related to Contribution 4. The paper presents the overall approach used within the project, including the control and software architecture or the robotic systems, called REUs. Additionally, this work depicts the mission phases for the lava tube exploration, which were later demonstrated in the field trials of Lanzarote.

- **Virtual reality lab for rover navigation using Mars datasets**. Raúl Castilla-Arquillo, Gonzalo J. Paz-Delgado, Matteo Madi, and Carlos J. Pérez-del-Pulgar. *17th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2023

  A high-fidelity Virtual Reality Lab to simulate planetary rovers is presented in this conference paper. Developed by UMA-SRL, the Virtual Reality Lab uses real data (Martian orbital imagery and rover's local images) to recreate a Martian environment with the help of AI. On the one hand, super-resolution methods are applied to orbital imagery to increase the quality of the textures of the scenario. On the other hand, Generative Adversarial Networks are used to create 3D terrain features such as rocks from the rover local images.

- **CoRob-X: Demonstration of a cooperative robot team in extensive field tests**. Thomas Vögele, Jorge Ocón, Thierry Germa, Shashank Govindaraj, Carlos J. Pérez-del-Pulgar, Fredrik Bakkevig Haugli, Raúl Domínguez, Leon Cedir Danter, Jonathan Babel, Iulia Dragomir, Anne-Claire Berthet, Fabio Polisano, Gonzalo J. Paz-Delgado, Laura M. Mantoani, Eric Törn, and Valerie Ciarletti. *17th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2023

  This conference paper shows the objectives and results of the CoRob-X project field trials in Lanzarote, Canary Islands, Spain. The paper summarizes the results of the Lunar lava tube exploration by analyzing the different mission phases, including the rappelling operation where Contribution 4 was tested.

1.5 Thesis outline

This doctoral dissertation is divided into seven main chapters, first, introducing the topic and analyzing the related state of the art, second, describing the four main contributions, third, presenting the performed experiments and results, fourth, stating the conclusions of the dissertation and some future work. Let us detail each chapter in the following:

1) **Introduction**. The first chapter answers where this doctoral dissertation fits in, why is it necessary to advance the technology in this field, and what problems the thesis solves. Additionally, this chapter puts the reader in context w.r.t. the circumstances where the author's work originated, and summarizes the contributions and publications related to this doctoral research.

2) **State of the art in autonomous mobile manipulation**. This chapter is an extensive analysis of the to-date situation of technology related to algorithms and architectures required to autonomously perform MM tasks in extraterrestrial surfaces, i.e. perception, path planning, motion planning, and motion control.

3) **Dynamic cost map update to enhance global path planning**. The theory behind Contribution 1 is presented in this chapter, describing the functioning of the proposed approach to dynamically update the cost map of the path planner once the rover is traversing the scenario. First, the used path planner is introduced to state its peculiarities; second, the procedures to update the costs are explained, both for the newly found obstacles and the terrain characteristics estimation.

4) **Robust motion planning for coupled mobile manipulation**. Contribution 2 is thoroughly explained in this chapter, i.e. the robust and efficient motion planner for MM in planetary exploration, focused on mobile robots with a highly restricted arm workspace. It is a combined path and motion planner, therefore, the improvements of the path planner for considering later MM tasks are first detailed, then describing the motion planner itself and the coupled controller required to coordinate the movements of the mobile platform and the robotic arm.

5) **Optimal motion planning for over-actuated mobile platforms**. The fourth chapter presents Contribution 3, the optimal but lightweight motion planner for over-actuated mobile platforms. To this purpose, first, it presents the generic approach to model mobile platforms composed of several kinematic chains. Afterward, the multi-staged warm-started optimal motion planner is depicted, with a precise description of each of the stages.

6) **Results**. This chapter encompasses all the numerical, simulation, laboratory and field tests performed during this doctoral research, the ones that validate

the proposed solutions and demonstrate their advantages and disadvantages from both quantitative and qualitative points of view. Specifically, four campaigns are included, one for each contribution: first, the round-trip experiments with the dynamic cost map update with HDPR of ESA; second, the thoroughly safe MM operations with SherpaTT in ADE; third, the optimal sample retrieval tests with ExoTeR at ESA-PRL; fourth, the coordinated rappelling to a lava tube skylight with SherpaTT and CoyoteIII in CoRob-X. This chapter also analyzes the controllers used to bring the optimal motion plans to the different platforms, i.e. a simple error-based replanning approach, and an MPC, which constitute the algorithm used for Contribution 4.

7) **Conclusions and future work**. The dissertation finishes with this chapter, which summarizes all the proposed solutions, discusses the results, and proposes some future work, i.e. other aspects related to the topic that need to be further investigated, problems that have not been completely solved, or solutions that can be yet improved.

Chapter 2

State of the art in autonomous mobile manipulation

"Ex nihilo nihil fit" -
"Nothing can be made from nothing"

Titus Lucretius Carus
De rerum natura, 1st century B.C.

2.1 Introduction

Teleoperation from Earth is the main hindrance that reduces the scientific return of Martian rovers, as introduced in the first chapter. This difficulty has been broadly overcome in the latest NASA Mars missions by gradually including autonomous navigation capabilities in planetary rovers. The benefits of an autonomous rover are clear, i.e. increased time and energy efficiency, notably increasing the number of tasks performed and the driving distance after reducing stale times caused by delays in teleoperation. However, most scientific tasks carried out by planetary rovers involve MM, i.e. moving the rover and the robotic arm together to place a scientific tool on a spot of interest in the scenario. And these MM operations are, to date, not autonomous on planetary rovers.

Autonomous MM will raise to a further grade the scientific return of Martian rovers, even enabling a broad new variety of functionalities like in-situ resource utilization, assembly, or construction. But, to bring autonomous MM to Mars, space agencies should first focus on MM applications on Earth: it is here where the technology is progressing. Autonomous MM is able to tackle repetitive tasks such as assembly, disinfection, or transportation, but also unstructured and complex tasks such as inspection, agriculture, or search and rescue. Consequently, there is a vast range of solutions available in the on-Earth applications literature (Thakar et al., 2023) that could eventually be used in space robotics too, adapting them to consider the requirements and constraints of planetary exploration missions.

Let us get into detail about the intelligence required by an autonomous robot, particularly GNC insofar as it is the most common architecture to provide autonomy

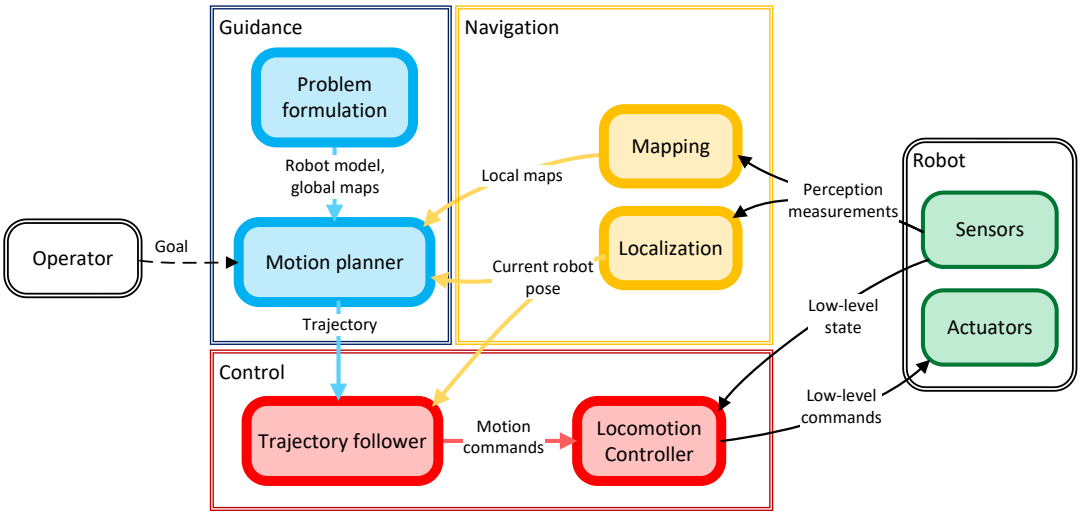


FIGURE 2.1: Simplified scheme of a GNC autonomous navigation architecture.

to space systems since the Apollo missions (Duncan and Gunnersen, 1964). Similar to the Sense, Think, Act robotics paradigm, a GNC architecture is subdivided into three main components as shown in Figure 2.1, which are the following:

- Guidance. This component aims to plan a feasible motion to reach the goal (Motion planner). For that purpose, it mainly requires the current state of the robot and a general description of the workspace (Problem formulation), with a model of the robot and a model of the scenario, usually in the form of global maps. Additionally, the motion planner commonly receives updated local maps of the scenario, to replan the motion if necessary, e.g. if a new obstacle is found.

- Navigation. Navigation uses the proprioceptive (internal) and exteroceptive (external) sensors to feed back to Guidance and Control valuable information about the robot and its surroundings. For instance, this information includes generating maps and detecting obstacles (Mapping), localizing or positioning the system (Localization), and measuring the robot's subsystems state.

- Control. As the component that actuates the system to interact with the real world, Control continuously commands the robot to follow the motion plan in function of the current state of the system (Trajectory follower), checking that the commanded motion is precisely tracked meanwhile filtering any external disturbances (Locomotion Controller), considering the real constraints and limits of the actuators.

All of the aforementioned components are also necessary to enable autonomous MM in planetary exploration. In the first place, since the rover still has to autonomously reach an area of scientific interest. In the second place, to perform the MM tasks inside that area. The main difference is in Guidance: an autonomous rover

can effectively navigate using a bi-dimensional path planner, whereas a mobile manipulator requires a more complex motion planner that accounts for the base and the manipulator. Also, the Control is more complex in MM, since the base and arm have to be commanded simultaneously to follow the planned motions. As a result, motion planning and control are the main hindrances that need to be surpassed to reach a high level of autonomy when performing MM operations.

MM motion planning and control are computationally expensive, too expensive for planetary applications, seeing as how complex it is to generate and perform motions considering the system's kinematic and dynamic constraints, the non-traversable areas in the scenario, uncertainty, collisions, or the efficiency and time requirements. The majority of motion planning problems on Earth can be defined as trajectory-constrained since the robot has to follow a predefined trajectory to perform a usually repetitive and well-defined task, e.g. assembly, disinfection, or construction. Trajectory-constrained problems for MM have been explored for a long time (Oriolo and Mongillo, 2005), achieving quite impressive results lately (Kabir et al., 2021). Nonetheless, motion planning for MM in planetary exploration is understood as a goal-constrained problem. Considering the goal as the target of scientific importance in the scenario, this definition means that there is no imposition on how or when to get to the goal as long as the system's integrity and limits are satisfied. In other words, goal-constrained problems have no particular interest in what movements the rover performs but in reaching the goal safely, avoiding obstacles or non-traversable areas in the scenario, and efficiently in time and/or energy terms. Though less restrictive than trajectory-constrained problems, goal-constrained ones have a broader amount of solutions, which generally implies a heavier computational effort.

So, in this chapter, the state of the technology of autonomous MM is exhaustively presented. First, Section 2.2 depicts in more detail the state-of-art GNC building blocks to enable autonomous navigation for a planetary rover, i.e. mapping, localization, path planning, and trajectory following. Second, adding MM to the loop, Section 2.3 analyzes different on-Earth approaches to solve the goal-constrained motion planning problem for MM, including sub-optimal and optimal motion planners. Later on, considering the great variety of ways to bring the planned motion to the MM robot while reacting to unexpected external events, in Section 2.4 a summary of reactive control and trajectory following approaches for MM on-Earth is included. Finally, Section 2.5 concludes the chapter by pointing out where the state of the art needs to be further pushed to perform autonomous MM in planetary missions.

2.2 Autonomous navigation in planetary missions

As introduced in the first chapter, autonomous navigation capabilities of planetary rovers have evolved substantially since the first Mars missions. The aforementioned

GNC architecture has been present in most of the rovers, with four key components being especially difficult to address when trying to increase the amount of self-driving time. These components are the most computationally demanding and safety-critical ones, i.e. mapping, localization, path planning, and trajectory following.

In summary, mapping is the component in charge of modeling the scenario where the rover will traverse; localization estimates where the system is placed w.r.t. a given reference frame; path planning generates a trajectory to reach a goal from the rover's position, and trajectory following continuously commands the system actuators to track the computed trajectory according to the evolving position of the rover. Let us detail the purpose, functioning, and state of technology regarding these four building blocks, throughout the history of planetary exploration to date.

Mapping and terrain assessment

Planetary exploration scenarios are characterized by being static but unstructured. Mapping and modeling a thoroughly known static scenario is straightforward, conversely, an unknown unstructured terrain presents significant uncertainty regarding the shape of the terrain, the presence of obstacles, and the soil properties. These three aspects are crucial to ensure the rover's safety and mission success. First, the shape of the terrain defines slopes, craters, cliffs, pits, or even big rocks, i.e. areas that the vehicle cannot access or go through. Second, the presence of obstacles, such as small rocks, puts in danger the system integrity when located in traversable areas. Third, the properties of the soil establish how the system will behave when traversing the terrain. Loose soil may lead to a dangerous sinking of the rover wheels, whereas rocky soil may damage them. To contemplate these aspects it is common, thus, the use of two main layers of information about the scenario, the global layer and the local layer. Let us describe these two layers in detail.

The global layer gives a general overview of the scenario characteristics and is used to generate an initial global plan for the mission. For that purpose, it is necessary to extract global 3D data from the scenario and arrange it as a heightfield, commonly called DEM in planetary applications. A DEM is a 2D grid where each node indicates the height of the terrain in that position of the scenario, which is practical to later compute slopes and detect the biggest obstacles, such as cliffs or craters as aforementioned. Building DEMs, also known as topographic mapping, has been extensively done using aerial stereo imagery (Lacroix, Jung, and Mallet, 2002) on Earth. Nevertheless, no aerial information is available on extraterrestrial surfaces, hence, global maps are generated using orbital imagery. For instance, (Gwinner et al., 2010) were able to generate DEMs of the Martian surface up to 50 m resolution, using images obtained with the High Resolution Stereo Camera (HRSC) from the Mars Express, and (Li et al., 2011) use the High Resolution Imaging Science Experiment (HiRISE) instrument to create some remarkable high-quality DEMs from Mars, up to 1 m resolution. Additionally, it is possible to merge information from orbital

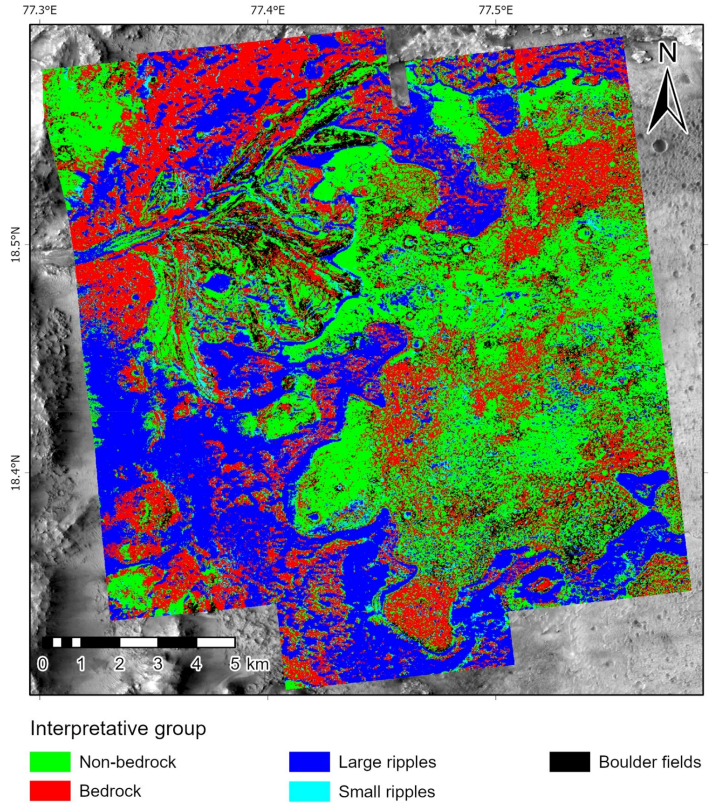


FIGURE 2.2: NOAH-H terrain classification of the Jezero crater (Wright et al., 2022).

imagery, descent images from the lander, and surface images from the rover and lander to generate higher-quality maps from Mars (Olson et al., 2007).

The orbital imagery can also be used to perform a first classification of different areas of the scenario, with a preliminary estimation of the soil properties at each one to evaluate the traversability of the terrains. (Rothrock et al., 2016) presented the Soil Property and Object Classification (SPOC), which uses a convolutional neural network to classify different areas on Martian orbital imagery. Another recent example is the Novelty or Anomaly Hunter-HiRISE (NOAH-H) (Wright et al., 2022), which uses deep learning techniques to classify the Jezero crater using HiRISE Martian imagery, as shown in Figure 2.2.

The local layer, on the other hand, permits the improvement of the global plan by using local information to update the global one, and replan online if necessary (Hedrick, Ohi, and Gu, 2020). The local information mainly comes from the rover's exteroceptive sensors, which give a more precise estimation of the soil characteristics and the presence of smaller obstacles. Focusing first on the wheel-soil interaction, also known as terramechanics, several soil parameters in off-road scenarios can mostly define a rover's behavior, such as vibration, rolling resistance, compactness, friction, and slippage, which is the most critical one. Terrains with a high slip ratio may sink the wheels of the rover, trapping it and jeopardizing the mission. This is what happened to the Spirit rover, as shown in Figure 2.3. Indeed, the Mars

FIGURE 2.3: Wheels of the NASA Spirit rover sinking in loose Martian soil due to high slippage (NASA/JPL-Caltech).

Exploration Rovers included a visual odometry algorithm, mainly used for localization, but also used to estimate the slippage online (Maimone, Cheng, and Matthies, 2007). This visual odometry has been enhanced in the Mars2020 Perseverance rover to predict future slippage, by growing uncertainty boxes and projecting potential slip ratios along the path between consecutive visual odometry updates (Verma et al., 2023). AI can also be used to model terramechanics. For instance, relying on the rover's proprioceptive sensors to provide precise slippage estimations (Gonzalez, Fiacchini, and Iagnemma, 2018) using supervised learning methods such as Support Vector Machine (SVM)s or artificial neural networks. It is worth mentioning that reconfigurable rovers, like Rosalind Franklin ExoMars, can avoid slippage by using a different locomotion mode such as wheel-walking (Azkarate et al., 2015). Besides, there are methods to estimate the slip ratio meanwhile using wheel-walking, so the system can return to normal driving when the terrain is safer (Pérez-Del-Pulgar et al., 2019).

Another critical function of the local layer is the detection and localization of local obstacles. These obstacles are the ones big enough to put the system's integrity in danger, but small enough to not be detected in the global layer. Local obstacles have been generally detected in planetary exploration by analyzing a locally generated DEM (Maimone, 2017). Getting images and generating a DEM requires some computation time, hence, the rover has to stop from time to time to assess the safety of its surroundings. This is the approach, for instance, of the Rosalind Franklin ExoMars rover (Bora et al., 2017). A solution to this is to include processors that can perform highly parallelizable computer vision operations in an efficient manner, such as the radiation-hardened FPGA onboard Perseverance (Verma et al., 2023). However, new approaches are trying to avoid the computational effort required to reconstruct the 3D data. For instance, authors in (Gerdes et al., 2020) detect local obstacles efficiently

by placing a calibration matrix in a particular Region of Interest (RoI) just in front of the rover. This allows them to estimate the distance to the pixels in the RoI, using stereo cameras, and quickly detect deviations from the expected values of the calibration matrix. When the deviations are considerable, then an obstacle is blocking the rover's path. Also, artificial intelligence is starting to gain popularity for local obstacle detection, as shown in (Liu et al., 2023), where the authors use a U-shaped Transformer framework to detect rocks on local images, separating the rock pixels from the background. Remark that modern AI super-resolution methods help to increase the quality of the rover's local images (Wang et al., 2021), which can assist in detecting potential hazards earlier, from further distances.

Local mapping is also crucial for the selection of scientific goals in the scenario. The scientific return of the mission can be remarkably raised if this high-level task is performed autonomously, as demonstrated by the AEGIS capability of the Curiosity rover (Francis et al., 2017). AEGIS is able to autonomously choose scientifically interesting targets in the local images of the rover, significantly raising the rate of scientific observations due to removing stale times from the operators to select the targets and command the rover. Particularly, the Chemistry and Camera complex (ChemCam), a remote geochemical spectrometer onboard Curiosity, increased by 25 % its pace of data collection thanks to AEGIS.

Conclusively, mapping implies the combination of the global and local layers to make up a complete model of the scenario. The global layer provides a general overview, which is useful for generating a high-level plan, and the local layer gives the system a more accurate description of the scenario, crucial to creating precise short-term plans to move and do science in the surroundings of the rover.

Localization

To autonomously reach a target, undoubtedly the rover has to precisely situate itself w.r.t. the mapped scenario, which is called localization. A bad localization may imply that the system does not reach the goal but some other erroneous position, or that the rover gets too close to hazardous areas (although thinking it is further from them). On-Earth off-road mobile robots can generally rely on global satellite localization systems such as Global Navigation Satellite System (GNSS) or Galileo, as we humans do on a daily basis to navigate our cities. Global Positioning System (GPS), for instance, can provide localization accuracy in the order of cm, far sufficient for most rover-sized mobile platforms. Nevertheless, there is nothing like GPS on Mars. Though global positioning can be performed on Mars by using orbital information from Martian satellites like HiRISE (Tao, Muller, and Poole, 2016), this can only be done from time to time when the orbiter is in view of the rover. Another solution is to use imagery from the lander to provide a global estimate of the rover localization, as was done with Sojourner (Volpe et al., 2000). Nevertheless, this approach restricts the rover's mobility to the surroundings of the lander.

As a result, planetary rovers have to rely on relative localization methods using their onboard sensors. The most basic relative localization approach is wheel odometry, i.e. estimating the robot's relative displacements in function of the readings of the wheel joints' encoders. Wheel odometry is very poor in estimating changes in the system's attitude, hence, it is commonly fused with orientation information retrieved from other onboard sensors such as an Intertial Measurement Unit (IMU). Known as inertial odometry, this was the initial setup of the MERs (Ali et al., 2005), which also included the so-called Sun finding function to determine the rover's heading by means of the direction to the sun. Though better than basic wheel odometry, inertial odometry cannot consider the soil properties to estimate the pose, resulting in considerable errors (in the order of 10 % of the covered distance) when the wheels experience slip.

Consequently, more advanced localization methods rely on visual information to improve the accuracy and cope with the soil properties, like visual odometry. This method consists of continuously obtaining stereo images of the system's surroundings and matching features between them. The features' position difference within the images provides an estimation of the relative displacement of the rover between the acquisition of those images. Hence, visual odometry also considers if the rover is slipping, resulting in a very robust localization as demonstrated by the MERs (Maimone, Cheng, and Matthies, 2007) and Curiosity (Grotzinger et al., 2012), although its computational cost restricted its use to critical cases, i.e. when traversing slippery terrain.

It is on Perseverance where the localization has significantly improved, thanks to using the FPGA for the visual odometry features matching (Verma et al., 2023). Perseverance has used visual odometry in 98 % of all its driving to date, with a 99.82 % success rate. Anyhow, since some positioning uncertainty still remains (in the order of 1 - 5 %), Perseverance takes this error into consideration to grow the hazardous areas on the scenario and also performs autonomous attitude corrections from time to time by tracking the location of the Sun. A similar approach will be used by the Rosalind Franklin ExoMars rover, i.e. visual odometry fused with wheel and inertial odometry information (Bora et al., 2017).

Finally, it is of interest to mention the SLAM methods. The concept of SLAM tries to cope with the inherent coupled uncertainty of localization and mapping when navigating in an unknown environment: the system cannot localize itself in a map that has not been built yet; the map cannot be properly generated if the location where the map information was obtained is undetermined. Consequently, SLAM methods tackle the uncertainty by performing localization and mapping coordinately, which has demonstrated great results on Earth in spite of heavy computational efforts (Stachniss, Leonard, and Thrun, 2016). This computational constraint is the reason why SLAM has not been brought to space exploration yet, although researchers are investigating new methodologies to achieve efficient planetary SLAM. As an example, particle filters and scan matching are used in (Geromichalos et al.,

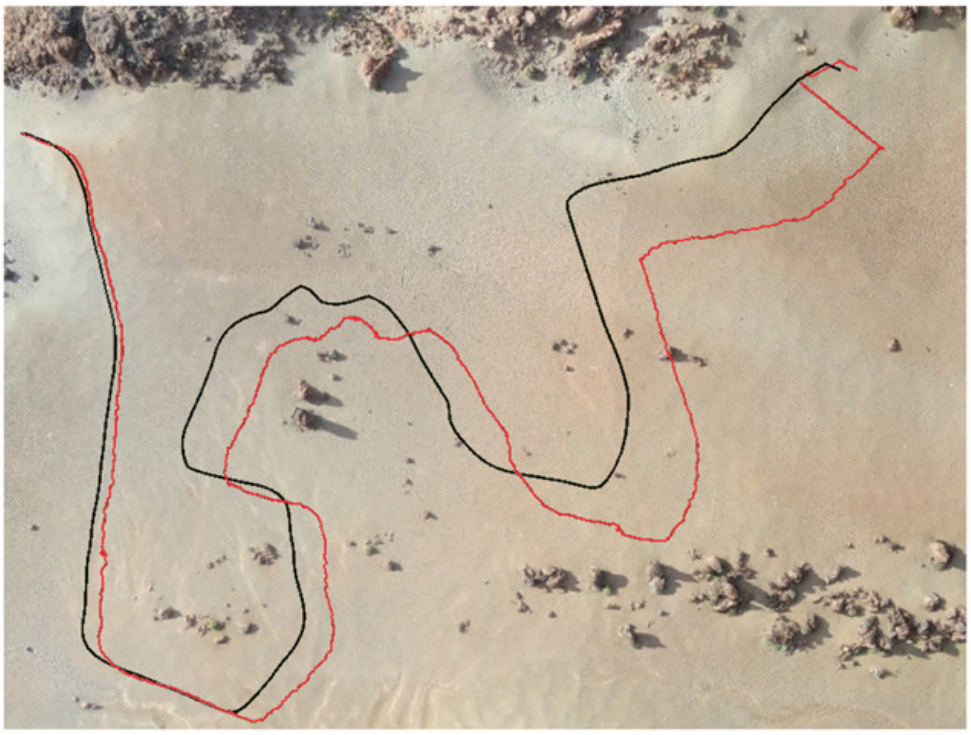


FIGURE 2.4: Estimated positions of a rover traverse using SLAM localization (red) vs ground-truth (black). The SLAM algorithm is able to perform a global pose correction at the end, minimizing the localization error (Geromichalos et al., 2020).

2020) for relative localization and global pose corrections of a planetary rover, achieving a pose estimation error of 0.5 - 1 % of the traversed distance as shown in Figure 2.4.

Path planning

The information gathered by the perception components (mapping, localization) is used to decide how to steer the robot to reach the mission targets, i.e. path planning. Extensively studied in the robotics literature, path planning can be defined as finding a feasible set of waypoints to reach a goal from the current position of the robot, considering the system locomotion capabilities and the characteristics of the scenario, mainly obstacles and non-traversable areas (slopes, sandy soils). Though there is a vast variety of solutions to the path planning problem in the literature, see (Sánchez-Ibáñez, Pérez-Del-pulgar, and García-Cerezo, 2021) for more details, let us focus on those algorithms applied to the use case of planetary exploration.

As extensively explained above, the characteristics of a planetary scenario are generally obtained from maps in two bi-dimensional layers, the global and the local ones. The global layer is used to generate a global path that gives the system a general plan to reach the target. To date, global paths of Mars exploration missions are manually, offline generated by ground operators, by drawing a series of waypoints on orbital imagery to avoid big obstacles or difficult-to-traverse areas, commonly called keep-out zones. Autonomous navigation comes into place, then, through the

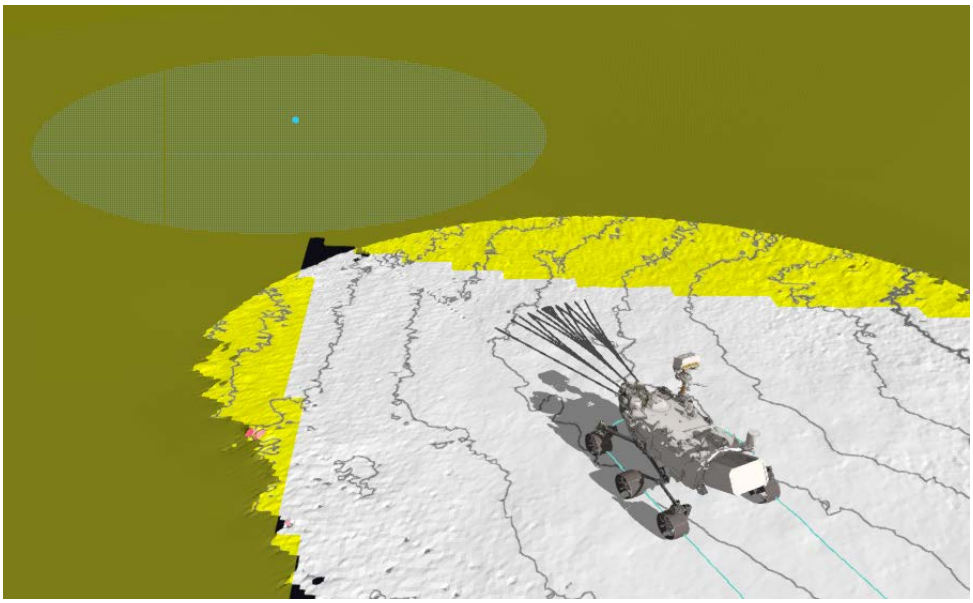


FIGURE 2.5: AutoNav data on Perseverance, evaluating some candidate arcs (grey lines) to get to the goal (blue circle) avoiding the keep-out zones (yellow areas) (Rankin et al., 2023).

local layer, which provides more detailed information on the rover's surroundings. Therefore, the path planner can repair the path online if new obstacles are found.

The first Martian rovers including autonomous path planning capabilities were the MERs with the so-called AutoNav. This software generates a series of arcs from the rover position and chooses one among them to continue the drive. A voting process selects the best arc to follow, considering three main aspects: hazards, steering time, and distance to the goal. Nevertheless, sometimes the voting process failed to find a suitable trajectory on the MERs if a complex arrangement of hazards was blocking the path, due to the simplicity of the method. To solve it, AutoNav was later extended to include a Field D* path planner (Carsten et al., 2009). Field D* is a grid-based algorithm using interpolation techniques to generate direct, low-cost paths on bi-dimensional maps (Ferguson and Stentz, 2006). As a result, the MERs were able to navigate around hazards in a more robust manner, using the path planner when the arc selection failed.

A similar AutoNav was later included in the Curiosity rover, using Field D* to steer the rover around hazards and the so-called keep-out zones (Rankin et al., 2021). Afterward, AutoNav was noticeably improved in Perseverance with the Approximate Clearance Evaluation Approximate Clearance Evaluation (ACE) algorithm (Verma et al., 2023). ACE is a kinematics solver able to estimate the potential wheel footprints and the clearance of the rover's belly, hence, evaluating more realistically the candidate arcs than if considering the rover as a circle encompassing the whole rover body. Therefore, Perseverance is able to traverse safely even in areas with a high density of hazards. Figure 2.5 exemplifies the path planning operation on Perseverance's AutoNav, evaluating a set of candidate arcs to get to the next global

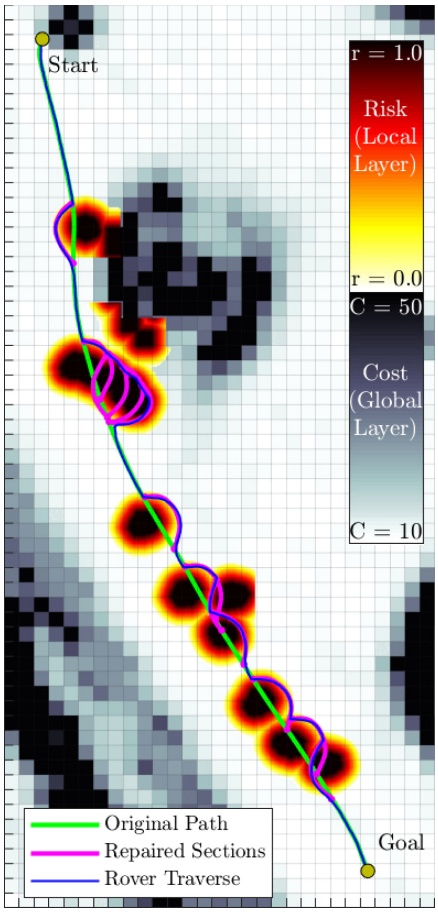


FIGURE 2.6: Example of a FMM-based path planner generating global and local trajectories in a planetary exploration use case (Sánchez-Ibáñez et al., 2019b).

waypoint.

Path planning for the Rosalind Franklin ExoMars rover will work similarly to NASA's method, i.e. a global path generated by ground operators and a local path planner able to react to new hazards. In this case, A* is the selected local path planning algorithm (Bora et al., 2017). A* is a classic path planning method (Hart, Nilsson, and Raphael, 1968), which later originated the Field D* algorithm mentioned above. It uses a heuristic to evaluate the cost of the candidate paths to the goal, thus accelerating the search process.

To date, path planners on Martian rovers use old methodologies that are simple, robust, and predictable. However, more advanced algorithms could eventually provide much better solutions. Let us highlight some modern path planners such as the ones based on FMM (Sethian, 1999), able to find the best solution with comparable computational costs to classic graph-search or sampling-based planners. FMM-based solvers always return the globally optimal, continuous, and smooth path to reach the goal from any point of the scenario if the solution exists, which is called completeness. Besides, FMM can be used in multiple layers to tackle global and local path planning (Sánchez-Ibáñez et al., 2019b), as shown in Figure 2.6. Optimality, smoothness, and completeness are crucial features for space missions, thus FMM is

starting to be tested in planetary exploration use cases (Gerdes et al., 2020). A similar approach is the Fast Sweeping Method (FSM) (Zhao, 2004), which also allows to consider curvature and orientation (Takei and Tsai, 2013) with the drawback of a slightly higher computational cost (Gomez et al., 2019). Remark also the Ordered Upwind Method (OUM) (Shum, Morris, and Khajepour, 2016), which has demonstrated to minimize the energy spent on the planned paths if combined with an anisotropic cost model. Anisotropic cost means a cost function that considers not only scalar values but also directions. This is useful to later obtain paths that minimize the risk of tip-over or slippage, for instance, approaching the slopes in a more perpendicular manner (Sánchez-Ibáñez et al., 2023).

Trajectory following

Once the path to reach the next target has been planned, the rover has to follow it, navigating through the different waypoints that compose the path until the goal is reached. The component that includes this capability is commonly called trajectory follower or trajectory controller. Basically, a trajectory follower continuously generates motion commands (linear and rotational speeds) for the robot to track the trajectory given its position. Like any controller, if external disturbances (localization errors, slippage) affect the robot's motion eventually provoking it to deviate from the trajectory, then the follower compensates for those disturbances by updating the motion commands to recover the right track.

Initially, Martian rovers did not include a trajectory follower as such. Instead, AutoNav would drive predefined distances along the planned arcs in an open loop, in view that the maneuver required to cover a fixed-curvature arc is geometrically defined. The accumulated positioning errors were compensated in the next plan, considering that the plan-drive loop covered only short distances. Thus, this loop was continuously repeated (Carsten et al., 2009). This approach had a main drawback when traversing highly slippery soils due to accumulating drift, therefore, NASA/JPL researchers developed a pure pursuit trajectory follower which was able to correct the motion commands considering the estimated slip (Helmick et al., 2006). Pure pursuit is one of the most renowned algorithms for trajectory following (Wallace et al., 1985), which computes the robot velocities according to geometric quantities defined by the user. Among others, these quantities include the look-ahead distance, i.e. how far is the next point in the trajectory that we are trying to reach, and the width of the safety corridor, i.e. the maximum distance that the robot is allowed to deviate from the trajectory before considering it cannot recover.

The Curiosity rover has the same AutoNav setup for trajectory planning and following. Nevertheless, a significant amount of damage on its wheels motivated a further development of the trajectory controller. Since the rover used to drive assuming a flat terrain, the velocity of the wheels was kept constant even if a wheel stepped on a rock. Hence, the wheel on the rock suffered higher forces and friction that eventually caused punctures and damage. The solution was a traction control

algorithm, which modifies the velocity of each wheel individually when one of the wheels is stepping on a rock or on rough soil (Toupet et al., 2020). This algorithm has kept Curiosity's wheels in acceptable conditions, so the rover has exceeded its design driving distance of 20 km (Rankin et al., 2021) and is still exploring Mars to date.

Perseverance, again, has a similar trajectory controller within its AutoNav, though including a notable improvement thanks to the computational capacity increase. This improvement consists of thinking while driving, i.e. the next plans and motion maneuvers are computed while the previous one has not finished yet, blending the motion commands if the planned arcs have similar curvature and direction (Verma et al., 2023). In consequence, Perseverance does not need to stop between plans, which results in a smoother and more continuous trajectory following.

Finally, regarding Rosalind Franklin ExoMars rover, it does include a closed-loop trajectory following functionality (Bora et al., 2017). Considering that ExoMars has steering actuators in all of its six wheels, the trajectory controller commands crabbing maneuvers (all six wheels steered to the same direction) to compensate the lateral errors w.r.t. the path, and full-Ackermann maneuvers or turns on-the-spot to correct the heading error.

2.3 Goal-constrained motion planning for MM

The motion planning problem of a robot can be defined as finding a feasible trajectory for each actuator of the system to perform a goal task by interacting with the real world (Latombe, 1991). Planning the motion for MM is a particularization of this general problem, i.e. finding a feasible trajectory of the mobile base and configurations of the manipulator in order to place the arm's end effector in certain poses, to perform a desired task. This entails many more challenges in comparison to planning a path for a mobile robot, considering a mobile manipulator is a complex system with multiple constraints and intricate system dynamics (Sandakalum and Ang, 2022). First, mobile manipulators are usually over-actuated or redundant, generally with 3 DoF for the mobile base and several more for the manipulator. Over-actuation provides flexibility since there are infinite possible manners to perform the desired task, conversely, the motion planning problem is hindered since it has to select only one of those solutions. Second, the coupling between the manipulator and the mobile base generates particular dynamic behaviors, which are not easy to handle efficiently by a motion planner. Third, mobile bases widely present non-holonomic constraints, which prevent the base from moving in certain directions in the function of its orientation, obstructing the planning process. On top of these three issues, any robot motion planner needs to consider collision avoidance, energy consumption, time spent, computational cost, joint limits, task fulfillment precision, and uncertainty.

The straightforward solution to tackle the aforementioned issues is to decouple the mobile base and the manipulator plans, moving the manipulator exclusively when the base is close to the goal, or when the arm could collide with the environment (Pilania and Gupta, 2018). That is the current solution in MM for planetary exploration, however, future missions present further challenges. Planetary missions' energy and time constraints demand high efficiency, which cannot be achieved with separate base-arm planning since the rover is missing the chance of moving the arm at the same time the base is reaching the goal. Additionally, high predictability is needed to be sure the mission is not going to be jeopardized before starting the motion. Reactive approaches for MM like the holistic one presented in (Haviland, Sunderhauf, and Corke, 2022) are not appropriate thereby, since there is no long-term awareness of how the system is going to behave. Nevertheless, short-term motion planning and reactive control can be applied to other space exploration use cases, as is later discussed in Section 2.4. Besides, the limited computational capacity of space exploration systems (Gerdes et al., 2020) impedes the use of demanding motion planners in processing time terms.

In summary, planning the motion for MM in space requires an energy/time efficient and predictable algorithm, without severely impacting the computational resources of the robot. These requirements are somehow contradictory, as can be concluded by observing the two main types of MM motion planners. On the one hand, sub-optimal motion planners are computationally lightweight (Sandakalum and Ang, 2022). But, as their name suggests, these planners never find the globally optimal solution for the problem, e.g. can not provide high efficiency on energy or time terms. On the other hand, optimal motion planners do maximize the motion efficiency, although they have a main drawback on their computational complexity, being their convergence speed expressly sensitive to constraints (Kabir et al., 2021) such as actuator limits, the base non-holonomic constraints, or collision avoidance. Let us get into details with a deep analysis of both types of MM motion planners.

Sub-optimal motion planning

Let us denominate sub-optimal to any algorithm that does not find a locally or globally optimal solution for a particular problem, but finds a feasible solution without optimizing any particular parameter. In the case of MM motion planning, algorithms that do find an optimal motion plan for one of the subsystems independently (mobile base or manipulator), but not for the whole system as one, are also considered sub-optimal. Sub-optimal motion planners are characterized by their low computational cost, which usually outplays any optimization-based method. Nevertheless, by definition, they flaws in the quality of the provided solutions mainly in terms of performance efficiency, i.e. invested time and energy. Additionally, some sub-optimal planners do not generate continuous or smooth motions, which hinders the later execution by the robot.

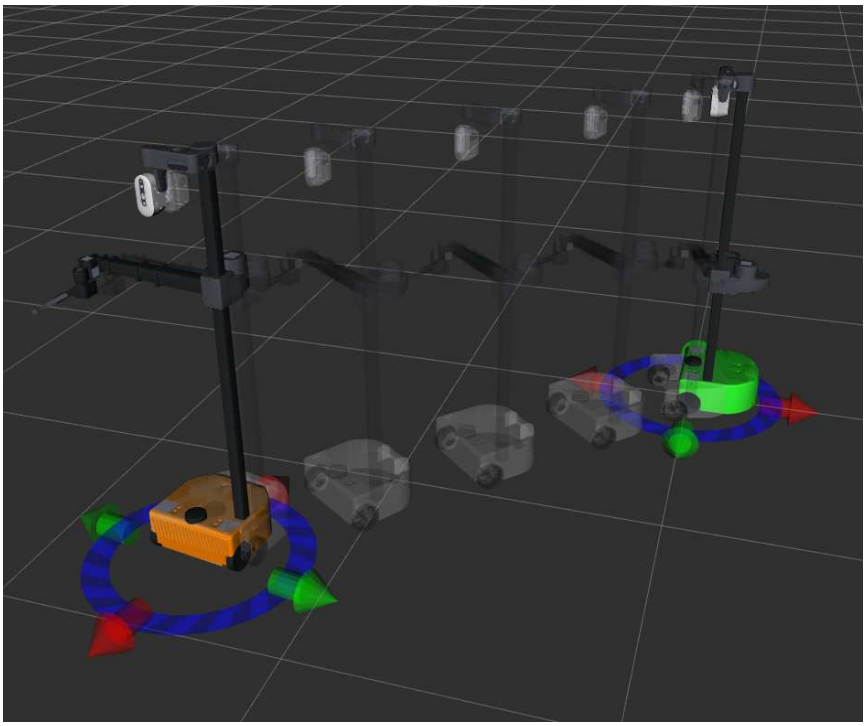


FIGURE 2.7: Example of a sub-optimal MM motion plan generated by MoveIt for the Stretch RE1 robot.

Decoupled motion planners are a clear example of sub-optimal motion planning, such as the Hierarchical and Adaptive Mobile Manipulator Planner with Base pose Uncertainty and its propagation to Arm motions (HAMP-BUA), presented in (Pilania and Gupta, 2018). This sub-optimal planner maintains a constant manipulator configuration until a possible collision with the environment is detected, or until the goal pose for the mobile base is reached. Another decoupled approach is available in the widely-known open-source motion planning library MoveIt[1], mainly used for robotic arms but also including plugins for MM. MoveIt MM motion planner follows a decoupled sub-optimal approach, combining a great variety of path planners for the mobile base, like the Dijkstra graph-search algorithm (Dijkstra, 1959) or A* (Hart, Nilsson, and Raphael, 1968), with several others for the manipulator, such as Rapidly exploring Random Trees (RRT) Connect (Kuffner and La Valle, 2000) or Probabilistic RoadMaps (Kavraki et al., 1996). note that RRT is a very popular algorithm mostly used in 2D path planning (LaValle, 1998), which has evolved throughout the years to solve a great variety of problems. An example of a MoveIt planned motion for the differential-drive robot Stretch RE1[2] and its manipulator can be observed in Figure 2.7. Note that the motion plan, although decoupled, is represented in Figure 2.7 as being performed at the same time.

As can be observed, decoupled motion planners commonly combine a Cartesian path planner for the mobile base with a joint space (or configuration space) motion

[1] `https://moveit.ros.org/`
[2] `https://hello-robot.com/product`

planner for the manipulator. In this regard, as aforementioned, there is a vast literature on path-planning algorithms that are used to plan the Cartesian trajectory of the mobile base, check Nav2[3] as one of the most renowned open-source libraries for autonomous navigation of mobile robots. See also (Sánchez-Ibáñez, Pérez-Del-pulgar, and García-Cerezo, 2021) for a recent and complete review of path planning algorithms.

Although decoupled solutions are simple and reliable, they lack the time/energy efficiency that a coupled motion provides. As a result, authors have lately investigated sub-optimal coupled MM motion planning, aiming to maintain a low computational cost and high reliability, while increasing performance efficiency. To do so, the straightforward and more intuitive approach is to first compute the mobile base path and later obtain the manipulator configurations in accordance with the base path. Optimized Hierarchical Mobile Manipulator Planner (OHMP) is a good example of a coupled motion planner (Li et al., 2020), which first computes a 2D mobile base path using Probabilistic RoadMaps (PRM)s, and, second, generates the manipulator configurations by transitioning between safe configurations whenever a collision state is detected. OHMP demonstrates high efficiency in solving the motion planning problem in complex static environments.

Literature also includes sub-optimal planners which consider the whole mobile manipulator as one single high-DoF system for planning. This is the case of the algorithm developed in (Pajak and Pajak, 2017). With the main focus on constraint compliance, penalty functions at the acceleration level are used to avoid the manipulator singularities and respect constraints such as the non-holonomic base. Additionally, a reactive algorithm modifies the motion of the whole robot in the vicinity of obstacles, getting the system out of danger. Another good example of a full-body MM motion planner is BI^{2}RRT* (Burget, Bennewitz, and Burgard, 2016), which is used to plan the motion of a mobile manipulator to transport liquids or pull carts. BI^{2}RRT* is an asymptotically optimal motion planner that generates solutions faster than standard RRT based planners. An improvement of RRT called RRT-GoalBias (Shao et al., 2021) is also able to quickly solve the MM motion planning problem, considering the full system constraints and obstacle avoidance. Nevertheless, it requires a further post-processing step to smooth the initially planned path, a step that is common in sub-optimal planners, particularly RRT-based ones.

To sum up, sub-optimal MM motion planners can be divided into decoupled, coupled, and full-body solutions. Decoupled MM planners only move the manipulator when the base is stopped, which is simpler but less efficient. Coupled MM planners boost efficiency by moving base and arm coordinately, maintaining a low computational cost. Full-body sub-optimal MM planners consider the whole robot as one, which raises further the motion efficiency but also increases the complexity of the algorithm, i.e. being middle-way between a sub-optimal coupled MM planner and an optimal one. Although more intuitive and computationally lightweight,

[3] `https://navigation.ros.org/`

these three types of sub-optimal motion planners are generally not comparable to the maximized efficiency that optimization provides, as reviewed in the following.

Optimal motion planning

Motion planners based on optimization techniques generally provide high-quality solutions at the expense of big computational efforts (the inverse to sub-optimal ones). This is due to the high-dimensional aspect of the MM motion planning problem, there are infinite manners to accomplish the goal tasks, i.e. infinite combinations of joint movements to reach the goal from the current state of the robot. From all these possibilities, the optimization algorithm has to select the one that gives the best outcome according to some selected criteria, like energy consumption, execution time, or distance to collisions. This generates an optimized solution after a computationally heavy search.

Let us introduce the Linear Quadratic Regulator (LQR), which is a well-known method used in optimal control. LQR can consider non-linearities if executed sequentially, solving at each iteration a linearized convex optimization sub-problem that drives the planner closer to the solution of the complete non-linear and non-convex problem. This approach is commonly called sequential convex optimization, e.g. Iterative Linear Quadratic Regulator (iLQR) (Li and Todorov, 2004) or Sequential Linear Quadratic Regulator (SLQ) (Sideris and Bobrow, 2005). SLQ and iLQR, as many other optimal control-based algorithms, find severe difficulties in considering collision avoidance, and system limits, i.e. constraints.

Constraints, although inevitable, impose behaviors that increase the problem's complexity and slow down the algorithm convergence (Todorov and Li, 2005, Sideris and Rodriguez, 2011). Authors in the motion planning field usually differentiate two types of constraints: equality and inequality constraints. Equality constraints are requirements that must be satisfied for the task to be completed, like joint position goals or end effector trajectories, or kinematics consistency demands, like a non-holonomic mobile base. These are easier to tackle for the optimizer since they notably reduce the available space of solutions where the algorithm has to search, imposing particular robot states at particular times. For instance, over-actuation and non-holonomic constraints are efficiently tackled with the kinematic planner presented in (Giftthaler et al., 2017), which makes use of Constrained SLQ (Sideris and Rodriguez, 2011). This method, integrated into an MPC loop, achieves impressive controlling frequencies (in the order of 50 Hz) while following MM end effector trajectories, although only applying equality constraints. Remark here that such a high frequency is achieved since MPC only computes the next few actuation commands, not a whole, long-term motion plan.

On the contrary, inequality constraints define boundaries that should not be overcome, such as joint limits or obstacles. These are much more difficult to handle for the optimizer since it has to ensure they are not violated anytime during

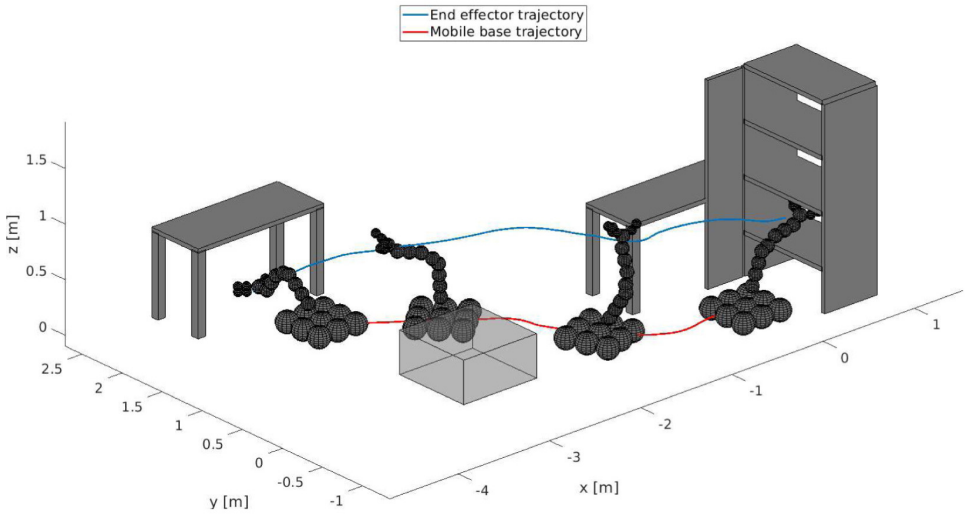


FIGURE 2.8: Example of an optimized collision-free motion plan for a mobile manipulator, with an omnidirectional mobile base in a static indoor environment (Petrović et al., 2020).

the motion. A renowned example of constraint-compliant optimization MM motion planner is the Stochastic Trajectory Optimization for Motion Planning (STOMP) (Kalakrishnan et al., 2011), which iteratively optimizes derivative-free cost functions to generate smooth motions, considering obstacles avoidance, task constraints, or motor torques. A similar stochastic optimization method for trajectory planning is presented in (Petrović et al., 2020), where the authors represent the robot trajectories as samples of a continuous-time heteroscedastic Gaussian Process (GP). This leads to collision-free, smooth trajectories that are less prone to local minima than gradient-based methods, as shown in the motion plan example of Figure 2.8. Another celebrated motion planner is Trajectory Optimization (TrajOpt) (Schulman et al., 2014), a sequentially quadratic programming convex optimization algorithm that generates collision-free trajectories for high-DoF robots such as mobile manipulators. TrajOpt introduces a penalty-increasing method that efficiently drives the system to comply with constraints.

Lately, researchers have focused on reducing the processing load of constraint-compliance optimization, trying to accelerate the convergence speed to make the most of these algorithms while maintaining an affordable computational effort (which is particularly critical for space applications). A usual approach to achieve this is to effectively describe the system. Consequently, authors in (Liao et al., 2019) propose a complete MM kinematic model, describing transfer and Jacobian matrices to tackle over-actuation. Afterward, the optimization motion planner is divided into two separate loops: an outer loop to penalize inequality constraints, like joint limits or obstacles, and an inner loop that finds the collision-free trajectory. Interior-point methods are used to solve the inequality constraints, and the Newton method for the equality constraints, which are mainly the goal end effector poses. Additionally,

a Virtual Kinematic Chain (VKC) is proposed in (Jiao et al., 2021), which is a distinctive MM modeling method that integrates the mobile base and arm kinematics, and the object to be manipulated. VKCs improved the success ratio of the trajectory optimization algorithm, which was used to plan the robot's motion to perform daily tasks in a household environment. Considering joint kinematic limits, collision avoidance and the task goals as inequality constraints led the algorithm to converge in less than 10 s, generating the whole motion plan for the mobile manipulator to open a door or pick up an object.

A different but equally interesting approach to accelerate the convergence speed of sequential convex optimization algorithms is the so-called warm starting. To warm start is to begin the optimization process from an initial solution to the problem. This solution does not need to be feasible, it will just place the algorithm close to a candidate solution. Herewith, the optimizer is boosted to converge faster since it starts nearer to the convex area neighboring the final solution. Warm-start is deeply studied in (Thakar et al., 2019), where the authors propose a sequential refinement method for optimization that keeps introducing problem constraints (grasping pose, gripper speed, collisions) regularly to the optimizer, continuously warm-starting and polishing the solution. This way the motion planner performance is improved in comparison with the cold-started planner, which was demonstrated by generating trajectories for MM pick-up tasks. Another warm start example for over-actuated MM is presented in (Lembono et al., 2020), where k-Nearest Neighbor (k-NN), Gaussian Process Regressor (GPR) or Bayesian Gaussian Mixture Regressor (BGMR) were used as approximation methods to warm start, in parallel, multiple trajectory optimizations. As a result, the motion planner was significantly improved, with a 71 % success rate and four times faster convergence when planning the motion of the 34 DoF humanoid robot Atlas (Boston Dynamics) to reach a given Cartesian pose. Finally, it is also fascinating to mention the use of artificial intelligence, specifically deep learning, to improve the performance of optimization-based motion planning algorithms, as shown in (Ichnowski et al., 2020). Here, two different neural networks are used to warm start the optimization algorithm, improving noticeably its overall performance in terms of computational cost.

2.4 Reactiveness: Control and trajectory following for MM

Once generated, the motion plan for the robot has to be executed. Ideally, this motion plan could be directly brought to the platform by commanding each of the robot joints as designed, nevertheless, real platforms present further inconveniences. First, the model's intrinsic errors. Any model has a balance regarding complexity vs accuracy, which induces some errors w.r.t. the real behavior of the robot. Second, the non-ideal performance of the sensors and actuators, i.e. dead zones, calibration errors, or coupling wear. These issues can not be always prevented or solved before a motion plan execution. Third, external disturbances. In planetary exploration, these

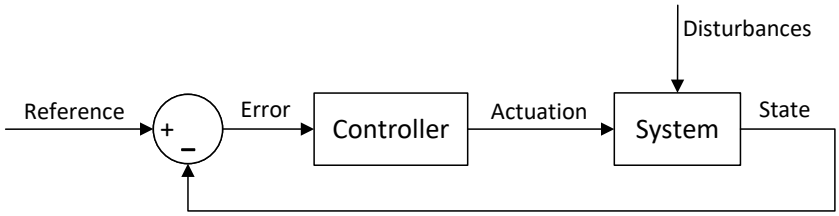


FIGURE 2.9: Basic control scheme to reach a desired reference considering external disturbances applied to the system.

are mainly system navigation errors, such as localization errors or mapping uncertainty. Localization errors, mainly caused by slippery soils, produce a drift in the motion of the system w.r.t. the plan. This drift can lead the rover away from its goal, eventually reaching hazardous areas of the scenario. Additionally, mapping uncertainty raises the necessity of continuously checking if there are new obstacles surrounding the rover that could harm the integrity of the system.

As a result, a controller is required to tackle the non-ideal behavior of the robot, letting the system properly follow any planned motion with an admissible level of error, as well as reacting to unexpected events. One option is a basic error-feedback controller, as illustrated in Figure 2.9. This classical approach simply makes use of the sensed information regarding the robot's state, to compare it with the desired motion reference and estimate the current error, which is compensated by a subsequent actuation to the system. Although this basic scheme can be used to solve a great number of control problems in robotics (Siciliano et al., 2009), the MM trajectory following problem requires more advanced techniques.

The traditional widely-used approach to follow MM motion plans is the use of two coordinated controllers to track simultaneously the reference trajectories for both systems, the mobile base and the manipulator. This type of control is quite straightforward and computationally negligible, however, it lacks the flexibility, versatility, and efficiency that a whole-body controller provides. Whole-body control is, in fact, the current trend, being MPC the most common method to achieve it (Camacho and Bordons, 2007). MPC gives the system a deeper self-knowledge before deciding how to command the joints, which allows the robot to quickly react to any unexpected disturbance meanwhile following the desired references as closely as possible. The main drawback of MPC is its computational demand, which is even greater if considering the system dynamics. Additionally, the latest advances in control theory have favored the development of reactive planning algorithms, i.e. control algorithms able to generate local plans for the system in real-time. Reactive planning is useful to reach local high-level goals meanwhile reacting to unexpected events, although the behavior is somehow unpredictable in the sense that there is no guarantee of how the system is going to behave in the long term. Thus, let us analyze the state of technology regarding these approaches: coordinated controllers, MPCs, and reactive planning.

Coordinated control

The most intuitive approach for a mobile manipulator to execute a motion plan is to include two controllers, one for the mobile base and another one for the robotic arm. The two controllers have to be coordinated though, considering that the objective is to precisely follow the planned trajectories for the mobile base and the end effector. If not coordinated, there is no guarantee that the performed motion corresponds to the planned one.

One of the first examples of coordinated control for mobile manipulators is proposed in (Egerstedt and Hu, 2000), with a globally stable, coordinated control algorithm to follow end effector trajectories. This controller is based on two concepts: the coordinated evolution of the reference points on the two trajectories (mobile base and end effector), and the use of controllers augmented by position and orientation error feedback. Similarly, in (Hentout, 2011) a strategy to follow end effector trajectories is presented, focusing on non-holonomic wheeled mobile manipulators. Several controller agents place the end effector as close as possible to the preferred configuration, while the mobile base avoids the obstacles in the scenario. Nevertheless, these approaches are very sensitive to drifts in the mobile base localization, notably affecting the accuracy of the end effector positioning.

Regardless of the existence of more advanced control techniques, as the ones explained later, coordinated control is still a useful approach for MM when robustness and low computational effort are required. For instance, (Burgess-Limerick et al., 2023) presents reactive manipulation on-the-move, which allows performing MM tasks such as a pick and place given a series of manipulation targets. This is done using a time-aware coupled controller that reacts to disturbances, avoids obstacles, and maximizes the manipulability of the system. Remark that the manipulability measures the capacity of the arm to achieve any velocity in a particular configuration, which is directly related to the distance of that configuration to the manipulator's singularities. Besides, this method is highly efficient due to its capacity to pick objects at the same time the mobile base is driving to the place goal.

Model Predictive Control

Predictive controllers provide flexibility, handling of constraints and non-linearities, and high performance (García, Prett, and Morari, 1989). As a result, they have been very popular in the industry since the late eighties, a popularity that has been growing and propagating into the robotics community, with MPC the robotics predictive controller par excellence. Whole-body MPC has been used to generate walking motions online (Herdt et al., 2010), track trajectories for mobile robots (Fnadi et al., 2021) or hexacopters (Neunert et al., 2016), enable quadrupedal locomotion (Farshidian et al., 2017) or even quadrupedal MM (Sleiman et al., 2021).

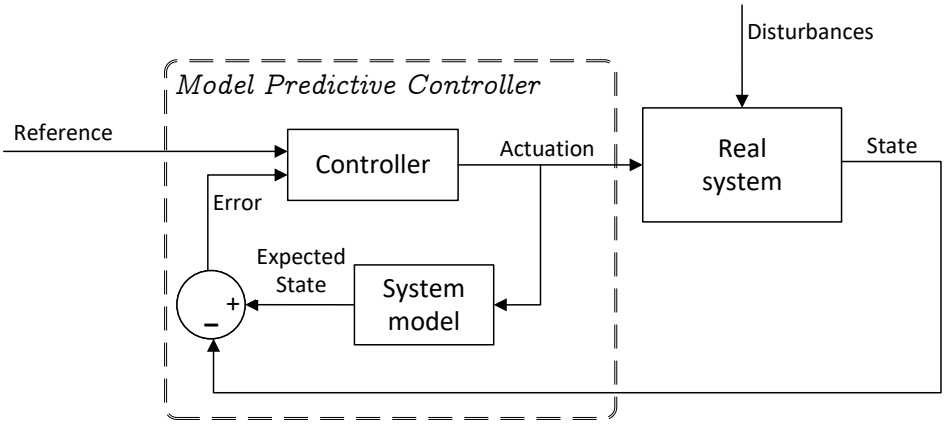


FIGURE 2.10: Basic MPC scheme.

In summary, MPC behaves like any standard controller but includes a prediction step using a model of the system, as shown in Figure 2.10. This model is used to estimate how the system should evolve after applying the controlled actuation. Then, the estimation is compared with the real measured state of the system, thus computing a tracking error that includes the effect of external disturbances. The tracking error is fed back to the controller, which is usually an optimization algorithm, making it more aware of how the system is evolving and will evolve.

Focusing on MM, MPC is a great solution to deal with constraints. This is demonstrated in (Avanzini, Zanchettin, and Rocco, 2018) with a constrained optimization MPC for tracking MM trajectories, tested with the Keller und Knappich Augsburg (KUKA) youBot mobile manipulator. The proposed MPC guarantees offset-free tracking, recursive feasibility, convergence, and stability, including constraints such as position, velocity and acceleration limits, collision avoidance, or camera Field of View (FoV) requisites. MPC also provides fast reactive capabilities to the system. This is shown, for instance, with the stabilization of the inherently unsteady ball-balancing robot presented in (Minniti et al., 2019). An SLQ-based MPC is integrated into the control loop, enabling MM tasks such as tracking end effector poses, following manual movements of a human user, or opening doors. Another example of real-time reactive MPC is presented in (Li and Xiong, 2019), with a nonlinear MPC implementation used for the avoidance of dynamic obstacles in goal-constrained MM. This approach makes use of the global robot Jacobian to generate the control rule, maximizing the distance to obstacles. Moreover, tracking end effector trajectories is one of the most common applications of MPC for MM. For instance, in (Pankert and Hutter, 2020) a receding horizon MPC scheme is used to track end effector task-space trajectories for a mobile manipulator, including visual information to avoid obstacles while respecting joint limits and stability constraints. MPC can be even used to track a reference lawnmower pattern trajectory for vineyard spraying. This is shown in (Vatavuk, Vasiljevic, and Kovacic, 2022), where MPC reduces unnecessary spraying waste and pollution while minimizing the robot accelerations and tool displacements.

One of the main issues when using MPC is the assessment of the control frequency vs the prediction time horizon. Also known as receding horizon optimal control, MPC generally defines a short finite time horizon where the prediction is performed, which keeps receding, or moving forward until the global goal is reached. Generating the actuation trajectory for the complete prediction horizon improves noticeably the smoothness of the solution since the motion is thoroughly optimized, i.e. from the current state of the system until the terminal set point, not only for the next few time steps. This is suggested with Nonlinear Model Predictive Horizon (NMPH) (Younes and Barczyk, 2021), nevertheless, a long prediction requires bigger computational efforts that imply reducing the control frequency. This balance between frequency and quality is being studied lately in the literature, searching for intermediate solutions to generate high-frequency smooth controls. The event-triggered replanning presented in (Luis, Vukosavljev, and Schoellig, 2020) is a remarkable option, which improves the smoothness of the robot movements in comparison to standard MPC approaches without severely impacting the computational resources. This is achieved by matching the reference for the motion planner with the platform's current state only when an external disturbance is detected, as a replanning algorithm does. Hence, the jittering, generated by the actuation discontinuities, is removed. Additionally, an adaptive fuzzy-logic system is used in (Yuan et al., 2023) to approximate the dynamics, which is the most computationally costly aspect of optimization algorithms as previously explained in Section 2.3. Consequently, faster dynamics estimation allows for the elongation of the prediction horizon. This fuzzy whole-body non-holonomic MPC makes use of a primal-dual neural network that solves the constrained quadratic programming problem, applied to a mobile dual-arm robot.

Reactive planning

Short-term control strategies are not always able to deal with uncertain, dynamic environments, major external disturbances, or unforeseen events if the severity of the disturbance is noteworthy. Such cases require a significant update of the planned motion with a reactive planning approach, i.e. an algorithm able to quickly generate a local replan to surpass the arising hazard. Note that this is an intermediate level between low-level control and global planning. Indeed, reactive planning should not substitute the global planning level, since it relies on local information that usually sinks in local minima, overlooking the mid and long-term behavior of the system. Consequently, safety-critical use cases like planetary exploration should consider reactive planning as an additional feature in the motion planning scheme, under the global planner, to be used only in the case of the emergence of critical disturbances.

Keeping this in mind, there is an extensive state-of-art regarding reactive planning, some of these works focusing on MM. MPC is a common approach for reactiveness as done in (Ide et al., 2011), where a quadratic programming method

is used to optimize the control inputs of a constrained real-time trajectory planner. This method is integrated into a Reference Hardware 2 mobile manipulator from MAEKAWA MFG, being the system capable of updating the trajectory in real time even if a new target was sent during the motion. But, clearly, MPC is not the only option. A framework for reactive MM is presented in (Dietrich et al., 2012), tested on a torque-controlled wheeled humanoid robot called Rollin' Justin. The proposed whole-body motion concept ensures the safety of the robot, complying with the physical constraints and self-collision avoidance as well as maintaining a proper level of manipulability. Reactive planning can be also achieved by mixing different algorithms, as exemplified with the hybrid control architecture presented in (Vasilopoulos et al., 2021) that uses a discrete logic planner and a continuous reactive controller. It is applied to complex MM tasks for a differential drive 2D robot in a series of simulations, demonstrating the algorithm's capacity to avoid unanticipated obstacles and fulfill the goal manipulation task. Another remarkable example of reactive planning is presented in (Haviland, Sunderhauf, and Corke, 2022), which achieves smooth transitions between tasks by integrating a behavior tree into the reactive planning and control system. This holistic controller is tested with the nonholonomic Frankie mobile manipulator and the holonomic Omni-Frankie one, able to efficiently and robustly perform pick and place tasks in a static indoor scenario.

It is worth mentioning the recent studies on Control Barrier Function (CBF)s and Control Lyapunov Function (CLF)s in robotics (Zeng et al., 2021). Based on quadratic programming, these methods solve safety-critical constraints in real-time, which has been proven to be useful for reactive planning of bipedal robots in complex undulating terrains (Huang and Grizzle, 2023). Although yet to be applied to MM systems, in the next few years CBF-CLF methods could emerge as the solution to whole-body reactive control and planning for computationally-limited systems.

Finally, let us highlight the impressive achievements of Reinforcement Learning (RL) for real-time planning and control, outperforming human pilots in autonomous drone racing (Song et al., 2023). Since RL algorithms are direct task-solvers, fusing the planning and control stages into a single goal-oriented solver, they have boosted versatility and flexibility to cope with external disturbances and model mismatches in comparison to optimal controllers. Anyhow, optimal controllers will find the best solution according to the model, which generally performs better than RL if the model is accurate enough and the external disturbances are not significant (as is usually the case of planetary rovers).

2.5 Summary and conclusions

From the low-level control of actuators to the high-level decision-making, multiple challenges have been surpassed in the last two decades to enable the degree of autonomy of nowadays robots. MM, in particular, has required further effort, considering the high intrinsic complexity of this kind of system, which unites the

issues of the mobile robotics and manipulation worlds into a single, usually over-actuated system. Although straightforwardly solved by separating the intelligence of both subsystems (mobile base and manipulator), present applications demand higher standards on efficiency, robustness, and computational cost, which cannot be reached using the simple decoupled approach. In consequence, state-of-art algorithms focus on coupled and whole-body approaches for autonomous MM, achieving impressive results when guiding and controlling multiple types of MM systems (wheeled, legged, aerial, orbital, submarine...).

Particularly, planetary exploration rovers require very advanced autonomous MM capabilities: energy and time-efficient, safety-critical, predictable, and robust motions, obtained using very limited computational resources. To do so, there are two particularly critical components of the autonomous MM architecture to be considered: the motion planner, and the motion control and reactive subsystems. As thoroughly analyzed above, the state-of-art Earth-focused contributions on these topics have reached impressive results, however, several challenges are yet to be overcome to bring these algorithms into a planetary exploration rover. On the one hand, most authors work with commercial Earth hardware. Space-qualified processors, nevertheless, have substantially less computational power, demanding the software components to be very lightweight and efficient. On the other hand, MM's contributions mainly focus on repetitive, industrial tasks, in completely structured, well-known indoor environments where the main issue is dynamic obstacles, which are usually humans. Planetary scenarios are exactly the opposite: independent and thoroughly different tasks in static unstructured environments, with a big level of uncertainty about the shape of the terrain and the presence of obstacles due to the low quality of the global information.

Taking the aforementioned challenges into consideration, let us depict the main gaps in the literature that can be improved to enable autonomous MM in planetary exploration.

- Seeing that the global information is poor on Mars or the Moon, which causes a lot of uncertainty for an autonomous navigation system, the local information gathered by the rover can be used to enrich the global information of the scenario that the rover has available. Feeding back the local information would reduce the necessity of local replans that decrease the overall time efficiency during the mission. Besides, this local information (obstacles, terramechanics characteristics) can be extrapolated from the current terrain being traversed to similar ones in the scenario, making use of a previous global terrain segmentation map. Such an algorithm would give the rover a better awareness of the scenario to efficiently plan the consequent motions, even more, so if the MM mission is a round-trip or is located in areas with regular characteristics.

- Regarding sub-optimal MM motion planning methods, they are suitable for

space exploration thanks to their low computational cost, however, further research is required to improve two main aspects. On the one hand, time and energy efficiency, which is one of the main drawbacks of sub-optimal MM planners, but is an important matter for increasing the scientific return in planetary missions. On the other hand, robustness, since any planned motion has to be completely safe w.r.t. collisions with the environment and self-collisions of both the mobile base and the manipulator, to ensure there is no chance of a collision threatening the mission.

- Optimal MM motion planners provide efficient, constrained-satisfying motions but demand big computational efforts, even bigger in the case of over-actuated MM. Such a high computational cost is not suitable for a planetary exploration rover. Multiple methods are arising in the literature to reduce the convergence time of the optimization, being warm start a widely-used approach. In fact, multi-staged warm starts have been demonstrated to noticeably reduce the average number of iterations until convergence to solve path-constrained trajectory generation problems.

- Recent planetary exploration missions are calling for new, collaborative use cases for teams of robots, e.g. Lunar caves exploration. To maintain the platforms safe when performing such tasks, whole-body control and reactiveness are essential, e.g. MPC, which would allow to quickly react to external disturbances. Nevertheless, the use on planetary rovers of this type of whole-body, high-frequency control technique is yet to be tested.

This doctoral research has tried to fill these four gaps, to enable autonomous MM in future planetary exploration applications. Let us demonstrate how in the following chapters.

Chapter 3

Dynamic cost map update to enhance global path planning

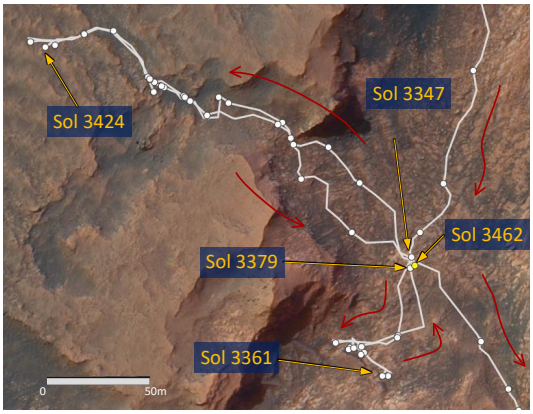
"Καμία απώλεια δεν πρέπει να είναι πιο θλιβερή για εμάς από το να χάσουμε το χρόνο μας, γιατί είναι ανεπανόρθωτη" - "No loss should be more regrettable to us than losing our time, for it's irretrievable"

Zeno of Citium
3rd century B.C.

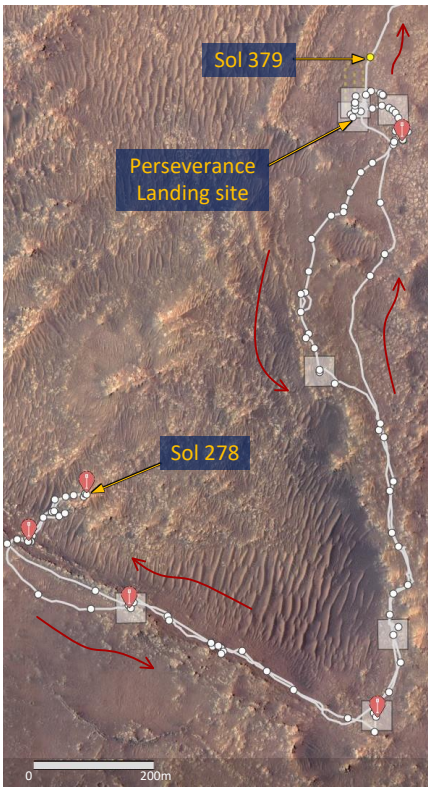
3.1 Introduction

Planetary rovers perform their scientific tasks trying to cover the wider area possible, to reach unexplored zones and uncover the secrets of the corresponding extraterrestrial surface. Nevertheless, sometimes these systems have to undo their steps due to insurmountable obstacles, to come back to the global plan after exploring a specific spot, or after a change in the high-level scientific commands. As a result, it is common for planetary rovers to cover the same area several times, as can be observed in the traverse maps of Spirit[1], Opportunity[2], Curiosity[3], and Perseverance[4]. Some examples of Martian rovers covering twice the same areas are shown in Figure 3.1, particularly from Curiosity (Figure 3.1a) and Perseverance (3.1b). Additionally, there are planetary missions that are directly planned as a round-trip, i.e. a rover has to get to a particular spot of the scenario, perform a task, and then return to the lander. Although now discontinued, this was the case of Sample Fetch Rover (Merlo, Larranaga, and Falkner, 2013), which was supposed to retrieve and bring back to the lander the soil samples left by Perseverance.

[1] https://mars.nasa.gov/mer/mission/tm-spirit/spirit-sol2555.html
[2] https://mars.nasa.gov/resources/22434/opportunitys-final-traverse-map/
[3] https://mars.nasa.gov/msl/mission/where-is-the-rover/
[4] https://mars.nasa.gov/mars2020/mission/where-is-the-rover/



(a) Curiosity's double round-trip from sol 3347 to sol 3462.



(b) Perseverance's round-trip after landing, which ended just before the Rapid Traverse Campaign in sol 379.

FIGURE 3.1: Round-trip examples of real Martian rovers. The white lines represent the estimated rover route, and the red arrows show the traverse direction of the rover. Data from NASA/JPL-Caltech.

An autonomous rover, as stated in Chapter 2.2, usually relies on two layers of information. On the one hand, the global layer, with low-resolution maps from orbital imagery, which is used for global planning. On the other hand, the local layer, with high-resolution maps obtained from local images of the rover, which is used for replanning. This two-independent-layered approach is acceptable if the rover never comes back in its own steps, however, this is not the usual case as stated above. If the rover has to cover the same area again but the global layer has not been updated, then it will go through the same hindrances once again: small obstacles, loose soils, dead-ends... This can be avoided just by dynamically updating the global layer online with the local layer information, i.e. by feeding back the gathered data during the traverse, regarding the detected obstacles and the soil properties. As a result, the motion planner will generate more efficient trajectories when moving on already covered areas, since the rover's awareness of the scenario characteristics is increased. Besides, this local information can be extrapolated to similar areas of the rover environment if using a global segmentation map. Hence, new trajectories to explore unknown areas will also be more efficient in comparison with those paths obtained only using global information.

Therefore, the first contribution of this thesis is presented in this chapter: a dynamic Guidance methodology that takes advantage of the information gathered by a rover during the traverse, to enhance the global path planner when reaching the area of scientific interest for the later MM task. Although with a clear application on round-trips to improve the planned motion during the return traverse, this approach benefits any exploration mission where the global information is not reliable enough, enriching it to get closer to reality. Innovations regarding this work are related to the online update of the global cost map used by the path planner, in two main ways. First, by including the already detected local obstacles to avoid them during the following traverses. Second, by extrapolating dynamically the estimated soil properties to similar areas, exploiting a previous terrain classification of the scenario.

For this purpose, this work has been supported by previous contributions that include: a trajectory control method that ensures the rover will follow a safe corridor (Filip, Azkarate, and Visentin, 2017), a computationally lightweight hazard detector (Gerdes et al., 2020), able to discern between traversable areas and obstacles during the rover traverse, and a global-local path planner with hazard avoidance capability (Sánchez-Ibáñez et al., 2019b), called Dynamic Multilayered path planner (DyMu). This path planner is based in FMM, a numerical scheme that has been used repeatedly throughout this doctoral research and is explained in detail in Section 3.2. Later on, DyMu is thoroughly described in Section 3.3, to analyze how the path planner works prior to feeding back the local information. Afterward, the proposed update methodology is explained, depicting the local obstacles update in Section 3.4 and the soil parameters update in Section 3.5. Finally, Section 3.6 summarizes the contents of this chapter.

3.2 Fast Marching Method

FMMs are numerical schemes to solve the non-linear Eikonal equations efficiently and optimally. These are used in a great variety of applications, e.g. computation of seismic waves, visibility, reflections, or path planning, among others (Sethian, 1999). First of all, FMM requires a proper representation of the scenario as an input cost map Ω. Being p a position vector, in 2D $p = [x\, y]$, and $\tilde{p}_j$ the closest discretized node to p_j given the map origin and resolution, the cost map Ω is defined as a discrete 2D (or 3D) grid, where each regularly scattered node $\tilde{p}_j$ has an associated cost $\Omega(\tilde{p}_j)$ that represents how easy and safe is for the platform to be placed in that position. Subsequently, obstacles should have the highest costs, and traversable areas the lowest ones. Areas surrounding obstacles should also have high costs, to avoid the platform getting close to them. Additionally, any other feature that influences the platform's behavior should be considered in the cost map. For instance, slopes and terramechanic properties of the soil in the case of rovers.

FMM numerically solves a particular non-linear Partial Derivative Equation (PDE) called the Eikonal equation, modeling the rate of propagation of a wave. This wave expands on the cost map Ω from the goal node $\tilde{p}_g$ visiting each node $\tilde{p}_j$ to generate the cost to go $Y(\tilde{p}_j, \tilde{p}_g)$, which indicates the accumulation of cost required to reach the goal $\tilde{p}_g$ from the node $\tilde{p}_j$. As indicated in (3.1), the rate of propagation of the wave at a certain node is equal to the cost at that node $\Omega(\tilde{p}_j)$.

$$||\nabla Y(\tilde{p}_j, \tilde{p}_g)|| = \Omega(\tilde{p}_j) \quad \forall \tilde{p}_j \in \Omega \tag{3.1}$$

The higher the cost $\Omega(\tilde{p}_j)$ the slower the propagation of the wave on that node $\tilde{p}_j$. In this case, the wave propagation starts from the goal $\tilde{p}_g$, therefore, the cost to go of the goal node is zero ($Y(\tilde{p}_j = \tilde{p}_g, \tilde{p}_g) = 0$).

The cost to go between the starting $\tilde{p}_0$ and the goal nodes $\tilde{p}_g$ is the minimum possible if $\Omega(\tilde{p}_j)$ always returns positive non-zero values. Thus, following the Dynamic Programming (DP) principles, any point $p \in \Omega$ is placed in the optimal path connecting the starting and goal nodes, $\Gamma(\tilde{p}_0, \tilde{p}_g)$, if the sum of the costs to go from the starting node to the point $Y(\tilde{p}_0, p)$ and from the point to the goal node $Y(p, \tilde{p}_g)$ is equal to the minimum cost to go $Y(\tilde{p}_0, \tilde{p}_g)$, as expressed in (3.2).

$$Y(\tilde{p}_0, p) + Y(p, \tilde{p}_g) = Y(\tilde{p}_0, \tilde{p}_g) \quad \forall p \in \Gamma(\tilde{p}_0, \tilde{p}_g) \in \Omega \tag{3.2}$$

Hence, the objective of FMM is to solve the optimization problem defined in (3.3-3.4), i.e. finding the optimal path $\Gamma(\tilde{p}_0, \tilde{p}_g)$ that minimizes the cost accumulated along the path $\Omega(\Gamma(\tilde{p}_0, \tilde{p}_g, l))$, being $\Gamma(\tilde{p}_0, \tilde{p}_g, l)$ a continuous function that returns a point $p \in \Omega$ given the path length l from the starting node $\tilde{p}_0$, with l_g the total length of the path.

$$\underset{\Gamma(\tilde{p}_0, \tilde{p}_g)}{\text{Minimize}} \quad Y(\tilde{p}_0, \tilde{p}_g) = \int_0^{l_g} \Omega(\Gamma(\tilde{p}_0, \tilde{p}_g, l))\, dl \tag{3.3}$$

$$\text{with} \quad Y(\tilde{p}_j = \tilde{p}_g, \tilde{p}_g) = 0 \tag{3.4}$$

Conclusively, given a cost map Ω, FMM returns the globally optimal, continuous path to reach a target goal node $\tilde{p}_g$ from a starting node $\tilde{p}_0$. The shape and quality of the path are dependent on the characteristics of the cost map. Let us explain this relationship in the following, together with how a cost map is extracted from a DEM within DyMu.

3.3 Dynamic Multilayered Path Planner

DyMu path planner is the result of a great effort to increase autonomy in mobile robots, particularly planetary exploration vehicles. As one of the contributions of

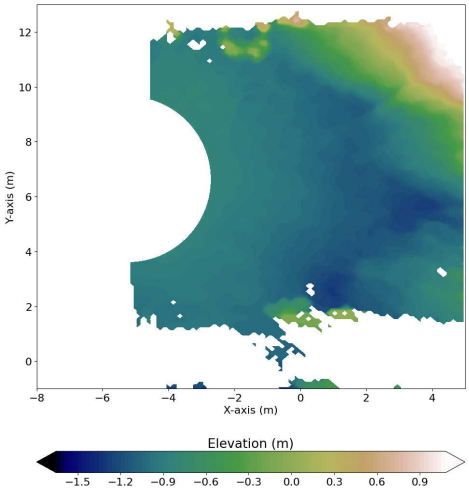
the doctoral thesis of (Sánchez-Ibáñez, 2022) titled "Extreme Path Planning for Exploration Mobile Robots", the goal of this path planner was to make rovers able to safely and autonomously wander on extreme surfaces, to reach hardly accessible areas.

As aforementioned, DyMu requires a description of the scenario as the cost map Ω, which represents with scalar costs the areas that are easier or harder to traverse, or the ones that should be avoided. To generate such a cost map in static and unstructured environments the first step is to obtain a DEM of the scenario, which is a 2D map where each node indicates the height of the terrain at that position. As explained in Chapter 2.2, the DEM can be extracted from aerial or orbital imagery (global DEM), or from the robot (local DEM), using exteroceptive sensors such as cameras or laser scanners.

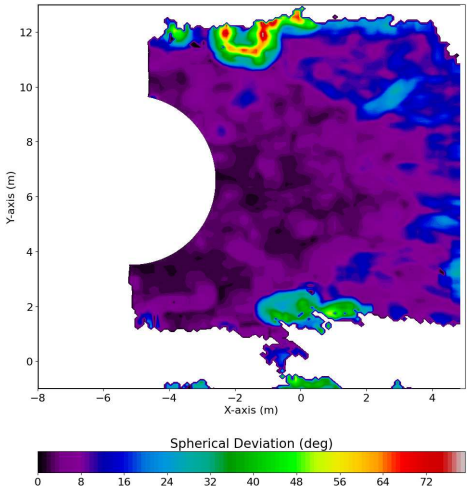
An example of the procedure to generate the cost map from a local DEM is shown in Figure 3.2. From the DEM, Figure 3.2a, the scenario can be characterized by estimating terrain features, detecting obstacles and non-traversable areas. The obstacle detection uses different metrics, such as the roughness, shown in Figure 3.2b, which is directly related to the spherical deviation of the vectors perpendicular to the surface. High roughness is derived from abrupt changes in the DEM, which indicates the presence of obstacles. Additionally, areas with high slopes, shown in Figure 3.2c, may also not be traversable for the robot and thus must be characterized as obstacles. The traversability map $\mathcal{T}$, Figure 3.2d, gathers this classification, assigning a category to each of the grid nodes, e.g. obstacles or safe nodes. It also includes a category for obstacle dilatation, which considers the size of the robot to avoid getting too close to unsafe areas.

Thereafter, the traversability map $\mathcal{T}$ is used to produce the cost map Ω, shown in Figure 3.2e. The cost map Ω assigns high costs to the nodes labeled as obstacles or non-traversable in $\mathcal{T}$, and low costs to the safe ones. DyMu, besides, includes two main improvements in the cost map w.r.t. similar algorithms. On the one hand, DyMu can handle different locomotion modes. As explained in Chapter 2.2, there are rovers capable of performing different locomotion modes such as wheel-walking (Azkarate et al., 2015), which are useful to traverse, for instance, loose soils. In this regard, DyMu implements a cost function to consider different locomotion modes, being able to divide subsequent paths into different sections depending on which locomotion mode is the most suitable to be used. On the other hand, DyMu includes two layers for different map resolutions, i.e. the global layer and the local layer. The global plan is generated on the global layer using orbital/aerial imagery. This global plan is repaired if necessary during the traverse using the local layer, after gathering information about the rover's surroundings with the rover's onboard sensors.

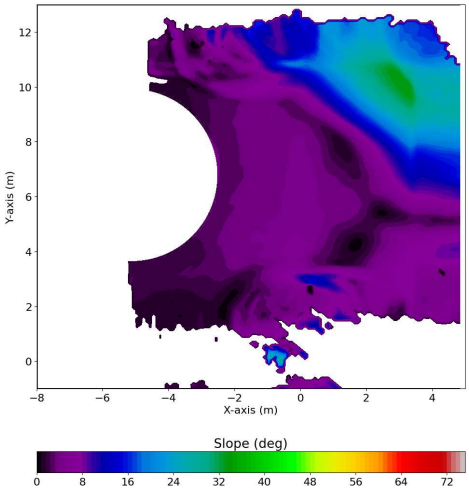
As aforementioned, DyMu uses FMM as the path planning method. FMM expands a wave on the cost map, as explained above, being this wave expansion severely affected by the cost distribution. Additionally, FMM is also the method used by DyMu for local path repairing, by expanding a second wave on the local



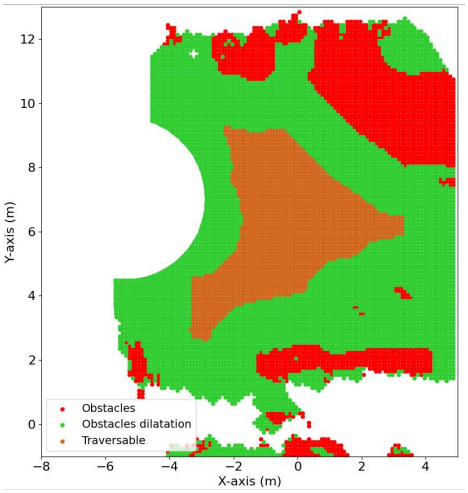
(a) DEM.



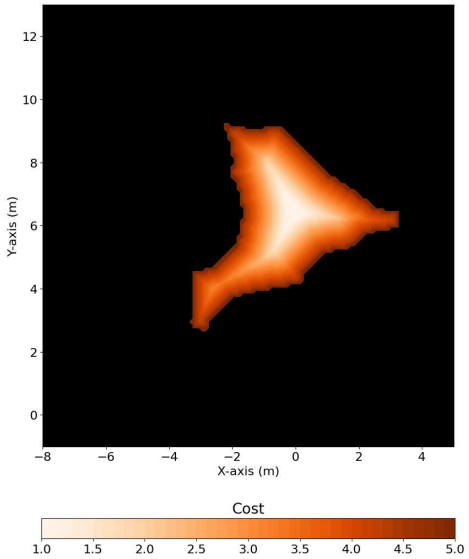
(b) Spherical deviation map, which represents the roughness of the terrain.



(c) Slope map.



(d) Traversability map $\mathcal{T}$.



(e) Cost map Ω.

FIGURE 3.2: Example of the required steps to generate a cost map from a DEM.

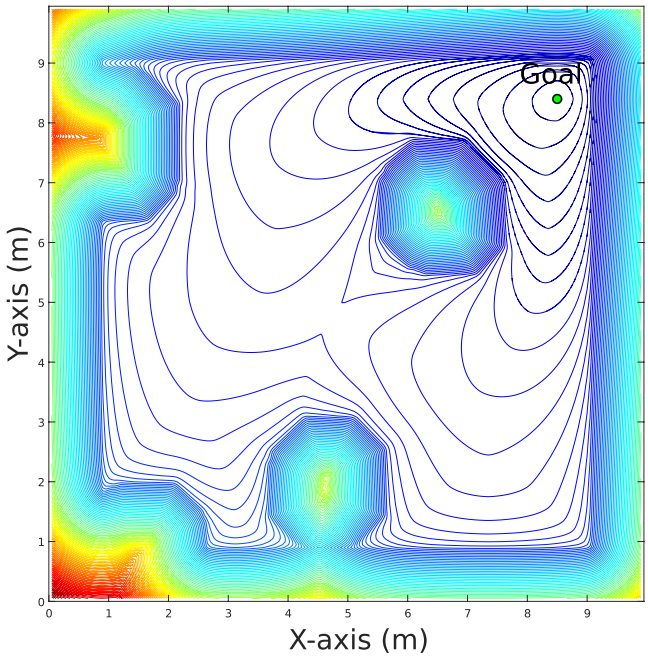


FIGURE 3.3: Example of a FMM wave expansion from the goal (a.k.a. cost to go Y), in a cost map with several obstacles.

layer of the cost map. Hence, let us provide more details about the FMM-based path planner and the local replanning methodology below.

Path planning with FMM

According to the FMM optimization problem defined in Section 3.2, the FMM path planning algorithm is divided into two main steps. First, the expansion of the wave on the cost map of the scenario. Second, the extraction of the trajectory, making use of the already expanded wave.

The first step, the wave expansion, aims at generating the cost to go map Y that defines how difficult is to reach the goal $\tilde{p}_g$ from any node $\tilde{p}_j$ in the scenario, depending on the cost map Ω. Starting from the goal, its four neighboring nodes are checked, and each one gets assigned an accumulated cost according to the corresponding costs in the cost map. Later on, the process is repeated prioritizing the nodes that have the lower accumulated cost, until the wave covers the whole map. For instance, in the simplest possible, unitary cost map (all nodes have a cost of one), the expanded wave would result in the Manhattan distance from the goal. Conversely, expanding the wave on a real cost map results in a more complex cost to go Y, as shown in Figure 3.3, where isopotential lines indicate the group of nodes where, if starting from those nodes, the same effort would be required to reach the goal. Note that the wave expansion on Figure 3.3 contains several obstacles including the map borders, since it is risky for the vehicle to get close to unknown areas.

Once the wave has been expanded, the second step is to extract the trajectory Γ. To do so the Gradient Descent Method (GDM) is used, i.e. the optimal trajectory

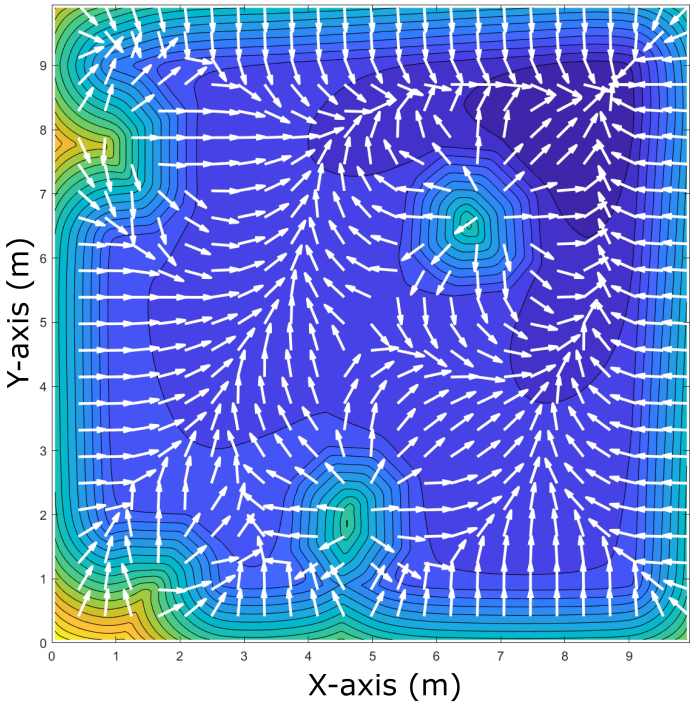


FIGURE 3.4: Gradient descent of the FMM wave (a.k.a. cost to go Y), indicating the direction to be followed to reach the goal.

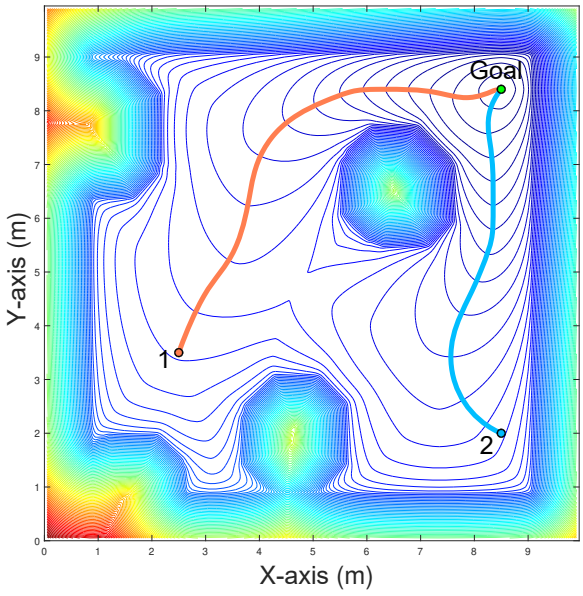


FIGURE 3.5: Planned trajectories to reach the goal from two different origins 1 and 2.

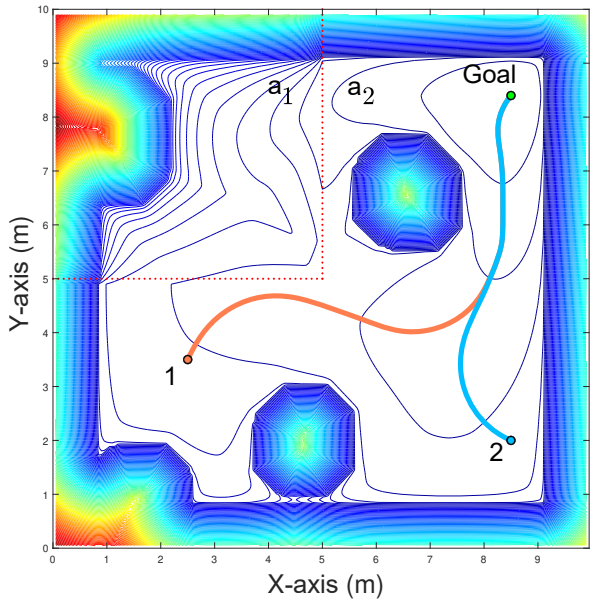


FIGURE 3.6: Planned trajectories to reach the goal from two different origins 1 and 2, considering two different terrains a_1 and a_2 in the cost map, being a_1 five times harder to traverse than a_2.

to reach the goal from any node in the map comes defined by the gradient descent direction of the expanded wave Y. This statement is clearly demonstrated in Figure 3.4, where each white arrow shows the descent direction of the cost to go Y at each node $\tilde{p}_j$. As can be observed, the arrows always point to the next isopotential line so, starting from any node in the map and following the arrows, one always finishes at the goal, and the covered trajectory is the optimal one to do so. Two examples of extracted trajectories starting from different origins A and B are shown in Figure 3.5. These trajectories are continuous, smooth, and optimal, as aforementioned.

Additionally, it is interesting to remark on the behavior of FMM when the cost map has a segmentation with different types of terrains. Let us say that, due to the soil properties and presence of obstacles, a rover traverses at 0.1 m/s average speed on a particular area of the terrain named a_1, and at 0.5 m/s average speed on another one called a_2. Then, it is sound that, in the cost map, the cost to traverse a_1 should be five times higher than the cost of a_2. As a result, planned trajectories on this cost map will try to avoid a_1. This can be clearly seen in Figure 3.6, which shows the planned trajectories in the same situation as Figure 3.5 but including the two areas a_1 and a_2. As can be observed, the trajectory from A now avoids getting into a_1, thus taking the long way around the obstacle to get to the goal. Note that this behavior is due to the relative proportion between the costs of a_1 and a_2, not because of their absolute values, i.e. the extracted trajectories would be the same if the costs of a_1 and a_2 are 5 and 1 respectively, or 50 and 10. This relative-cost behavior of FMM is key to understanding how to handle a segmented terrain to generate the cost map, or how to update the costs of the different terrains, as will be explained later.

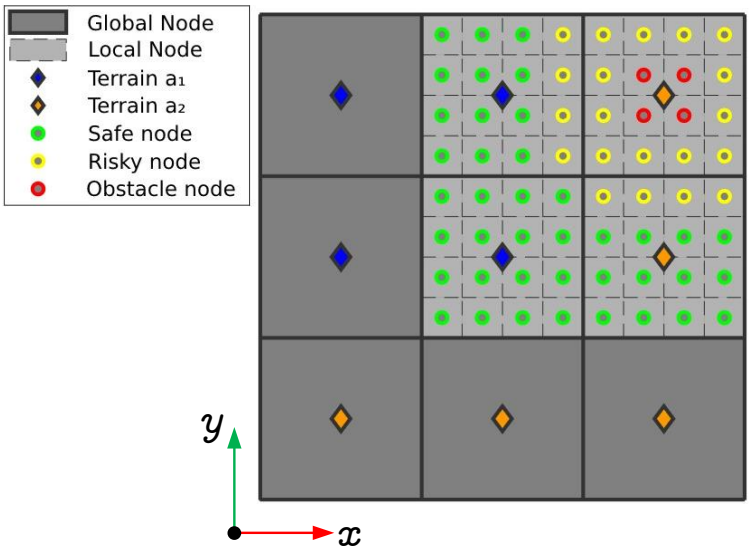


FIGURE 3.7: Depiction of the global and local layers in the proximity of an obstacle (Sánchez-Ibáñez et al., 2019b).

Local replanning for hazard avoidance

The replanning procedure proposed in (Sánchez-Ibáñez et al., 2019b) is based on the use of two different map layers with different resolutions, as aforementioned. This two-layered approach is depicted in Figure 3.7, where the global nodes, with bigger sizes, are subdivided into several local nodes. The global nodes provide information about the type of terrain they belong to, or if they are completely occupied by obstacles. Initially, these nodes, i.e. the global layer, are enough to generate a valid global plan. Nevertheless, new small-sized obstacles are detected during the traverse, which need to be handled by the local layer, for two main reasons. On the one hand, to precisely represent the area occupied by the new obstacle (red), as well as the risk area surrounding it (yellow), which indicates the area where it is dangerous to place the vehicle due to the risk of collisions. On the other hand, a precise representation of the obstacle in the local layer permits the repairing of the local trajectory with greater detail, avoiding the obstacle with the shortest path, i.e. in a more efficient manner.

After detecting an obstacle during the traverse, the repairing (or replanning) process takes place in four main steps. First, the global nodes close to the newly detected obstacles are subdivided into local nodes. Second, every local node covered by the obstacle is characterized as non-traversable. Third, the risk area is expanded, modifying the costs of the local nodes located closer to the obstacle. Fourth, a heuristic FMM is performed on the local layer to replan the trajectory.

This last step, the FMM trajectory replanning, is also the most flexible since two different approaches for trajectory planning are available: Conservative and Sweeping. As depicted in Figure 3.8, Conservative repairing (green) consists of generating a path that reconnects to the original one as soon as possible, i.e. repairing the blocked trajectory to just avoid the obstacle. Conversely, the second approach, i.e.

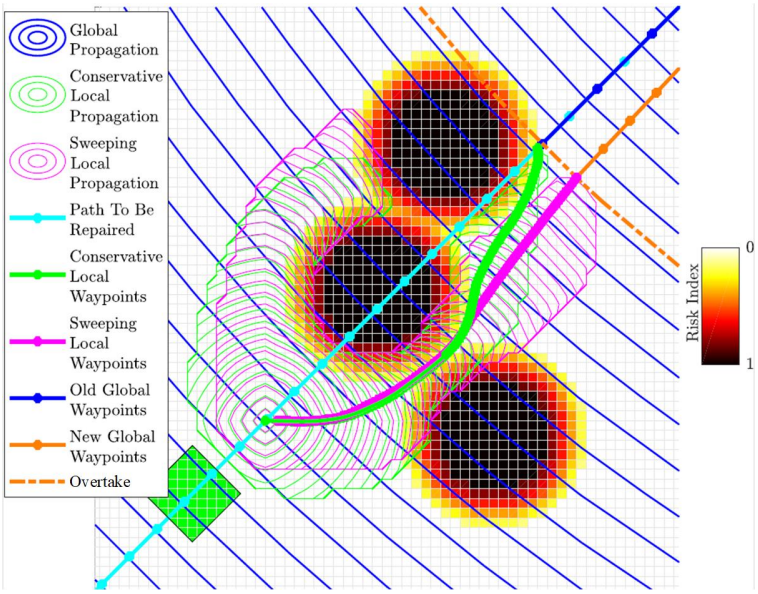
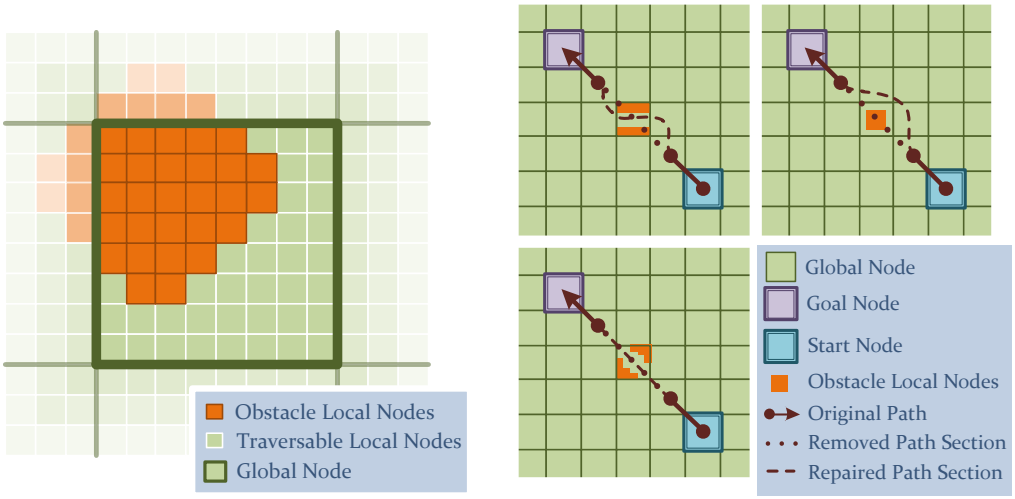


FIGURE 3.8: Repaired trajectories in the local layer according to two different methods, *Conservative* and *Sweeping* (Sánchez-Ibáñez et al., 2019b).

Sweeping replanning (pink), generates a completely new trajectory from the current position of the vehicle until the goal, but considering the newly found obstacle. Both approaches have their benefits and drawbacks, check (Sánchez-Ibáñez et al., 2019b) for more details. However, in this work, Conservative repairing has been used, given that a repaired trajectory provides direct information about how much effort the vehicle required to surpass the obstacle, which is not as straightforward when using the Sweeping method. The magnitude of the deviation, as will be explained later, is a variable that determines how difficult is to traverse the area surrounding the new obstacle, which gives information regarding how to update the cost of the neighboring global nodes.

3.4 Obstacles cost update

Usually, the cost map is discretized into nodes, each one with a scalar cost. The first update procedure lies in increasing the cost of single nodes based on the obstacles found within them. In this way, the objective is to drive the rover away from these nodes in further planning executions. Nevertheless, it is very important to determine how this increase is achieved to avoid being either too conservative or risky. To solve this issue, we essentially subdivide the area covered by any node $\tilde{p}_j^G$ (here referred to as a global node) into a number ε_j^T of smaller regular portions named local nodes. The planner hereby represents the area near the rover on a layer made up of these local nodes, which is hence called the local layer. Therefore, by using a higher resolution, the planner produces more precise paths, while also representing obstacles more accurately. The latter is convenient for computing the fraction of the global node area covered by these obstacles, as illustrated in Figure 3.9a. We refer to



(a) Graphical example of a global node where the hazard density is $\eta = 28/64 = 0.4375$

(b) Three examples of obstacle arrangement affecting the repaired path.

FIGURE 3.9: Illustration of the key concepts behind the cost update parameters: hazard density (a) and trafficability (b).

such fractions as hazard density η, which is defined in equation (3.5), where ε_j^o is the number of local nodes that are occupied by obstacles within the global node $\tilde{p}_j^G$.

$$\eta_j = \frac{\varepsilon_j^o}{\varepsilon_j^T} \in [0,1] \tag{3.5}$$

This parameter seems at first glance useful to determine the increase of the global node cost, but there is still important information missing regarding the obstacles. It refers to the way the obstacles are overcome, as seen in Figure 3.9b, and how this fact would be taken into consideration to improve the global node cost. During the repairing process (Sánchez-Ibáñez et al., 2019b), the local planner generates a path to avoid the obstacles within a global node using the local nodes. So, the traverse distance by the repaired path (d_r) can be compared to the original traverse distance without obstacles (d_o) to determine the difficulty of overcoming the obstacles within the global node. It has been defined as trafficability ξ (3.6). This parameter measures the deviation produced by the repairing process by a ratio between the original and obstacle avoidance paths' length.

$$\xi_j = \frac{d_o}{d_r} \in [0,1] \tag{3.6}$$

With these two parameters in mind, the cost function for a global node $\tilde{p}_j^G$ is formulated as stated in 3.7.

$$c_j^G = \bar{c}_j^G(2 + \eta - \xi) \tag{3.7}$$

Where $\bar{c}_j^G$ is the original cost of the global node $\tilde{p}_j^G$, and c_j^G is its updated cost after detecting an obstacle. As can be deduced, the values of η and ξ of all traversable

nodes are initialized to zero and one respectively. While the rover drives and encounters obstacles, these values are dynamically updated as a consequence. In the implementation, we have decided to increase the value of η not only on the respective node but also on its Moore neighbors. This is done to avoid introducing sharp cost gradients that, as a consequence, may let abrupt turns appear in the path. Besides, in case the value of η exceeds a threshold, the node is considered an obstacle, and therefore it cannot be trespassed by any path in further planning executions. Furthermore, the value of ξ is set to all those nodes containing waypoints that were substituted by others computed on the local layer.

3.5 Terrain features cost update

The performance of a vehicle when traveling on a certain segmented terrain is determined by multiple factors, such as terrain parameters like slip, sinkage, or obstacle density. These parameters, called terrain features in the following, define the cost map that will be used in the path planner but are difficult to estimate initially. Nevertheless, these values will be more accurate during the traverse, having information provided by the onboard sensors of the vehicle. For this reason, this section describes a dynamic segmented terrain cost updating during the traverse, which takes advantage of the measured terrain features.

The proposed method is based on an initial terrain segmentation with an estimated cost, which is out of the scope of this work. Remark that the cost of each area of terrain is considered reliable in the sense that it is proportional to the expected performance of the rover and there are no high deviations in the estimated cost along an area of terrain. For instance, there are two terrains with different initial costs, if the rover finds the real cost of any terrain is higher than the initial estimation, the other terrain should increase its cost proportionally if it has not been traversed by the rover. Keeping this proportionality in mind, we assume there is a similar error in the initial cost of both terrains, but we still rely on the qualitative aspect of the initial cost estimation, i.e., which terrains are easier or harder to traverse. Once the features of the second terrain have been estimated, this proportionality can be updated and, hence, the cost map. To guarantee this statement, the cost ratios method has been defined.

The objective of the method is to obtain the cost ratios ($\hat{c}$) between the classified terrains, as a function of the estimated terrain features. The gathered information on the traversed terrains is compared between them to find the cost ratios. The non-traversed terrains do not provide any feedback information, thus their cost will be calculated in attempting to maintain a proportionality w.r.t. the already traversed terrains. Once the cost ratios are computed, new absolute costs are defined for each area of terrain.

A set of M terrain areas has been defined as $\mathbb{A} = \{a_1, a_2, ..., a_m, ..., a_M\}$. Each one has a set of N_f terrain features defined as $\mathbb{F} = \{f_1, f_2, ..., f_n, ..., f_{N_f}\}$, and the corresponding set of the estimated costs for each terrain area is $\mathbb{C} = \{c_1, c_2, ..., c_m, ..., c_M\}$. This cost represents the initial values of the global nodes that are within each area of terrain. Once the rover performs a traverse, it crosses a subset of terrain areas $\mathbb{A}_U \subseteq \mathbb{A}$ whose costs are updated and stored in a subset $\mathbb{C}_U \subseteq \mathbb{C}$. Each updated cost c_m is calculated using a parameter ς_m, called terrain hardness, which is calculated as in (3.8), where $\overline{f_{m,n}}$ represents the estimated value of the feature $f_n \in \mathbb{F}$ in the terrain $a_m \in \mathbb{A}_U$, and $\mathbb{W} = \{\varrho_1, \varrho_2, ..., \varrho_n, ..., \varrho_{N_f}\}$ are weights of each terrain feature.

$$\varsigma_m = \frac{\sum_{n=1}^{N_f}(\varrho_n \cdot \overline{f_{m,n}})}{\sum_{n=1}^{N_f}\varrho_n} \tag{3.8}$$

The proportion between the terrain hardness of every traversed terrain is obtained through the cost ratio $\hat{c}_m$, which is defined in (3.9), where ς_0 represents the terrain hardness of the first traversed terrain that belongs to $\mathbb{A}_U$. Obviously, the cost ratio of the first traversed terrain is 1.

$$\hat{c}_m = \frac{\varsigma_m}{\varsigma_0}; \qquad \forall m | a_m \in \mathbb{A}_U \tag{3.9}$$

Finally, considering that the non-traversed terrains should maintain the same cost ratio with regards to the lowest estimated cost of the traversed terrains, aiming at maintaining the same probability of being traversed, the final updated costs of the traversed terrains $\mathbb{C}_U$, are obtained as expressed in (3.10), (3.11) and (3.12), where $\hat{c}_{min}$ and c_{min} are the minimum cost ratio and the initial absolute cost from the traversed terrains. As a result, the terrain a_m with the minimum cost ratio $\hat{c}_m = \hat{c}_{min}$ will preserve the previous lowest cost of the traversed terrains c_{min}, and the rest will be proportional to the obtained cost ratios.

$$c_m = c_{min} \cdot \frac{\hat{c}_m}{\hat{c}_{min}}; \qquad \forall c_m \in \mathbb{C}_U \tag{3.10}$$

$$\hat{c}_{min} = \hat{c}_m | \hat{c}_m \leq \hat{c}_j \qquad \forall a_m, a_j \in \mathbb{A}_U \tag{3.11}$$

$$c_{min} = c_m | c_m \leq c_j \qquad \forall a_m, a_j \in \mathbb{A}_U \tag{3.12}$$

In this work, two terrain features have been used to update the cost. The first one is the slip ratio (σ), which is defined as a proportion between the surface speed of the wheel compared to the ground-truth speed of the vehicle, as expressed in (3.13), where v_w is the surface speed of the wheel and v_r is the ground-truth speed of the rover. The former is easily measured according to the rotational speed and the radius of the wheels; the latter could be estimated by visual odometry or a localization procedure.

$$\sigma = \frac{v_w - v_r}{v_w} \tag{3.13}$$

To avoid divisions by zero, as needed in (3.9), the feedback data related to slip ratio ($\overline{f_{m,1}}$) is obtained as in (3.14), where $\overline{\sigma_m}$ is the mean of all the gathered samples of the slip ratio in the terrain a_m.

$$\overline{f_{m,1}} = \frac{1}{1 - \overline{\sigma_m}}; \qquad \forall a_m \in \mathbb{A}_U \tag{3.14}$$

The second terrain feature used is the already mentioned trafficability (ξ), described in the previous section. To ensure that higher values of the feature mean worse performance of the vehicle, the feedback data related to trafficability ($\overline{f_{m,2}}$) is computed as shown in (3.15), where $\overline{\xi_m}$ is the mean of all the gathered samples of trafficability in the terrain a_m.

$$\overline{f_{m,2}} = \frac{1}{\overline{\xi_m}}; \qquad \forall a_m \in \mathbb{A}_U \tag{3.15}$$

As will be later explained in Chapter 6.2, the rover continuously takes samples of σ and ξ, which the planner saves to compute the average value before updating the costs. However, not all the samples are valid due to external disturbances in the measurements. To take this into account, a simple statistical procedure has been included, which decides whether to accept or reject the obtained samples, even switching the whole set of samples if a significant deviation is detected. This method is based on the standard deviation of the mean, with Student T-tests.

3.6 Summary and conclusions

In this chapter, a new dynamic cost map update approach is presented. As the first contribution of this thesis, this update procedure is integrated into the DyMu path planner, which is used as a building block for most of the work performed during this doctoral research as is shown throughout this dissertation.

The presented feedback update methodology improves global path planner performance when approaching an area of scientific interest (where to perform a MM task) in planetary exploration missions. This approach particularly benefits round-trip missions, but it is also advantageous to increase the rover awareness of the scenario in any use case where the global information is not sufficiently accurate. This approach is divided into two update methods. On the one hand, the individual node cost is increased considering the nearby recently detected obstacles. On the other hand, the costs of the terrain areas are updated using the gathered information during the traverse regarding the soil properties or the presence of obstacles. These costs are extrapolated to terrain areas with similar features, making use of a previous terrain classification of the scenario. Finally, the proposed algorithms have been validated by means of simulation and field tests, as explained later in Chapter 6.2.

Note that in this work only two features for the terrain areas are considered, the slip ratio and the trafficability. Nevertheless, adding a greater variety of soil parameters would substantially improve the rover's awareness of the terrains and, thus, the efficiency of the subsequent planned paths. Additionally, the isotropic costs considered in this work have limitations to represent overturn risks or slip ratios on big slopes w.r.t. the rover heading, these would be solved by including the cost update methodology in an anisotropic planning algorithm, such as the one presented in (Sánchez-Ibáñez et al., 2023). The aforementioned issues are proposed as future work.

After improving the quality of the planned paths with this first contribution, the efficiency of the global guidance to reach the area of scientific interest is increased. Anyhow, it is still required to include in the loop a motion planner for the robotic arm, so both systems (mobile base and manipulator) move coordinately to perform the scientific MM operations within the area of interest. The following chapters of this thesis present, thus, two different MM motion planners, which are the second and third contributions of this doctoral thesis.

Chapter 4

Robust motion planning for mobile manipulation

"Nemo liber est, qui dominus sui non est" -
"No man is free who is not master of himself"

Epictetus
1st century A.D.

4.1 Introduction

Though autonomous navigation technologies have reached impressive heights on the latest Martian rovers, increasing the autonomy level is yet difficult once the rover has to perform a MM task in the area of scientific interest. In particular, the motion planning problem, Guidance, is hindered by the existence of infinite solutions due to the high kinematic redundancy of these systems, as extensively explained in Chapter 2.3. Besides, planning the motion for MM in planetary exploration presents further challenges, such as high motion efficiency, low computational effort, predictability, and robustness. The latter is particularly critical to ensure compliance with mission-critical constraints, like obstacle and self-collision avoidance. Given the current state of the art, sub-optimal methods, such as a combined path and motion planner like OHMP (Li et al., 2020), arise as a good approach to tackle the motion planning problem for MM in planetary exploration, with low computational cost and high motion efficiency. This type of sub-optimal coupled path and motion planner is usually divided into two main steps. First, the rover base path is generated, trying to reach the goal in the shortest manner while avoiding obstacles in the scenario. Second, the arm motion is computed w.r.t. the already known base path, ensuring that the arm does not collide with the environment or the rover itself (what is called a self-collision).

Note that the fact that the manipulator motion is based on a previously computed rover path provides several advantages. First, if the planned rover path uses

globally optimal approaches like FMM (Sethian, 1999), which has already been extensively presented in this thesis, then the complete motion plan is partially optimal, and the efficiency is increased. Second, the arm motion range is enclosed in the surrounding volume of the rover path, thus the complexity of the problem is substantially reduced, and so is the computational cost of the algorithm. Third, if the path planner finds a solution to place the rover close to the goal considering the forthcoming retrieving operation, then it is guaranteed that the process is feasible, i.e. a safe motion plan for the manipulator will be found, improving the predictability and robustness of the algorithm. Nevertheless, further research is required to bring this type of sub-optimal MM motion planner to planetary rovers, in two main senses.

On the one hand, regarding the rover base path, the stage of approaching the goal has to guarantee that the rover will be located in the correct orientation to place the goal within the manipulator workspace. In consequence, planning and controlling the final orientation of the rover base is essential to avoid unnecessary maneuvers, which induce wasting energy and time. An algorithm that could solve this problem was presented in (Takei and Tsai, 2013), where a globally optimal path planner that considers curvature and orientation was proposed. However, this algorithm is based on the FSM (Zhao, 2004) which is only recommended for simple path planning scenarios with few obstacles (Gomez et al., 2019). In other cases, similar advanced path planning methods are computationally cheaper, such as FMM. In fact, for a properly defined cost map representing a particular scenario, FMM always returns a safe (avoiding obstacles or non-traversable areas) and globally optimal (with maximized time-energy efficiency) path to reach the goal, if existent.

On the other hand, the manipulator motion planner has to rigorously consider the collisions with the environment and the self-collisions, which is particularly challenging bearing in mind that planetary rovers generally have a highly restricted arm workspace, due to the presence of multiple onboard scientific and engineering tools. Online self-collision avoidance is a considerably expensive task in computational terms, meanwhile, an offline computation of the manipulator workspace can notably reduce the computational effort to avoid self-collisions.

Hence, this chapter presents the second contribution of this thesis: an efficient and robust motion planner that combines path and motion planning for planetary exploration MM, considering a highly restricted arm workspace. The motion planner proposed in (Sánchez-Ibáñez et al., 2019a) is extended, resulting in the following enhancements:

- An improvement in the DyMu path planner (Sánchez-Ibáñez et al., 2019b), also used within the first contribution of this thesis, to ensure the base trajectory places the rover with the goal sample inside the manipulator workspace, by planning and controlling the final rover orientation during the stage of approaching the goal.

- A novel motion planning algorithm for the manipulator that considers offline

the real workspace of the arm to avoid collisions, hence directly obtaining safe plans to reach the goal sample location. Besides, the planner increases the efficiency of the motion plan thanks to a coordinated base-arm motion. To this end, the motion planner generates the end effector path using FMM 3D, considering the offline computed collision-free workspace of the manipulator w.r.t. the mobile base trajectory, to later generate the arm joints motion plan.

- A custom coupled arm-base trajectory controller for the motion plan to be properly followed by the system.

The rest of this chapter is organized as follows. The first improvement, i.e. the developed path planning algorithm with control of the final rover orientation, is explained in Section 4.2. Later on, Section 4.3 details the second improvement, the motion planning algorithm for MM with a highly restricted arm workspace, and also details the coupled trajectory controller. Lastly, Section 4.4 concludes this chapter and proposes some future work.

4.2 Path planning with final approach control

The proposed motion planner for MM consists of two main stages: path planning for the mobile base and motion planning for the manipulator. The path planning stage uses the DyMu planner (Sánchez-Ibáñez et al., 2019b), making the most of the improved methodology that states the first contribution of this thesis (see Chapter 3). In this regard, a DEM provides the height information necessary to compute a traversability map $\mathcal{T}$, which is used to generate a cost map Ω_B. Note that the 2D cost map for the mobile base is denominated Ω_B in the following, instead of Ω. This is to differentiate it from the 3D cost map used to plan the manipulator movements, called Ω_M, as explained later.

Let us summarize the path planning procedure. First, the traversability map $\mathcal{T}$ is extracted from the DEM, to differentiate safe and unsafe areas using different parameters such as roughness and slope. Second, the cost map Ω_B is generated in accordance with the traversability $\mathcal{T}$. The cost map Ω_B assigns high costs to the nodes labeled as obstacles or non-traversable in $\mathcal{T}$, and low costs to the safe ones. Additionally, to increase the smoothness of future paths, cost gradients are generated in the non-traversable nodes' proximity, as defined in (4.1), with $\tilde{p}_j \in \mathcal{T}$ a node in the map, $\dot{c}_B$ a configurable parameter defining the cost sharpness, and $\mathcal{D}_B(\tilde{p}_j)$ the Euclidean distance from the node $\tilde{p}_j$ to the closest non-traversable node in $\mathcal{T}$. Remark that $\mathcal{D}_B$ is calculated using standard computer vision algorithms within Open Source Computer Vision Library (OpenCV)[1].

$$\Omega_B(\tilde{p}_j) = 1 + \dot{c}_B / \mathcal{D}_B(\tilde{p}_j) \quad \forall \tilde{p}_j \in \mathcal{T} \tag{4.1}$$

[1] `https://opencv.org/`

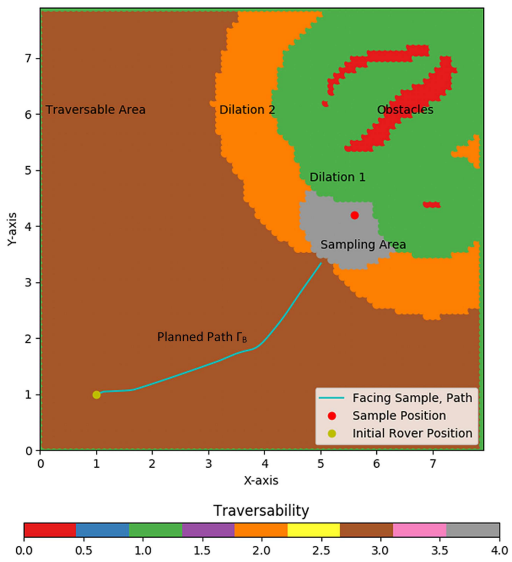
In summary, once the cost map Ω_B is generated, FMM computes the optimal path to reach the goal node $\tilde{p}_g$ starting from $\tilde{p}_0$, i.e. Γ_B, which is composed of N_B waypoints $\Gamma_B(n)$. FMM models the propagation rate of a wave by numerically solving the Eikonal equation, i.e. a non-linear partial derivative equation. This wave is called the cost to go Υ_B. Given, Υ_B, FMM extracts Γ_B by following the descent direction of the gradient of Υ_B, what is called GDM. Check Chapter 3.2 for a detailed description of FMM, GDM, and the path planning process.

Although functional in most use cases, this path planning approach is not sufficient for a rover that has to approach a sample for a MM operation, since the path planner has to place the rover close enough to the sample and with proper final orientation, to ensure that the sample is finally inside the manipulator workspace. This is further hindered in the case of a highly restricted manipulator workspace since only some particular rover poses leave the sample inside that workspace. For instance, if there is not enough space in front of the rover for manipulation tasks, the planner has to leave the rover with a certain angle w.r.t. the goal or even turned 90° to perform the manipulation on one of the rover sides.

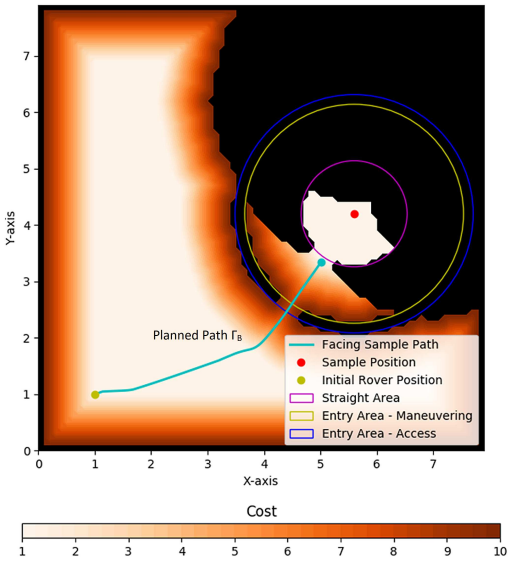
Given these requisites, two aspects need to be considered to ensure the goal is finally inside the arm workspace. On the one hand, the last segment of the path has to be a straight line towards the sample, thus restraining the rover orientation in those last waypoints. This is necessary because, in the vicinity of obstacles, approaching the goal in a straight line benefits the existence of a safe maneuver to place the sample inside the manipulator workspace. On the contrary, a curved approach to a target near an obstacle is prone to leaving the rover's wheels too close to the obstacle, which may impede moving the platform in a manner that leaves the target inside the manipulator workspace. Since the path planner relies on FMM, which produces a continuous and optimal path given a cost map, the shape of the resulting path is subject to variations in the cost of the nodes. Under this premise, the Frontal Approach Cost Editing (FACE) algorithm is presented, which modifies the cost map in the close neighborhood of the sample to ensure that the rover approaches it following a straight line. On the other hand, the later trajectory controller must steer the rover precisely enough to ensure that the sample is finally inside the arm workspace, as aforementioned. Performing such an orientation-controlled but arm-compliant final maneuver requires a tailored modification of the trajectory controller, thus, the Last Section Control (LSC) is proposed. These two improvements together, FACE and the LSC, ensure the goal sample is reachable by the arm, as further explained below.

Frontal Approach Cost Editing (FACE)

The main purpose of FACE is to modify the cost map in the proximity of the sample to ensure that the last waypoints of the path are headed toward the target. This modification is exemplified in Figure 4.1, with the rover going from an initial position (yellow dot) to a position close to the sample (red dot). First, the traversability



(a) Traversability map $\mathcal{T}$. The nodes are classified into different categories of traversability as part of the terrain assessment. The obstacles are dilated to account for the rover morphology. The sampling area is defined around the sample location depending on the arm reachability.



(b) 2D cost map Ω_B. The cost is defined for each node according to its category in $\mathcal{T}$. Besides, traversable nodes located within the blue-yellow ring may or not be turned into obstacle nodes. After projecting the obstacles onto the ring from the sample position, only a small opening remains traversable in this particular case.

FIGURE 4.1: Showcase of an example in which FACE is used to ensure the planned path Γ_B goes straight towards the sample in its final waypoints.

map is generated, shown in Figure 4.1a. The planner expands the area of the obstacles to consider the shape of the rover, as aforementioned, although FACE includes two different dilatation areas. The nodes within the first dilatation (green) are not reachable by either the center of the rover or the manipulator. The second dilatation (orange) indicates nodes where the rover center cannot enter, but the manipulator can reach. Note that, even though the rover does not stop on top of the goal, FMM initially plans the path connecting the initial and the goal positions. This is to, in a later step, remove the last portion of the path so the rover stops earlier at a certain distance from the sample, under the maximum arm reachability, what is called the sampling area (grey).

The traversability map is used to generate the cost map as in Figure 4.1b. Here, the values of cost, positive and non-zero, are proportional to the proximity to non-traversable nodes as in (4.1). Those non-traversable nodes are colored in black and their cost is infinite. The 2D path that FMM gets between two points is affected by how these values of cost are assigned. An area with constant cost will make the paths go straight, while a gradient of the cost will curve the paths. Therefore, the sampling area should be set as a uniform cost to generate straight paths to approach the sample. However, there may be obstacles in the vicinity, and they must not be omitted.

FACE is aware of this necessity to make the paths go straight meanwhile avoiding existing obstacles, and edits the cost ensuring the resulting path complies with

those requirements. As can be seen in Figure 4.1b, the non-traversable nodes not only include obstacle and dilatation nodes, but also a series of nodes that are located within a ring centered at the sample location. This ring is the one formed by the blue and yellow circumferences, whose radius values are configurable by the user. FACE determines which nodes from this ring are set to obstacle and which are not, forming an opening so the path goes through it, straight towards the sample.

For that purpose, FACE uses an application of FMM called shadowing, which was originally presented in the work of Sethian (Sethian, 1999). In a few words, given a point and obstacles around it, the shadowing serves to project shadows as if the point in question were a light source. In this case, such a point is the sample location. The projected shadows pass through the ring, converting traversable nodes into non-traversable ones. Only those traversable nodes inside the ring that are not touched by the projected shadows form the opening. This means any path passing through such nodes will later go straight toward the sample location as if they were following the light rays. Finally, the resulting path from using FMM is shortened by removing the last waypoints as aforementioned, as the rover is supposed to stop at a certain distance from the sample. In Figure 4.1b the last waypoints entering the straight area (which contains the sampling area defined in the traversability map) are the ones that are removed.

Last Section Control (LSC)

The trajectory controller ensures the robot follows the planned path under certain conditions. For instance, the conservative pure-pursuit algorithm ensures the rover does not deviate from the path beyond a certain distance (Filip, Azkarate, and Visentin, 2017). This can be seen as forming a corridor that delimits the area where the rover can be located. Inside this corridor, the conservative pure-pursuit provides the rover with motion commands depending on the waypoints it tracks. Nevertheless, as aforementioned, the rover must orient itself to make the sample reachable by the arm at the last waypoints of the path. This adds a new problem that the trajectory controller must account for. Not only the rover has to drive performing maneuvers that reduce its deviation w.r.t. the planned path, but it also has to reach this heading at the end. A feasible solution would be to decouple the problem. First, the rover is focused on reaching the last waypoint. Second, it performs a turn on the spot, or point-turn, to set its heading to the desired one. However, this solution is not efficient in time terms. To reduce the number of maneuvers at the end, a custom control approach named LSC is presented, which takes over the previous trajectory controller when the rover is close enough to the sample.

Figure 4.2 shows a scheme of how the LSC works. As can be checked, a single full-Ackermann maneuver cannot ensure the rover is well positioned w.r.t. the sample according to where its arm is placed (colored in purple). Reaching the last waypoint of the path with a certain heading requires starting from a unique position (colored in orange). It is then imperative to control the rover to not only make it stop

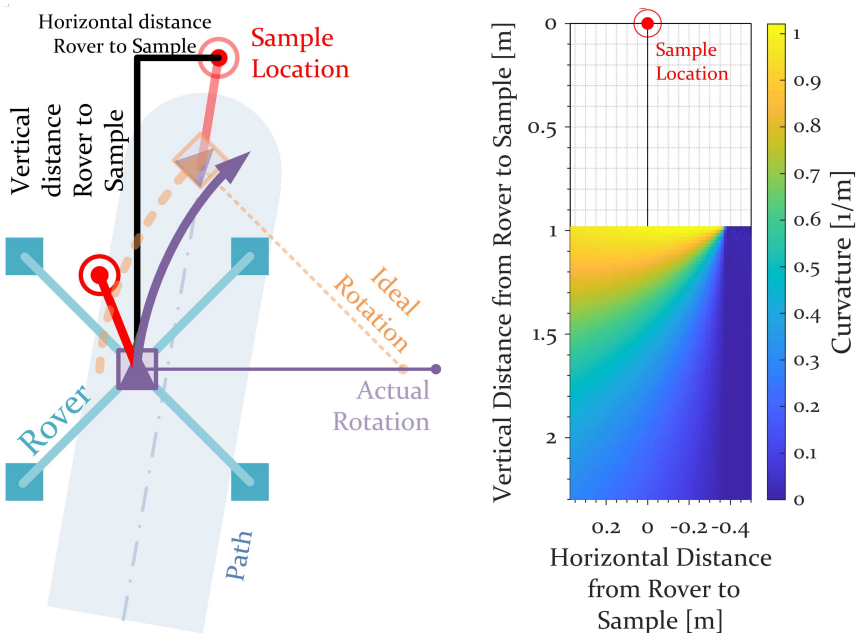


FIGURE 4.2: LSC scheme. On the left, the scheme depicts how the rover cannot arrive at the final waypoint and orientation using a single turning radius for a constant full-Ackermann maneuver. On the right, the vertical and horizontal distances between the rover and the sample serve as input to the LSC to get the curvature the rover should use to turn.

at a position close to the final waypoint (admitting a certain level of error) but also reduce the distance between the optimal pick-up position (in red) and the location of the sample. Having all this in mind, the LSC takes as inputs only the vertical and horizontal distances from the rover's central reference frame to the sample location. According to these values, the LSC uses a look-up table, plotted at the right side of Figure 4.2, to dynamically retrieve the curvature of the turning full-Ackermann maneuver the rover must perform. As the rover advances this look-up table is consulted iteratively, providing a different value of curvature depending on how the vertical and horizontal distances evolve. Following the example of Figure 4.2, if the horizontal distance is between -0.3 to $-0.5\,\text{m}$, the rover can practically just go straight, as the optimal pick-up position is more or less in line with the sample with none or little horizontal distance error. For values of horizontal distance higher than $-0.3\,\text{m}$, the rover has to make a right turn as it advances, which is more pronounced as it comes closer to the sample.

4.3 Motion planning with restricted arm workspace

Once the rover base path is computed, it is used by the MM motion planner to generate the arm joints profile of the arm throughout the trajectory, generating a coordinated base-arm motion. The motion planner tries to ensure the operation safety, i.e. avoiding collisions, efficiently. To do so, a representation of the safe manipulator

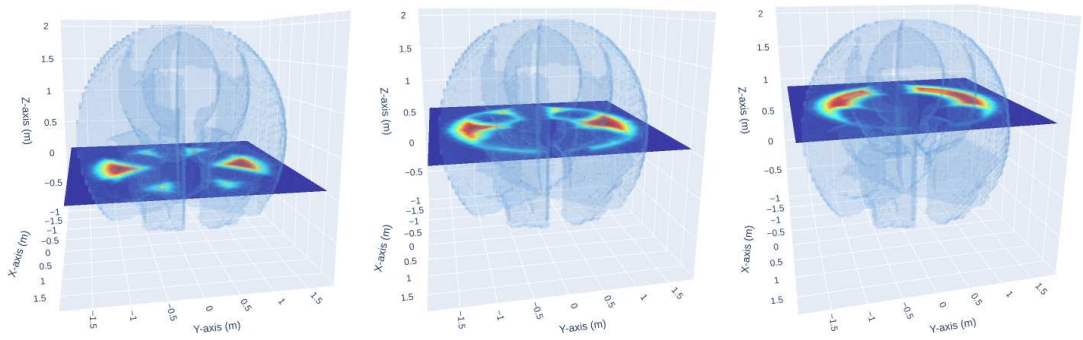
workspace in the form of a reachability volume is generated offline, only once, before planning the manipulator movements. The reachability volume is obtained by employing the collision detector algorithm of the Dynamic Animation and Robotics Toolkit (DART) (Lee et al., 2018), to detect self-collisions and collisions with the environment for any specified configuration of the system joints. DART makes use of the Unified Robot Description Format (URDF) (Quigley et al., 2009) of the rover, mainly requiring the kinematic and collision models of the system. URDF is a robot descriptor, which assembles the kinematics, dynamics, collisions, and even visuals of a robot in a straightforward manner, describing the characteristics of the joints and links that compose the robot, and their relationships. Though many robot descriptors exist in the literature, URDF is the most renowned one and has worldwide support.

The reachability volume is used by the motion planner, together with the previously computed rover base path, to generate the arm joints profile to get to the goal, in three main steps. Firstly, a 3D cost volume is generated, formed by a tunnel surrounding the rover base trajectory. This tunnel represents the manipulator Cartesian workspace at each waypoint of the rover base trajectory, using the reachability volume but also considering the scenario (obstacles, soil shape). This way, the later computed motion plan is necessarily safe if encompassed inside this tunnel. Secondly, the manipulator trajectory is planned inside the tunnel using FMM in a 3D version. Note that we refer to the manipulator path in the following since it could be the path for the end effector or the wrist. Third, the manipulator joint profile is obtained by establishing a relation between the base and the manipulator trajectories. For each waypoint of the base path, a suitable waypoint of the manipulator trajectory is assigned, and then the corresponding arm joints configuration is computed making use of the IK model of the manipulator.

Finally, a coupled trajectory controller is required to ensure that both the base path and the manipulator joint profile are properly tracked by the system. The reachability volume computation, the motion planner itself, and the coupled trajectory controller are explained in further detail below.

Manipulator workspace representation

Let us define W as the manipulator workspace, i.e. a set of 3D discrete nodes $\tilde{p}_j$ where the manipulator end effector can be ideally placed considering its movement range. Then, $W_R \subset W$ is defined as the safe manipulator workspace considering self-collisions, i.e. a Boolean 3D map indicating for each $\tilde{p}_j$ if the arm is safe and reachable (true), or if the manipulator cannot reach the position or is in a collision state (false). Then, the concept of reachability volume Ω_R is introduced. In a similar way to the Capability Map defined in (Porges et al., 2014), Ω_R is a 3D map ($\Omega_R \in W_R$), where $\Omega_R(\tilde{p}_j)$ indicates to what extent it is dangerous to place the manipulator in $\tilde{p}_j$ w.r.t self-collisions. The main benefit of using Ω_R within the motion planner is that, if every manipulator configuration leaves the arm inside Ω_R, it is guaranteed



(a) (b) (c)

FIGURE 4.3: Example of a distance to collisions volume $\mathcal{D}_M$, where the safer nodes $\tilde{p}_j$ are represented in red, and the ones closer to collisions are shown in blue.

that no self-collision will occur during the execution. This is assuming that the arm trajectory controller does not deviate substantially from the plan but, if it does, the chances of self-collisions are also minimized, since Ω_R will maintain the arm the farthest possible from the closer self-collision. Besides, since Ω_R is calculated only once, offline, it is substantially advantageous to reduce the computational cost of the motion planner, being Ω_R unique for a robot as long as its kinematic characteristics do not change.

To compute Ω_R it is necessary, first, to estimate the safe manipulator workspace W_R, which requires several tools. On the one hand, URDF is used to 3D model the physical properties of the system, including the collision geometry, which is slightly expanded to ensure safety even in case of inaccuracies. On the other hand, the URDF is embedded in a collision detector algorithm, in this case, the one provided by DART. This is an open source library where kinematic and dynamic algorithms are developed, in the fields of robotics and computer animation. The collision detection checks whether there is a collision or not for a certain scenario and configuration of the robot joints. Lastly, the IK of the manipulator is also necessary, to correlate each position $\tilde{p}_j \in W_R$ with its corresponding arm joints configuration q. Note that, in this work, each W_R is representative of only one solution of the IK, i.e. up/down elbow and right/left shoulder.

To obtain W_R, each node $\tilde{p}_j$ inside the manipulator workspace W has to be checked. For every particular $\tilde{p}_j$ and every possible orientation $\phi = [\varphi\, \vartheta\, \psi]$ (roll, pitch, yaw) of the end effector, the IK is used to extract the arm joints configuration q to reach that end effector pose (position $\tilde{p}_j$ and orientation ϕ). Then, DART checks if q leads to a collision, employing the URDF model of the robot. Whenever a collision is found, the node $\tilde{p}_j$ is categorized as unsafe in W_R, and a new one is selected. On the contrary, if every end effector orientation ϕ is covered and no collision is detected, then $\tilde{p}_j$ is classified as safe in W_R.

Finally, Ω_R is generated by defining a cost shape inside W_R, which indicates to what extent it is hazardous to place the manipulator at each node $\tilde{p}_j$. For this purpose, the distance to collisions volume $\mathcal{D}_M$ is required. This is a 3D discrete volume

($\mathcal{D}_M \in W_R$) where $\mathcal{D}_M(\tilde{p}_j)$, indicates the distance from $\tilde{p}_j$ to the closest unsafe position (collision) in W_R, as shown in Figure 4.3. Thus, unsafe positions in W_R have assigned distance 0 in $\mathcal{D}_M$, and safe positions in W_R have assigned the Euclidean distance from $\tilde{p}_j$ to the closest unsafe one in W_R. Using $\mathcal{D}_M$ a linear cost function inside the cost volume Ω_R has been defined, which is inversely proportional to $\mathcal{D}_M(\tilde{p}_j)$ as stated in equation (4.2).

$$\Omega_R(\tilde{p}_j) = 1 + \dot{c}_M / \mathcal{D}_M(\tilde{p}_j) \quad \forall \tilde{p}_j = [x, y, z] \in W_R \tag{4.2}$$

Where $\Omega_R(\tilde{p}_j)$ is the cost associated with the node $\tilde{p}_j$; $\dot{c}_M$ is a parameter that defines the sharpness of the cost inside Ω_R, and $\mathcal{D}_M(\tilde{p}_j)$ is the distance of the position $\tilde{p}_j$ to the closest unsafe position in W_R. Remark that the minimum cost in Ω_R is 1, as needed for correct performance of the later path planning algorithm with FMM. Additionally, the cost of the non-reachable or non-safe nodes in W_R is set to infinity.

In the particular case of a manipulator with a spherical wrist, the IK model is easily obtainable with a kinematic decoupling of the position and orientation problems (Siciliano et al., 2009). Defining $q = [q^p\, q^o]$, the first DoF set the wrist position (position joints, q^p) and the last 3 DoF set the end effector orientation (orientation joints, q^o). Making the most of this kinematic decoupling, the safe workspace W_R of a manipulator with a spherical wrist represents all the positions $\tilde{p}_j$ for the wrist where the full range of the orientation joints $\mathbb{Q}^o$ is reachable and safe, i.e. without self-collisions. This methodology simplifies substantially the motion planning procedure, as will be explained in the posterior subsection.

The procedure to generate Ω_R for a manipulator with a spherical wrist is, thus, summarized in algorithm 1. First, according to the kinematic configuration of the manipulator, the wrist workspace W (*getManipulatorWorkspace*) and the maximum range $\mathbb{Q}^o$ of the orientation joints q^o (*getOrientationJointsRange*) are obtained. Second, the safe workspace of the manipulator wrist W_R is computed, checking if the manipulator is colliding (*isArmColliding*) for every node $\tilde{p}_j = [x, y, z] \in W$ and configuration of the orientation joints $q^o \in \mathbb{Q}^o$. Third, once W_R is generated, the Euclidean distance volume to the closest collision $\mathcal{D}_M$ is calculated (*computeDistanceToCollisions*), using again standard computer vision techniques from OpenCV, but in 3D. This distance volume $\mathcal{D}_M$ defines the cost function inside the reachability volume, altogether with $\dot{c}_M$, parameter that configures the cost function sharpness (*getCostSharpness*). Finally, the reachability volume Ω_R is generated, checking every node $\tilde{p}_j = [x, y, z] \in W$, and assigning the cost in (4.2) if the position is safe ($W_R(\tilde{p}_j)$ is *true*), or infinite cost if the position is not reachable ($W_R(\tilde{p}_j)$ is *false*).

Manipulator motion planning

Before generating a coupled motion, first of all, it is required to decide when to start the arm motion w.r.t. the rover base path. That is, to estimate the waypoint $\Gamma_B(\iota)$

Algorithm 1 Computation of the reachability volume, Ω_R, for a manipulator with spherical wrist

```
W ← getManipulatorWorkspace()
Q^o ← getOrientationJointsRange()
for p̃_j = [x, y, z] in W do
    W_R(p̃_j) ← true
    q^p ← getWristIK(p̃_j)
    for q^o in Q^o do
        q ← [q^p q^o]
        if isArmColliding(q) then
            W_R(p̃_j) ← false
            break
        end if
    end for
end for
D_M ← computeDistanceToCollisions(W_R)
ċ_M ← getCostSharpness()
for p̃_j = [x, y, z] in W do
    if W_R(p̃_j) is true then
        Ω_R(p̃_j) ← 1 + ċ_M / D_M(p̃_j)
    else
        Ω_R(p̃_j) ← ∞
    end if
end for
```

of the base trajectory where the arm should start moving, with ι the index of that waypoint. This is necessary to ensure that the manipulator finishes its movements just when the rover gets to the last waypoint of the trajectory $\Gamma_B(N_B)$, to avoid idle arm time if finishing the arm motion too soon or idle rover time if finishing too late. The selection of $\Gamma_B(\iota)$ depends on two main factors. On the one hand, the time required for the rover to follow the trajectory, which is estimated subject to the rover base nominal speed and the length of the trajectory Γ_B. On the other hand, the time required for the arm to perform the complete motion, which can be estimated based on the initial and final arm joints' configurations, and the arm joints' speed. Note that the final arm configuration can be computed depending on the goal pose and the last rover base waypoint $\Gamma_B(N_B)$.

Then, a 3D cost volume Ω_M is generated on top of the rover base trajectory, following algorithm 2. First, the rover base trajectory Γ_B (*getRoverBaseTrajectory*) and the trajectory index where to start the arm movements ι (*getIndexStartArmMovement*) are fetched. Second, the cost volume Ω_M is initialized (*initialize3DCostVolume*) considering every node as an obstacle, with infinite cost. Third, the deployment distance d is defined (*getDeploymentDistance*). This is a configurable parameter that sets the fore distance (X-axis) from the arm base to the manipulator end effector (or wrist) at each waypoint, eventually characterizing when the arm will be mainly mobilized during the rover movement, as is later explained in detail. The deployment distance, clearly, always has to be smaller than the maximum range of the manipulator.

Algorithm 2 Computation of the 3D tunnel cost volume Ω_M

$\Gamma_B \leftarrow$ getRoverBaseTrajectory()
$\iota \leftarrow$ getIndexStartArmMovement()
$\Omega_M \leftarrow$ initialize3DCostVolume()
$d \leftarrow$ getDeploymentDistance()
$\Pi \leftarrow$ getReachabilityVolumeSlice(X-axis, d)
for $\Gamma_B(n)$ **in** Γ_B, **from** $n = \iota$ **to** $n = N_B$ **do**
    ${}^{w}T_{\Gamma_B(n)} \leftarrow$ getTransform($\Gamma_B(n)$)
    for $\tilde{p}_j = [x, y, z]$ **in** Π **do**
        ${}^{\Gamma_B(n)}T_{\tilde{p}_j} \leftarrow$ getTransform($\tilde{p}_j$)
        ${}^{w}T_{\tilde{p}_j} \leftarrow {}^{w}T_{\Gamma_B(n)} \cdot {}^{\Gamma_B(n)}T_{\tilde{p}_j}$
        $\Omega_M({}^{w}T_{\tilde{p}_j}) \leftarrow \Omega_R({}^{\Gamma_B(n)}T_{\tilde{p}_j})$
    end for
end for

Fourth, a slice of the reachability volume is extracted (*getReachabilityVolumeSlice*), $\Pi \subset \Omega_R$, located in the X-axis at d distance. This slice is used to build the tunnel cost volume, as follows. At each waypoint $\Gamma_B(n)$ in the rover trajectory Γ_B (from $\Gamma_B(\iota)$ onwards), Π is placed in the corresponding nodes of the 3D grid Ω_M by means of simple geometrical transformations (*getTransform*), being ${}^{j}T_{\kappa}$ the transformation matrix from j to κ. Repeating this process for every waypoint in Γ_B ends up creating a tunnel-shaped cost volume. Note that the tunnel is built differently at the first and last waypoints. The first waypoint $\Gamma_B(0)$ places every slice Π from the beginning of the reachability volume Ω_R, in the X-axis, to the deployment distance d. Conversely, the last waypoint $\Gamma_B(N_B)$ places every slice Π from d until the end of Ω_R. This way, the tunnel is completely closed, which reduces the computational cost of the later path planning algorithm.

Finally, the generated 3D cost volume Ω_M is used to plan the arm path Γ_M inside the tunnel, which is composed of S waypoints $\Gamma_M(s)$. The planner used for this purpose is FMM in a 3D version, which still provides optimal, smooth, and continuous trajectories with low computational cost (Sethian, 1999). FMM3D is defined similarly to standard FMM, as in Chapter 3.2, but considering now the height z as a third dimension in the 3D cost volume Ω_M and its nodes. The generated manipulator motion will be partially-optimal thanks to the use of FMM3D, in the sense that it is optimal in compliance with an already optimal rover path, but does not consider the mobile manipulator problem as a whole, which could lead to local minimum solutions.

Before obtaining the definitive arm joints profile $\Lambda = \{q_0, q_1, ..., q_n, ..., q_{N_B}\}$, which indicates the joints position q_n of the manipulator at every waypoint $\Gamma_B(n)$ of the rover base trajectory, it is necessary to establish a relation between the rover and manipulator trajectories, Γ_B and Γ_M. Each rover waypoint $\Gamma_B(n)$ is assigned with one or more waypoints of the wrist trajectory $\Gamma_M(s)$, subject to the previously defined deployment distance d. This assignment takes place as follows. Starting from

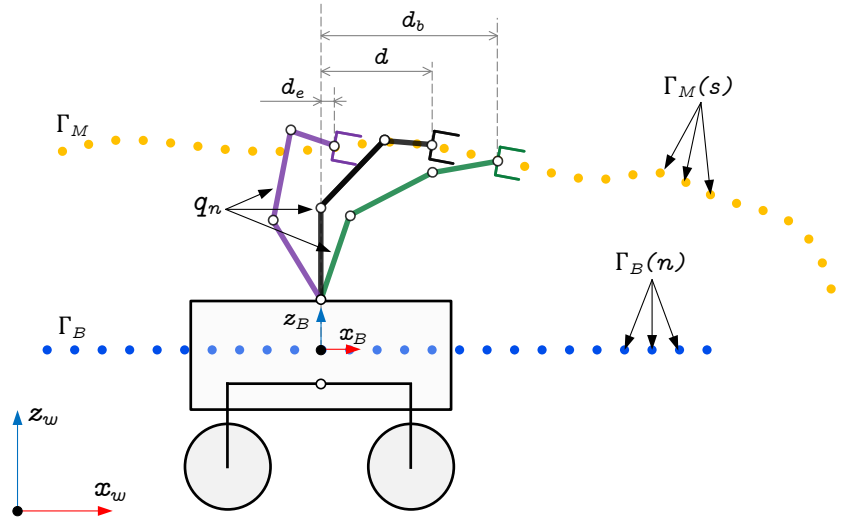
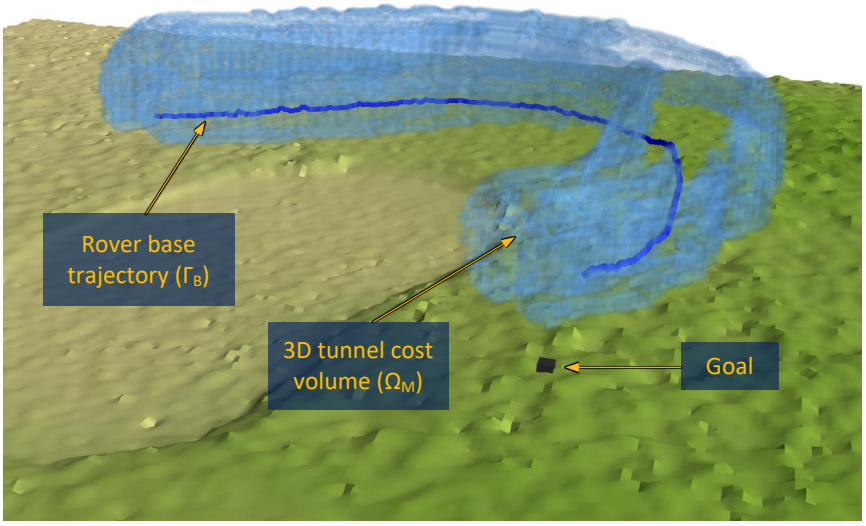


FIGURE 4.4: Scheme showcasing different deployment distances d. For a particular rover base waypoint $\Gamma_B(n)$, a small deployment distance (d_e) maintains the arm end effector closer to the body. Conversely, a big deployment distance (d_b) keeps the end effector farther.

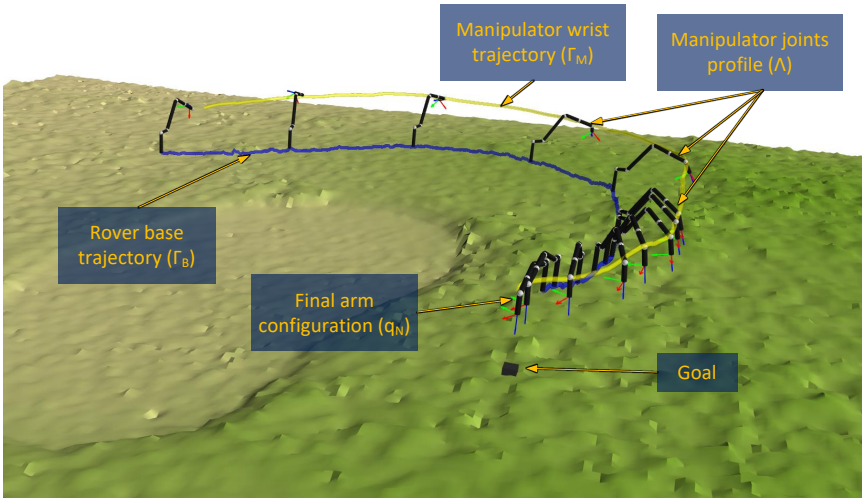
the last rover waypoint $\Gamma_B(N_B)$, each $\Gamma_B(n)$ gets assigned every manipulator waypoint $\Gamma_M(s)$ that is at a distance lower than d from the current base waypoint $\Gamma_B(n)$. The current base waypoint $\Gamma_B(n)$ will, thus, seize some assignments from the base waypoints that have already been checked. This procedure maintains the arm at d distance from the rover since the arm will always try to reach a manipulator waypoint $\Gamma_M(s)$ that is at d distance from the current base waypoint $\Gamma_B(n)$.

As a result, the motion planner can be configured to behave differently based on the deployment distance d, as can be observed in Figure 4.4. A long d (longer than half of the workspace W_R size, d_b) causes the arm (green) to mobilize early during the operation, keeping the end effector further from the rover body. This approach is called beginning deployment. Conversely, a short d (shorter than half of the workspace W_R size, d_e) leads to the opposite effect, i.e. the arm (purple) mainly moves at the end of the operation, maintaining the end effector closer to the body, what is called end deployment. Additionally, there exists the possibility of increasing progressively the deployment distance (from d_e to d_b) during the trajectory, which generates smoother arm motions. Called progressive deployment, this approach is very interesting since it optimizes the efficiency of the motion plan by distributing equally the arm motion throughout the whole rover base trajectory, as it is deeply analyzed in Chapter 6.3.

Finally, the complete manipulator motion profile Λ is generated by means of the IK model of the manipulator. At each rover waypoint $\Gamma_B(n)$, the IK generates the corresponding manipulator configuration q_n, sequentially obtaining the motion profile. An example of a 3D cost volume Ω_M (i.e. the tunnel) is shown in Figure 4.5a; the later manipulator trajectory and its corresponding arm profile are shown in Figure 4.5b. As can be observed, it corresponds to a progressive deployment



(a) 3D cost volume Ω_M, which is a light blue tunnel surrounding the rover path Γ_B. The rover path is represented as a blue line, and the goal sample is represented as a black box.



(b) Arm motion profile Λ, including the rover path Γ_B as a blue line, the manipulator trajectory Γ_M as a yellow line and the manipulator as a chain of black links. The goal sample is represented as a black box.

FIGURE 4.5: Mobile manipulator motion plan example for a progressive deployment distance d.

configuration of the motion planner, i.e. the arm is smoothly deployed during the motion. Remark that manipulators with spherical wrists have crucial advantages when computing Λ. In the first place, the position joints q^p can be obtained with the IK of the wrist. This reduces the computational cost of the algorithm since the complexity of the IK of the wrist is lower. In the second place, the orientation joints q^o can be linearly deployed considering that the whole manipulator trajectory is safe, as aforementioned. This linear deployment of q^o is simple, predictable, optimal, and computationally negligible.

Coupled base-arm trajectory controller

It is necessary to use a coupled trajectory controller to execute the generated plan. This component has to ensure, on the one hand, that the rover base properly tracks the base trajectory Γ_B, as explained in Section 4.2. On the other hand, it also has to command the adequate position and/or velocities to the arm joints to track the motion profile Λ, depending on the current rover base waypoint $\Gamma_B(n)$. Clearly, executing both motion plans (base, arm) coordinately is not straightforward, since the time required to reach the next waypoint $\Gamma_B(n)$ and configuration q_n will differ during the execution. To avoid this, a coupled base-arm trajectory controller is proposed.

The coupled controller works as follows. First, the modified conservative pure-pursuit trajectory controller, which includes the LSC presented in Section 4.2, generates the commands to follow the rover base trajectory, ensuring that the rover is always close to the trajectory inside the safety corridor. Second, the nominal speeds of the rover base and the manipulator's joints are used to estimate the required time for both systems to reach the next waypoint $\Gamma_B(n)$ and configuration q_n. Third, if one of the systems (base, arm) needs more time than the other to reach the next waypoint, which is likely to happen, then the other system is slowed down proportionally, to ensure both systems reach the next waypoint $\Gamma_B(n)$ and configuration q_n at the same time, as planned. Even though this approach slows down the whole motion if one of the systems is not fast enough, the coordination of both systems is significantly faster than a decoupled solution, as will be demonstrated in Chapter 6.3.

4.4 Summary and conclusions

This chapter presents the second contribution of this doctoral thesis: a coupled base-arm motion planning algorithm to perform a MM task once a planetary rover has reached an area of scientific interest. Considering the case of a highly restricted arm workspace, the proposed algorithm is able to generate a motion plan that, on the one hand, leaves the goal reachable for the manipulator by precisely planning and controlling the last stretch of the rover base path and, on the other hand, ensures the safety of the manipulator motion by preventing self-collisions offline using the real

workspace of the manipulator. The motion planner makes the rover able to reach a goal avoiding the obstacles in the scenario and smoothly orient itself to leave the goal inside the manipulator workspace, while the arm performs its particular motion plan to be ready to carry out the desired task as soon as the rover reaches the goal. Finally, the proposed motion planner has been validated by means of simulation, field tests, and a benchmark with an off-the-shelf MM motion planner, as explained later in Chapter 6.3.

Though this MM motion planner guarantees the safety of the robot during the operation in a simple, comprehensive, and straightforward manner, it relies on a geometric computation of the trajectories both for the mobile base and the manipulator. As a result, this method cannot optimize task-level parameters like energy spent. Furthermore, despite the fact that the motion is computed coordinately, this planner does not consider the whole MM system as one. Consequently, it cannot use the DoFs of the mobile base to compensate for the limitations of the manipulator, or vice-versa. These issues can be solved, nevertheless, by using optimization-based full-body motion planners. Let us explore this type of optimal solution in the next chapter.

Chapter 5

Optimal motion planning for over-actuated mobile platforms

"Obex agendi procedit agendi. Quod obstatur, fit via" -
"The impediment to action advances action. What stands in the way becomes the way"

Marcus Aurelius
Meditations, 2nd century A.D.

5.1 Introduction

Motion planning and control of mobile systems is a well-known problem in the literature, as explained in the previous chapters. It entails many challenges depending on the characteristics of the platform, the scenario, and the goal task. Nonetheless, it is remarkable that most systems have a common challenge for the motion planner: the computational effort. Mobile platforms' computational resources are usually limited, even more for space (Gerdes et al., 2020; Araguz, Bou-Balust, and Alarcón, 2018), air (Kratky et al., 2021) or underwater (Huang, Chen, and Guo, 2018) applications; or small-sized systems like sub-gram robots (St. Pierre and Bergbreiter, 2019). On top of that, the computational requirements increase substantially according to the system complexity, such as platforms that have redundant DoF, i.e. have more actuators than required to reach a particular pose. These are commonly called over-actuated platforms, and a clear example can be found in mobile manipulators. The over-actuation entails the existence of infinite solutions for the motion planning problem since the same final pose can be reached differently depending on the selected joints to be moved. Although a drawback in computational terms, over-actuation is an advantage for motion performance, since there is a great variety of solutions where to look for the one that best fits the problem's needs.

Multiple techniques have been used to solve the motion planning problem for over-actuated platforms, such as optimal control, potential fields, artificial intelligence, or probabilistic algorithms. As extensively analyzed in Chapter 2.3, one of the most suitable techniques to tackle over-actuation without excessive computational load is optimal control: since there are infinite combinations of joint movements to reach the goal, the optimization will find at least the locally optimal one by defining a cost function that could take into consideration energy consumption, required time, etc. Optimal control methods find computational hindrances when considering constraints and dynamics. This can be mitigated by reducing as much as possible the average number of iterations until convergence of the optimization, which is achievable through the so-called warm start. The objective of warm start is to provide the optimization algorithm with a fast-calculated initial solution to the problem, which is not necessarily feasible but is used as a starting point that places the algorithm close to the convex area surrounding a local (or global) optimal solution. Hereby, the optimization is boosted to find the solution faster, in much fewer iterations.

The latest contributions highlight the use of several stages to warm start the optimization, as shown in (Thakar et al., 2019), where a sequential refinement method for optimization was used to generate trajectories for a mobile manipulator to pick up parts. This method keeps introducing problem constraints (grasping pose, gripper speed, collisions) sequentially to the optimization problem, continuously warm-starting and refining the solution, which improves the performance of the motion planner in comparison with the cold-started planner. These results were extended by the authors in (Kabir et al., 2021), where a multi-staged warm-started motion planner for a group of robots (up to three manipulators or a mobile manipulator with two robotic arms) was presented, including a deep analysis of the best sequence of introduction of the problem constraints (position, velocity, orientation of the end effector, collisions). The same multi-staged warm-started motion planner was also used in (Thakar et al., 2022) for surface disinfection with mobile manipulators, generating first a path to cover the goal area employing a branch and bound-based tree search. This generated path was used as a constraint to generate the mobile manipulator actuators' trajectory through the motion planner. As can be observed, these multi-staged warm start approaches have been applied to path-constrained trajectory generation problems, i.e. tasks where the end effector trajectory is precisely known a priori, thus, it is added as a problem constraint in terms of end effector position, velocity, and orientation. However, there are other cases where the initial trajectory is completely unknown or undefined, which can be denominated as goal-constrained trajectory generation problems, as previously mentioned in this thesis.

Seeing the current state of the art, it is clear that over-actuated systems are especially difficult to handle by optimization motion planners, with the computational

requirements becoming a bottleneck as the system constraints and complexity increase. To tackle this issue, a multi-staged warm start arises as an interesting approach, which has demonstrated promising results in solving path-constrained trajectory generation problems. Such an computationally-lightweight approach could be brought to a planetary exploration rover, to perform a MM task inside an area of scientific interest. Hence, the third contribution of this thesis is presented, a Multi-staged Warm started optimization Motion Planner (MWMP) for over-actuated mobile platforms, which reduces the computational cost by sequentially introducing the system constraints. MWMP can be used for any over-actuated mobile system, although it is mainly aimed at MM.

The presented motion planner includes two main advancements to the state-of-the-art. First, a generic model for over-actuated mobile platforms, that can be applied to systems with several serial kinematic chains. This model considers the system dynamics and external disturbances to optimize the platform energy consumption later during the planning stage. Second, a novel sequence for multi-staged warm-started motion planning, aimed to solve the goal-constrained trajectory generation problem on over-actuated mobile platforms. The first warm start stage is a path planner, making the most of the first contribution of this doctoral thesis, that computes an initial trajectory for the mobile platform, the second stage takes that initial trajectory as a warm start and solves only the unconstrained problem, and the third stage uses this unconstrained solution as a warm start to obtain the final motion plan with a constrained optimization solver.

The rest of this chapter is organized as follows. Section 5.2 shows the first contribution of this work: the generic model for over-actuated mobile platforms. The second contribution, MWMP, is presented and explained in detail in Section 5.3. Lastly, Section 5.4 states a summary of this chapter and possible lines of future work.

5.2 Generic mobile platform model

Considering $\mathbb{T} = \{t_0, t_1, \dots, t_n, \dots, t_N\}$ as the set of N equally-sized time steps that define the time horizon, being Δt the time step size, let us define a state space model under linear discrete approximations as depicted in (5.1):

$$x(n+1) = A(n)x(n) + B(n)u(n) \tag{5.1}$$

Where $A(n)$ and $B(n)$ are the state transition and input distribution matrices respectively, and $x(n)$ and $u(n)$ are the state and actuation vectors of the system respectively, at time step t_n. Note that a discrete model is used, i.e. n is used as an index to get the values at time step t_n ($t_n = n\Delta t$). Also note that, in the following, any ${}^{j}P_{\kappa}$ expresses the pose of κ w.r.t. j, as ${}^{j}P_{\kappa} = [p\ \phi]$, i.e. a position vector, in 3D $p = [x\ y\ z]$, and an orientation vector, in 3D $\phi = [\varphi\ \vartheta\ \psi]$, the roll, pitch, yaw Euler Angles. Assuming a generic mobile platform composed of a set of K kinematic

chains, ${}^{j}P_{\kappa}$ is the pose of the tip link reference frame of the kinematic chain κ w.r.t. j, having w as the world reference frame. For instance, the center reference frame of a mobile platform would be ${}^{w}P_1$, and if it is equipped with a manipulator, the manipulator end effector pose w.r.t. the mobile platform would be ${}^{1}P_2$.

Thereafter, a generic mobile platform is represented as a set of K kinematic chains, with the pose of its tip link reference frame represented as ${}^{j}P_{\kappa}$. The position of all the actuation joints belonging to kinematic chain κ are denoted as vector q_{κ}, including rotational and translational joints. Thus, the corresponding state vector $x(n)$ of a generic mobile platform is defined as (5.2).

$$\begin{aligned} x(n) = \big[{}^{w}P_1 \; {}^{w}\dot{P}_1 \cdots {}^{w}P_K \; {}^{w}\dot{P}_K \; {}^{1}P_2 \; {}^{1}\dot{P}_2 \cdots {}^{K-1}P_K \\ {}^{K-1}\dot{P}_K \; q_1 \; \dot{q}_1 \; \ddot{q}_1 \; q_2 \; \dot{q}_2 \; \ddot{q}_2 \cdots q_K \; \dot{q}_K \; \ddot{q}_K \big]^{T} \end{aligned} \tag{5.2}$$

With ${}^{w}\dot{P}_{\kappa}$ the speed of the kinematic chain κ w.r.t. the world reference frame; ${}^{\kappa-1}\dot{P}_{\kappa}$ the speed of the kinematic chain κ w.r.t. the kinematic chain $\kappa - 1$; and $\dot{q}_{\kappa}$, $\ddot{q}_{\kappa}$ the speed and acceleration, respectively, of each actuation joint of κ. Conversely, assuming a force/torque controlled platform, the actuation vector $u(n)$ is defined as (5.3).

$$u(n) = \Big[e_1 \, e_2 \cdots e_K \; \delta_1 \cdots \delta_Z \; \Big]^{T} \tag{5.3}$$

Where e_{κ} represents the actuation effort vector, i.e. forces or torques of the joints of κ, and δ_z are external disturbances applying forces to the system, for instance, gravity. The total number of external disturbances is defined as Z. Note that, even though the external disturbances δ_z are included in the model inside the actuation vector, doubtlessly they are not under the control of the system. Hence, the values of the disturbances vector δ_z should remain fixed according to their estimates, e.g. $9.81\,\mathrm{m/s^2}$ for gravity.

Once defined the state and input vectors, a generic representation of the state transition matrix $A(n)$ is shown in (5.4), where $\mathbb{I}_j$ represents the identity matrix with size (jxj), $||q_{\kappa}||$ the number of DoF of the kinematic chain κ, ${}^{j}\mathcal{R}_{\kappa}$ a rotation matrix given the pose defined in ${}^{j}P_{\kappa}$, ${}^{j}\mathcal{J}_{\kappa}$ the Jacobian matrix relating articular with Cartesian speeds for the kinematic chain κ w.r.t. j, I_{κ} and V_{κ} the inertia and Coriolis/centrifugal matrices of κ respectively, and Δt the time step size. On the other hand, the input distribution matrix $B(n)$ is depicted in (5.5), with $\beta_z^{\kappa} = -I_{\kappa}^{-1}\gamma_z^{\kappa}$, and γ_z^{κ} a matrix representing the effect of the perturbation δ_z into the joints of kinematic chain κ. It is crucial to remark that, with these definitions of $A(n)$ and $B(n)$, some of their terms could be non-linear, which would eventually hinder the proper functioning of the motion planner. For that purpose, the use of Taylor Series Linearization (TSL) on those terms is recommended.

Finally, it is key to define the system limits, via state-input and pure state constraints. On the one hand, the actuation effort e_{κ} limits have to be specified as state-input constraints as expressed in equation (5.6), where $C(n)$ is filled with zeros since

$$A(n) = \begin{bmatrix}
\mathbb{I}_6 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & {}^{w}\mathcal{J}_1\Delta t & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & {}^{w}\mathcal{J}_1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & \mathbb{I}_6 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & {}^{w}\mathcal{R}_{K-1}\,{}^{K-1}\mathcal{J}_K\Delta t & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & {}^{w}\mathcal{R}_{K-1}\,{}^{K-1}\mathcal{J}_K & 0 \\
0 & 0 & \cdots & 0 & 0 & \mathbb{I}_6 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & {}^{1}\mathcal{J}_2\Delta t & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & {}^{1}\mathcal{J}_2 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & \mathbb{I}_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & {}^{K-1}\mathcal{J}_K\Delta t & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & {}^{K-1}\mathcal{J}_K & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \mathbb{I}_{||q_1||} & \mathbb{I}_{||q_1||}\Delta t & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \mathbb{I}_{||q_1||} - I_1^{-1}V_1\Delta t & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & -I_1^{-1}V_1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \mathbb{I}_{||q_2||} & \mathbb{I}_{||q_2||}\Delta t & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \mathbb{I}_{||q_2||} - I_2^{-1}V_2\Delta t & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & -I_2^{-1}V_2 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & \mathbb{I}_{||q_K||} & \mathbb{I}_{||q_K||}\Delta t & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \mathbb{I}_{||q_K||} - I_K^{-1}V_K\Delta t & 0 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -I_K^{-1}V_K & 0
\end{bmatrix} \tag{5.4}$$

$$B(n) = \begin{bmatrix}
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
I_1^{-1}\Delta t & 0 & \cdots & 0 & \beta_1^1\Delta t & \cdots & \beta_Z^1\Delta t \\
I_1^{-1} & 0 & \cdots & 0 & \beta_1^1 & \cdots & \beta_Z^1 \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & I_2^{-1}\Delta t & \cdots & 0 & \beta_1^2\Delta t & \cdots & \beta_Z^2\Delta t \\
0 & I_2^{-1} & \cdots & 0 & \beta_1^2 & \cdots & \beta_Z^2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & I_K^{-1}\Delta t & \beta_1^K\Delta t & \cdots & \beta_Z^K\Delta t \\
0 & 0 & \cdots & I_K^{-1} & \beta_1^K & \cdots & \beta_Z^K
\end{bmatrix} \tag{5.5}$$

these constraints do not depend on the states, $D(n)$ indicates with ones or zeros which actuation effort limit is being defined, and $r(n)$ includes the actual limit values. On the other hand, as pure state constraints it is important to define the kinematic chains world ${}^{w}\dot{P}_{\kappa}$ and relative ${}^{\kappa-1}\dot{P}_{\kappa}$ speed limits, and the position q_{κ}, velocity $\dot{q}_{\kappa}$ and acceleration $\ddot{q}_{\kappa}$ limits of the actuation joints, as done in equation (5.7). Particularly, $G(n)$ indicates which state limit is being defined, filled adequately with ones or zeros, and $h(n)$ includes the values of those limits.

$$C(n)x(n) + D(n)u(n) + r(n) \leq 0 \tag{5.6}$$

$$G(n)x(n) + h(n) \leq 0 \tag{5.7}$$

Considering this general definition of $x(n)$, $u(n)$, $A(n)$ and $B(n)$, the motion planning problem for a mobile platform with K serial kinematic chains can be redefined as finding a set of actuation efforts (forces/torques) e_{κ} that generate a motion profile (q_{κ}, $\dot{q}_{\kappa}$, $\ddot{q}_{\kappa}$) for each joint of the platform, to place the tip link of the last kinematic chain in certain poses (${}^{w}P_{K}$, ${}^{w}\dot{P}_{K}$), given the effect of external perturbations (δ_{z}) and the system limits, expressed through the state-input constraints ($C(n)$, $D(n)$ and $r(n)$) and the pure state constraints ($G(n)$ and $h(n)$). Note that some of the states are not strictly necessary in the model (${}^{w}\dot{P}_{\kappa}$, ${}^{\kappa-1}\dot{P}_{\kappa}$, $\ddot{q}_{\kappa}$), but, as will be explained later, are helpful to set desired behaviors by tuning their corresponding costs, or to establish system constraints as aforementioned. Also bear in mind that, although required for the appropriate behavior of the motion planner, the linearization with $A(n)$ and $B(n)$ induces execution errors. These errors' significance depends on the time step size Δt, being negligible if Δt is sufficiently small.

5.3 Multi-staged motion planning

The proposed MWMP sequence is designed to deal with highly complex, over-actuated systems, looking for an efficient solution to the motion planning problem without severely impacting the computational resources of the system. For that purpose, a sequential warm start procedure is defined in this section, which reduces substantially the average number of iterations until convergence and, thereupon, the computational cost of the planner. A general overview of the functioning of the algorithm and its evolution through the different stages is shown in Figure 5.1, which is further explained below.

Each stage is executed in a particular order, sequentially. The first stage, called Path Planning Warm Start (PPWS), consists of a path planner based on the FMM, although any other path planning algorithm could be used. FMM has been selected since it provides globally optimal and smooth solutions, with comparable computational cost to other non-optimal state-of-the-art planners (Sethian, 1999). As can be observed in Figure 5.1, the path planner requires three inputs to compute the trajectory: the platform initial and goal positions w.r.t. the world frame, ${}^{w}p_{1}(0)$ and

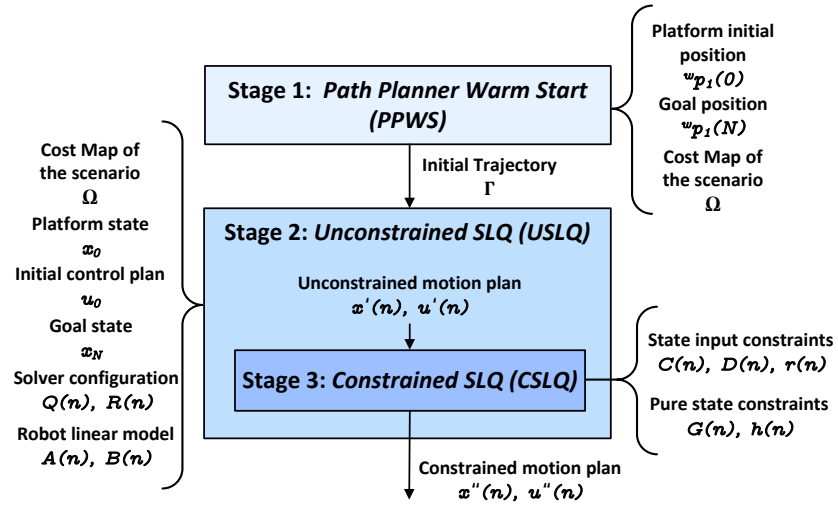


FIGURE 5.1: Scheme summarizing the general functioning of the multi-staged motion planning approach. In the first stage (PPWS) an initial trajectory Γ is computed using FMM. In the second stage (USLQ), this trajectory is used to warm start the SLQ optimization algorithm to solve the unconstrained motion planning problem. If the planned motion $x'(n), u'(n)$ satisfies the constraints, the algorithm finishes. Otherwise, the third stage (CSLQ) takes the unconstrained solution as a warm start and computes the final motion plan $x''(n), u''(n)$, using again the SLQ optimization solver but with constraints compliance.

$^{w}p_1(N)$ respectively, and a cost map Ω representing the characteristics of the scenario, with higher costs where the mobile platform finds difficulties to traverse, like obstacles. Given this information, FMM takes two different steps to generate a path: first, a wave expansion, second, the trajectory extraction. This separation is advantageous since it allows the generation of new trajectories quickly as long as the goal remains the same, as will be explained later. As an output, the path planner generates the global optimal path Γ to reach the goal $^{w}p_1(N)$ from the platform initial position $^{w}p_1(0)$ on the cost map Ω.

To accelerate the convergence speed, the extracted trajectory Γ is forwarded as a warm start to the next stage called Unconstrained Sequential Linear Quadratic regulator (USLQ), which makes use of an optimal solver called SLQ regulator (Sideris and Bobrow, 2005), which is based on the Riccati equation. Although many other optimal solvers could also be used, SLQ has been selected due to its efficiency when solving non-linear discrete optimal control problems with near-quadratic convergence, besides, requiring only first derivative information of the system states. As inputs, the second stage requires the current and goal states of the platform, x_0 and x_N, an initial actuation plan u_0 usually filled with zeros, a configuration for the solver i.e. the quadratic costs defined by the state matrix $Q(n)$ and the input matrix $R(n)$, and a linear state space model of the system, represented by the state transition matrix $A(n)$ and the input distribution matrix $B(n)$. USLQ outputs, then, a complete unconstrained motion plan for the whole time horizon $\mathbb{T}$, with $x'(n)$ the states and $u'(n)$ the actuation.

The above solver does not consider constraints yet, which implies faster convergence but could lead to unfeasible solutions. Thus, in the third stage, every generated solution is checked to confirm if the system constraints are satisfied. If they are, then the unconstrained motion plan is taken as correct and the planning pipeline finishes, $x''(n) = x'(n)$ and $u''(n) = u'(n)$, being $x''(n)$ and $u''(n)$ the final motion plan states and actuation respectively. The overall computational cost of MWMP is reduced noticeably if this happens, since this third stage, called Constrained Sequential Linear Quadratic regulator (CSLQ) (Sideris and Rodriguez, 2011), is the computationally most expensive. Otherwise, again the optimal solver is used to consider the system constraints, using the unconstrained solution $x'(n)$, $u'(n)$ as a warm start to boost the CSLQ performance, as will be shown with the obtained results in Chapter 6.4. This way CSLQ only has to refine the unconstrained solution to ensure constraint compliance, i.e. it already starts in the vicinity of the constrained optimal solution. Two main types of constraints are to be defined. On one hand, the system state-input constraints, which indicate actuation caps under particular system states, for instance, joint effort limits, defined by the state-input constraints distribution matrices, with $C(n)$ for the state and $D(n)$ for the input, and the state-input constraints level $r(n)$. On the other hand, the pure state constraints represent restrictions on the state vector that can not be overcome, for instance, joint position limits, defined by the pure state constraints distribution matrix $G(n)$ and the pure state constraints level $h(n)$.

After finding a constraint-compliant solution, the complete motion plan $x''(n)$, $u''(n)$ is ready to be followed by the platform. Any tracking or control algorithm could be used to accurately follow the planned motion and compensate for disturbances. For instance, MPC, Event-triggered replanning, or other procedures could be adequate for this purpose, depending on the platform and mission requirements. Remark that the main prerequisite to using the motion planner is to obtain a state space model of the mobile platform, defining $A(n)$ and $B(n)$, with the generic procedure for modeling over-actuated mobile platforms including several kinematic chains presented above.

Finally, in the following the two main methods used in the motion planner are analyzed in detail, FMM as a path planner in the first stage and SLQ as an optimal solver for motion planning in the second and third stages.

Trajectory planning

The goal of the first stage, PPWS, is to generate an initial reference trajectory for the mobile platform to reach the goal, which will be later used as a warm start of the optimization algorithm to accelerate its convergence, as can be seen in Figure 5.1. In particular, we propose FMM (Sethian, 1999) again as the warm start path planner. As extensively presented during this doctoral dissertation, FMM has been selected since it extracts a globally optimal, smooth, and continuous path to reach the goal given a cost map. See Chapter 3.2 for a detailed description of the algorithm.

In summary, FMM generates the optimal trajectory Γ from the starting position of the platform until the goal. Considering 1 as the kinematic chain defining the mobile platform, then $\Gamma(0) = \tilde{p}_0 = {}^{w}p_1(0)$ and $\Gamma(N) = \tilde{p}_g = {}^{w}p_1(N)$. Following the nomenclature defined in Chapter 3.2, the objective of FMM is to minimize the cost accumulated along the path $\Omega(\Gamma(\tilde{p}_0, \tilde{p}_g, l))$ by solving the optimization problem defined in (5.8-5.9), with $\Gamma(\tilde{p}_0, \tilde{p}_g, l)$ the continuous function that gives a point $p \in \Omega$ depending on the path length l from the starting point $\tilde{p}_0$, being l_g the total length of the path, and Υ the so-called cost to go, i.e. the expanded FMM wave.

$$\underset{\Gamma(\tilde{p}_0, \tilde{p}_g)}{\text{Minimize}} \quad \Upsilon(\tilde{p}_0, \tilde{p}_g) = \int_0^{l_g} \Omega(\Gamma(\tilde{p}_0, \tilde{p}_g, l))\, dl \tag{5.8}$$

$$\text{with} \quad \Upsilon(\tilde{p}_j = \tilde{p}_g, \tilde{p}_g) = 0 \tag{5.9}$$

Apart from the Cartesian positions derived from the FMM path planner, it is beneficial to enrich the generated path Γ with some more information. On the one hand, the orientation of the platform at each waypoint is computed to also warm start the orientation states of the system. Including the robot heading in the trajectory is particularly helpful for platforms with non-holonomic constraints, since it gives the mobile platform a big hint on how to properly follow the path, being the yaw obtained geometrically considering the position of two consecutive waypoints. On the other hand, each waypoint should be timestamped, which can be easily done by interpolating the total expected time for finishing the operation t_N into each time step t_n. Note that there are many different approaches to estimating t_N according to the characteristics of the system, its nominal speed, or the use case, which is out of the scope of this work.

Later on, so the next stages USLQ and CSLQ converge faster, the trajectory Γ is used to warm start the pose of the platform ${}^{w}P_1$ at each time step t_n, in two main senses. First, ${}^{w}P_1(n)$ is initialized to $\Gamma(n)$. Second, although ${}^{w}P_1(n)$ will freely evolve during the optimization, a quadratic cost is defined so ${}^{w}P_1(n)$ uses $\Gamma(n)$ as a reference solution, as explained later.

To finalize, remark that the wave propagation is, for FMM, the most computationally expensive step, and the trajectory extraction is computationally negligible. This is very convenient for replanning the motion since a new optimal trajectory from the current pose of the platform can be obtained quickly, i.e. without recomputing the cost to go Υ. This only needs to be done once, offline, or in case the goal changes. Again, refer to (Sánchez-Ibáñez et al., 2019b) and Chapter 3.3 for more details about DyMu, the FMM-based path planner used within this stage.

Sequential Linear Quadratic optimal solver

The next stages of MWMP make use of an optimal solver, which tackles the unconstrained problem in stage two (USLQ) and the constrained one in stage three (CSLQ), to generate a motion plan for the platform, as shown in Figure 5.1. As aforementioned, the solver is called SLQ (Sideris and Bobrow, 2005), (Sideris and Rodriguez, 2011).

The standard formulation of a discrete-time optimal control problem is shown in (5.10-5.13).

$$\underset{u(n),x(n)}{\text{Minimize}} \quad J = \Phi(x(N)) + \sum_{n=0}^{N-1} L(x(n),u(n),n) \tag{5.10}$$

$$\text{subject to} \quad x(n+1) = f(x(n),u(n)), \quad x(0) = x_0 \tag{5.11}$$

$$C(n)x(n) + D(n)u(n) + r(n) \leq 0 \tag{5.12}$$

$$G(n)x(n) + h(n) \leq 0 \tag{5.13}$$

Where $x(n)$ is the state vector at time step t_n, noticeably, $x(0)$ is the initial state and $x(N)$ is the final one, and $u(n)$ is the actuation vector; x_0 defines the initial state of the system, at time step $t_0 = 0$. Additionally, (5.12) represents the state-input inequality constraints with the constraints distribution matrices $C(n)$, $D(n)$ and the constraints level vector $r(n)$, and (5.13) the pure state constraints with the constraints distribution matrix $G(n)$ and the constraints level vector $h(n)$, as aforementioned.

In this formulation, J is defined as the total cost to go, and it is composed of $\Phi(x(N))$, the terminal cost, and $L(x(n),u(n),n)$, the intermediate cost. Assuming a quadratic performance index, these are defined in (5.14) and (5.15) respectively.

$$\Phi(x(N)) = \frac{1}{2}[x(N) - x^0(N)]^T Q(N)[x(N) - x^0(N)] \tag{5.14}$$

$$\begin{aligned} L(x(n),u(n),n) = \frac{1}{2}&[x(n) - x^0(n)]^T Q(n)[x(n) - x^0(n)] \\ &+[u(n) - u^0(n)]^T R(n)[u(n) - u^0(n)] \end{aligned} \tag{5.15}$$

With $Q(n)$, $R(n)$ defined as the state and input quadratic cost matrices respectively, at time step t_n, and $x^0(n)$, $u^0(n)$ the state and input references or targets. Note that $x^0(N) = x_N$ is the terminal state goal and $Q(N)$ is the terminal state cost matrix, which is usually configured to have considerably high costs to ensure that the goal defined in $x^0(N)$ is accomplished. Note also the importance of properly tuning $Q(n)$ and $R(n)$ to precisely represent the desired behavior of the system.

Solving the aforementioned discrete-time optimal control problem, the objective of the algorithm is to generate the motion plan $x(n)$, $u(n)$ for every time step t_n within the given time horizon $\mathbb{T} = \{t_0, t_1, \dots, t_n, \dots, t_N\}$. To that purpose, several

inputs are required. On one hand, the current state of the system x_0 and the desired goal state x_N are needed. On the other hand, as extensively explained above, an initial trajectory Γ is fed to the solver to accelerate the convergence speed. Consequently, the corresponding intermediate state costs $Q({}^wP_1, n)$ must be tuned with appropriate costs at every time step. These have to be high enough to help the solver, guiding it closer to the globally optimal path, but low enough to avoid forcing the solver to follow exactly the provided trajectory, which would reduce the variety of possible solutions to be explored.

Additionally, obstacle avoidance is always a requirement for any mobile platform. Although the warm start trajectory already considers the presence of obstacles in the scenario, another layer of obstacle avoidance is required, since the solver will draw a probably similar but new trajectory. Thus, USLQ and CSLQ need the same cost map Ω used in FMM at PPWS. $\Omega({}^wP_1(n))$ corresponds to a repulsive cost that gets the system away from danger, which means that the cost increases as the system gets closer to obstacles. This way the generated trajectories for the platform base pose ${}^wP_1$ dynamically get away from obstacles during the motion planning process in function of the cost map Ω, meanwhile trying to follow the warm start trajectory Γ. The intermediate cost defined above in (5.15) needs, then, to be reformulated as in equation (5.16), adding the repulsive cost $\Omega({}^wP_1(n))$, i.e. the cost value associated to the platform pose ${}^wP_1$ at time step t_n. Remark that it is key to maximize the continuity and linearity of the cost map, otherwise the solver will find difficulties to converge when encountering non-linearities in the costs.

$$\begin{aligned} L(x(n), u(n), n) = & \frac{1}{2}[x(n) - x^0(n)]^T Q(n)[x(n) - x^0(n)] \\ & + [u(n) - u^0(n)]^T R(n)[u(n) - u^0(n)] \\ & + \Omega({}^wP_1(n)) \end{aligned} \tag{5.16}$$

Finally, an overview of the functioning of the solver for the discrete-time optimal control problem defined in (5.10-5.13) is depicted in Algorithms 3 and 4. Summarizing, given the current state x_0, the current actuation plan $u(n)$, the warm start trajectory Γ, the quadratic costs $Q(n)$ and $R(n)$, the system model $A(n)$ and $B(n)$, the cost map of the scenario Ω, the state-input $C(n), D(n), r(n)$ and the pure state $G(n), h(n)$ constraints, this solver computes efficiently the motion plan $x(n), u(n)$ by iteratively obtaining step plans $\bar{x}(n), \bar{u}(n)$ to be applied to the current solution. To do so, the current active constraints are stored in $C_c(n)$, $D_c(n)$, and $G_c(n)$, which are later used to consider the constraints during the LQR solution computation. In particular, the state-input constraints are directly handled within the *Predefinitions* step, and the pure state constraints are managed later within the *State constraints management* step.

Algorithms 3 and 4 are based on the SLQ solver presented in (Sideris and Bobrow, 2005) and (Sideris and Rodriguez, 2011), with a few differences. First, the state target sequence $x^0(n)$ is initialized with the warm start trajectory, as aforementioned.

Algorithm 3 SLQ solver

Initialization
$x(0) \leftarrow$ getCurrentState()
$u(n) \leftarrow$ getCurrentControlPlan()
$x^0({}^wP_1) \leftarrow$ getWarmStartTrajectory()
$C(n), D(n), r(n), G(n), h(n) \leftarrow$ getConstraints()
repeat
  Linearization and quadratization
  $x(n) \leftarrow$ forwardSimulateSystem($x(0)$, $u(n)$)
  $Q(n), R(n) \leftarrow$ getQuadraticCosts()
  $A(n), B(n) \leftarrow$ getLinearizedSystem($x(n)$)
  $\Omega \leftarrow$ getObstaclesRepulsiveCost()
  $C_c(n), D_c(n), G_c(n) \leftarrow$ getActiveConstraints($C(n), D(n), r(n), G(n), h(n), x(n), u(n)$)
  Reference tracking
  $\bar{x}_0(n) \leftarrow Q(n)(x(n) - x^0(n))$
  $\bar{u}_0(n) \leftarrow R(n)(u(n) - u^0(n))$
  Predefinitions
  $\hat{D}(n) \leftarrow (D_c(n)R(n)^{-1}D_c(n)^T)^{-1}$
  $\hat{r}(n) \leftarrow -D_c(n)R(n)^{-1}\bar{u}_0(n)$
  $\hat{A}(n) \leftarrow A(n) - B(n)R(n)^{-1}D_c(n)^T\hat{D}(n)C_c(n)$
  $\hat{R}(n) \leftarrow B(n)R(n)^{-1}[\mathbb{I} - D_c(n)^T\hat{D}(n)D_c(n)R(n)^{-1}]B(n)^T$
  $\hat{Q}(n) \leftarrow Q(n) + C_c(n)^T\hat{D}(n)C_c(n)$
  $\hat{x}^0(n) \leftarrow \bar{x}_0(n) + C_c(n)^T\hat{D}(n)\hat{r}(n)$
  $\hat{u}^0(n) \leftarrow -B(n)R(n)^{-1}[\bar{u}_0(n) + D_c(n)^T\hat{D}(n)\hat{r}(n)]$
  Backward Pass - Riccati matrix difference equation
  $\hat{P}(N) \leftarrow Q(N)$
  $\hat{s}(N) \leftarrow \Omega({}^wP_1(N)) + \bar{x}_0(N)$
  for $n \leftarrow (N-1)$; n **in** $\mathbb{T}$ **do**
    $\hat{M}(n) \leftarrow (\mathbb{I} + \hat{R}(n)\hat{P}(n+1))^{-1}$
    $\hat{P}(n) \leftarrow \hat{Q}(n) + \hat{A}(n)^T\hat{P}(n+1)\hat{M}(n)\hat{A}(n)$
    $\hat{s}(n) \leftarrow \hat{A}(n)^T\hat{M}(n)^T\hat{s}(n+1) + \hat{A}^T(n)\hat{P}(n+1)\hat{M}(n)\hat{u}^0(n) + \hat{x}^0(n) + \Omega({}^wP_1(n))$
  end for
  Pure state constraints (ζ) management
  if isConstrained() **then**
    for ζ_m **in** $\mathcal{C}_x$ **do**
      $m \leftarrow$ getTimestepOfConstraint(ζ_m)
      $\Psi_c(m) \leftarrow G_c(m)$
      $y_c(m) \leftarrow 0$
      for $n \leftarrow (m-1); n--; n > 0$ **do**
        $\Psi_c(n) \leftarrow \Psi_c(n+1)\hat{M}(n)\hat{A}(n)$
        $y_c(n) \leftarrow y_c(n+1) + \Psi_c(n+1)\hat{M}(n)[\hat{u}^0(n) - \hat{R}(n)\hat{s}(n+1)]$
      end for
      $\mathcal{H}(\zeta_m), \Psi(\zeta_m), y(\zeta_m) \leftarrow h_c(m), \Psi_c(0), y_c(0)$
      for ζ_j **in** $\mathcal{C}_x$ **do**
        $n \leftarrow min(\zeta_m - 1, \zeta_j - 1)$
        $\mathcal{F}_{\zeta_m,\zeta_j}(n+1) \leftarrow 0$
        for $i \leftarrow n; i \geq 0$ **do**
          $\mathcal{F}_{\zeta_m,\zeta_j}(i) \leftarrow \mathcal{F}_{\zeta_m,\zeta_j}(i+1) - \Psi_c(i+1)\hat{M}(i)\hat{R}(i)\Psi_j(i+1)^T$
        end for
        $\mathcal{F}(\zeta_m, \zeta_j) \leftarrow \mathcal{F}_{\zeta_m,\zeta_j}(0)$
      end for
    end for
    $\nu \leftarrow \mathcal{F}^{-1}[-\Psi\bar{x}(n) + y + \mathcal{H}]$
    for $n \leftarrow 0, ..., N$ **do**
      $\hat{s}(n) \leftarrow \hat{s}(n) + \sum_{\zeta \in \mathcal{C}_x; \zeta \geq n} \Psi_c^T(n)\nu(\zeta)$
    end for
  end if

Algorithm 4 SLQ solver: continue

Forward Pass
for $n \leftarrow 0$; n **in** $\mathbb{T}$ **do**
  $\hat{v}(n) \leftarrow \hat{M}(n)(\hat{u}^0(n) - \hat{R}(n)\hat{s}(n+1))$
  $\bar{x}(n+1) \leftarrow \hat{v}(n) + \hat{M}(n)\hat{A}(n)\bar{x}(n)$
  $\lambda(n+1) \leftarrow \hat{s}(n+1) + \hat{P}(n+1)\bar{x}(n+1)$
  $\mu(n) \leftarrow \hat{D}(n)[C_c(n)\bar{x}(n) - D_c(n)R^{-1}(n)B(n)^T\lambda(n+1) + \hat{r}(n)]$
  $\bar{u}(n) \leftarrow -R^{-1}(n)[B^T(n)\lambda(n+1) + D_c(n)^T\mu(n) + \bar{u}_0(n)]$
end for
Computed step plan appliance
if checkConstraints$(x(n), \bar{x}(n), u(n), \bar{u}(n), C(n), D(n), r(n), G(n), h(n))$
 or isUnconstrained() **then**
  $\alpha \leftarrow$ computeLineSearch$(x(n), x^0(n), u(n), u^0(n), \bar{u}(n))$
else
  $\alpha \leftarrow$ satisfyConstraints$(x(n), \bar{x}(n), u(n), \bar{u}(n), C(n), D(n), r(n), G(n), h(n))$
end if
$x(n) \leftarrow x(n) + \alpha\bar{x}(n)$
$u(n) \leftarrow u(n) + \alpha\bar{u}(n)$
Termination conditions
convergence $\leftarrow$ checkTermination$(x(n), x^0(n), u(n), u^0(n), \bar{u}(n), \alpha)$
until convergence

Second, the obstacle repulsive cost $\Omega({}^wP_1(n))$ is included in the backward pass. Third, at each iteration, the constraints compliance is checked. If no new constraint is violated, then a standard *Line Search* for α is performed. The line search consists of finding the best α, which is the step size used to apply the step solutions $\bar{x}(n)$ and $\bar{u}(n)$, to reduce to the minimum the total cost to go J at each iteration of the solver. Otherwise, if any constraint is violated, α is generated particularly to satisfy the constraints. Fourth, several termination conditions are defined, on top of the algorithm convergence itself. In particular, one of these conditions checks that the last kinematic chain pose ${}^wP_K$ is close enough to the goal pose ${}^wP_K(N)$ to perform the desired task, depending on a given threshold, and another one ensures that the motion plan is thoroughly safe. Note that Algorithm 3 encompasses both the unconstrained and the constrained solvers, with $[C(n), D(n), r(n), G(n), h(n), C_c(n), D_c(n), G_c(n)] = 0$ in the unconstrained case, which means that $\hat{A}(n) = A(n)$, $\hat{R}(n) = B(n)R(n)^{-1}B(n)^T$, $\hat{Q}(n) = Q(n)$, $\hat{x}^0(n) = \bar{x}_0(n)$, $\hat{u}^0(n) = -B(n)R(n)^{-1}\bar{u}_0(n)$ and $\mu(n) = 0$. Besides, as can be observed, for the unconstrained case the *State constraints management* is not required, and the *Computed step plan appliance* is reduced to the first *Line Search*.

5.4 Summary and conclusions

In this chapter MWMP is presented, a motion planner for over-actuated mobile platforms capable of dealing with system dynamics and constraints, such as non-holonomic constraints or joint limits, without severely impacting the computational resources of the system. This is achieved by employing a multi-staged warm start approach, which initializes in several steps the optimal solver, SLQ. In particular, a novel pipeline with three different stages is used: first, an FMM-based path planner;

second, an unconstrained SLQ motion planner; third, a constrained SLQ motion planner. Although this pipeline is functional for most cases, it finds convergence hindrances when the constraints are too restrictive, since the planner is not able to escape local minima solutions. To solve this, it is foreseen as future work the division of the constrained SLQ stage into several steps, including each constraint sequentially, which aids the solver convergence as suggested in (Kabir et al., 2021).

Furthermore, a generic state space model for over-actuated mobile platforms has been presented, which can model platforms composed of several kinematic chains in a straightforward way. Nevertheless, this model is only valid for mobile platforms with open kinematic chains. Future work will improve the generic model to include closed-chain systems.

Thanks to the reduced computational cost of MWMP, this motion planner can be used in MM planetary exploration applications. Hence, MWMP has been applied to two different use cases, for which it has been integrated into two motion controllers. On the one hand, a tailored replanning controller, to tackle low-frequency disturbances with great smoothness in goal-constrained trajectory following problems. On the other hand, a receding horizon MPC, to cope with high-frequency disturbances in dynamic reactive problems. See Chapter 6 in the following for results of the planetary exploration experimental campaigns including MWMP and the controllers. Particularly, simulated and real experiments with MM rovers were performed, in two use cases: a sample tube MM retrieval and a rappelling MM support.

Chapter 6

Results

"Si aliquid bonum operibus assequeris, labor cito praeterit, sed bonum manet"
-
"If you accomplish something good with hard work, the labor passes quickly, but the good endures"

Gaius Musonius Rufus
1st century A.D.

6.1 Introduction

Although theoretically functional, all of the presented algorithms require experimental validation to confirm their usefulness: science demands proof. For autonomous robots, proof means bringing the algorithms to the platform, getting the system to an interesting scenario and use case, performing tests, and analyzing the results. This experimentation procedure demands multiple complex hardware and software components to work together, coordinately, before testing the new methods. Consequently, robotics experimental tests are generally expensive and time-consuming.

Roboticists have developed, thus, an alternative to ease experimentation: simulations with the software-in-the-loop concept. Software-in-the-loop consists of using the same software architecture as the real robot in the simulations, to check that the different components and their connections work correctly before bringing them to the platform. Since the software architecture is the same in simulations, transferring any modification to the real system is straightforward. Besides, simulation results are analyzed to improve the new methods before the real experimentation, and also to organize the most interesting experimental tests w.r.t. the validation, before actually going to the real site.

Following the software-in-the-loop concept, during this doctoral research multiple simulation, laboratory, and field tests have been performed to validate the algorithms presented in the previous chapters. Find a summary of all the tests in Table 6.1. Note that those tests marked with a "*" are not analyzed in this dissertation to

TABLE 6.1: Synopsis of experimental campaigns.

Contribution	Location	Platform	Tests		
			Simulation	Laboratory	Field
1	Decos (The Netherlands)	HDPR (ESA)	Yes	Yes*	Yes
2	Galopprennbahn Bremen (Germany)	SherpaTT (DFKI)	Yes	Yes*	Yes
3	ESTEC (The Netherlands)	ExoTeR (ESA)	Yes*	Yes	No
4	Lanzarote (Spain)	SherpaTT and CoyoteIII (DFKI)	Yes	Yes*	Yes

maintain the document's conciseness, since they were either too preliminary, without including all of the involved systems, or did not provide valuable results. Four main test campaigns have been performed, one for each main contribution of this doctoral dissertation. First, round-trip navigation with HDPR from ESA, emulating a sample fetching mission including the dynamic cost map update shown in Chapter 3. Second, an autonomous MM operation with the rover SherpaTT from DFKI, using the robust motion planner presented in Chapter 4 to cope with its highly restricted arm workspace. Third, an optimal retrieval of a Martian sample tube with ExoTeR in the ESA-PRL in ESTEC, using the optimal motion planner depicted in Chapter 5. Fourth, a rappelling campaign with the DFKI rovers SherpaTT and CoyoteIII in a Lunar analogue lava tube in Lanzarote, using again the optimal motion planner presented in Chapter 5 but integrated within an MPC loop. Let us show these results in more detail in the following.

6.2 Round-trip navigation using dynamic cost maps in planetary surfaces

The proposed method in Chapter 3, i.e. the dynamic cost map update procedure during the traverse, was validated by means of several experimental tests. The tests emulated a round-trip mission, where the rover HDPR had to autonomously reach a certain spot in the scenario, perform a task, and return to the starting point. The rover HDPR, developed by ESA-PRL, is characterized by having great autonomy and weight-carrying capacity (Hewitt et al., 2018). The selected scenario for the test campaign was Decos, a Mars-like terrain including rocks and craters, which is located close to ESTEC in Noordwijk, The Netherlands. First, simulation experiments were carried out in a virtual environment that emulates the rover dynamics and its interaction with terrains and obstacles. Second, a field test was performed to validate the integration of the new cost map update feature within the overall

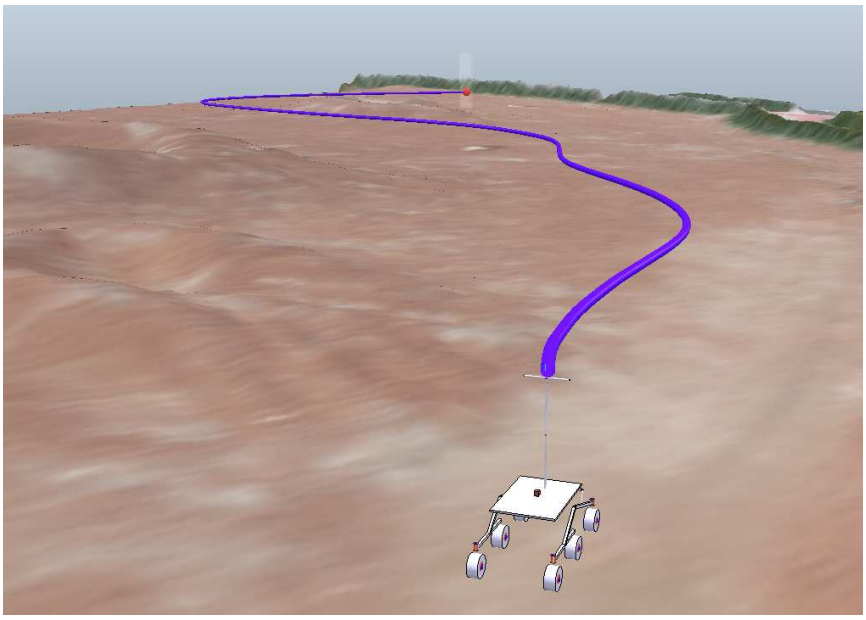


FIGURE 6.1: CoppeliaSim simulation environment that includes the virtual scenario of Decos, the model of HDPR, the goal (red ball), and the followed path (blue line).

GNC architecture that is being developed by ESA in collaboration with UMA-SRL (Azkarate et al., 2020). A video that illustrates both simulation and real tests was recorded[1]. The results of these tests show how the return traverse is improved due to the dynamic updating of the cost map, as follows.

Simulation campaign

As aforementioned, the software in the loop concept was used to perform several simulations. In this sense, the CoppeliaSim simulator was used together with a virtual machine that implemented HDPR's software architecture, i.e. all the software components used in the real rover from ESA. The virtual environment included several models. First, a model of Decos was integrated by including a DEM and textures of the real scenario, which were obtained using the eBee drone from senseFly Inc. Second, a simplified model of HDPR, which mimics its kinematic configuration and locomotion. Figure 6.1 shows an image of the simulation environment, with the rover following a predefined path.

The software used to control every different component in the GNC architecture of the rover was implemented within the Robot Construction Kit (ROCK)[2], which is a real-time operating system for robots created and maintained by DFKI RIC. The proposed cost update algorithm was developed in C++ within the DyMu path planner (Sánchez-Ibáñez et al., 2019b), which can be found in the UMA-SRL Github organization[3]. The communication between ROCK and CoppeliaSim was implemented

[1] https://youtu.be/e1Tqmn8HFvU
[2] https://www.rock-robotics.org/
[3] https://github.com/spaceuma/ARES-DyMu_cpp

via the remote Application Programming Interface (API) of the simulator, together with several components in the control software.

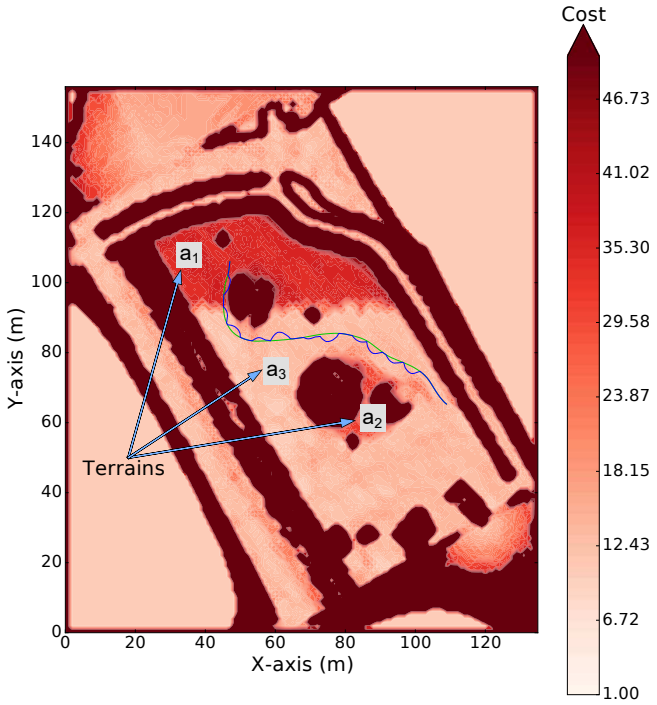
The objective of the simulation was to test the functioning of the proposed method for a round-trip mission, therefore, two paths were planned: an initial path to reach a certain localization in the map, and a second one to come back to the starting point. The initial cost map was computed according to a DEM of the terrain, assigning higher costs to big slopes and obstacles.

During the traverse, the rover gathered information about the environment that was used later to update the cost map before planning consecutive paths, e.g. to return to the starting point. To test the performance of the obstacle avoidance method, the rover encountered several obstacles not considered in the initial cost map. It was emulated by generating random obstacles during the simulation. Additionally, the proposed method for cost map updating based on terrain features was tested in the following way. Since the experimental terrain is composed of only one type of terrain, it is on purpose wrongly segmented into three terrains, as shown in Figure 6.2a. The first one (a_1) takes up the northern path of the scenario and was considered as loose soil with a high slip ratio ($\overline{\sigma_1} = 0.8$). The second terrain ($a_2$) occupies the surrounding area of the central craters, and it was also considered as a high slip terrain ($\overline{\sigma_2} = 0.75$). Finally, the third terrain (a_3) covers the rest of the scenario, without slip as a rough terrain ($\overline{\sigma_3} = 0$). The trafficability of the three terrains was initially estimated as one ($\overline{\xi_1}, \overline{\xi_2}, \overline{\xi_3} = 1$) since this feature has to be necessarily measured on the traverse.

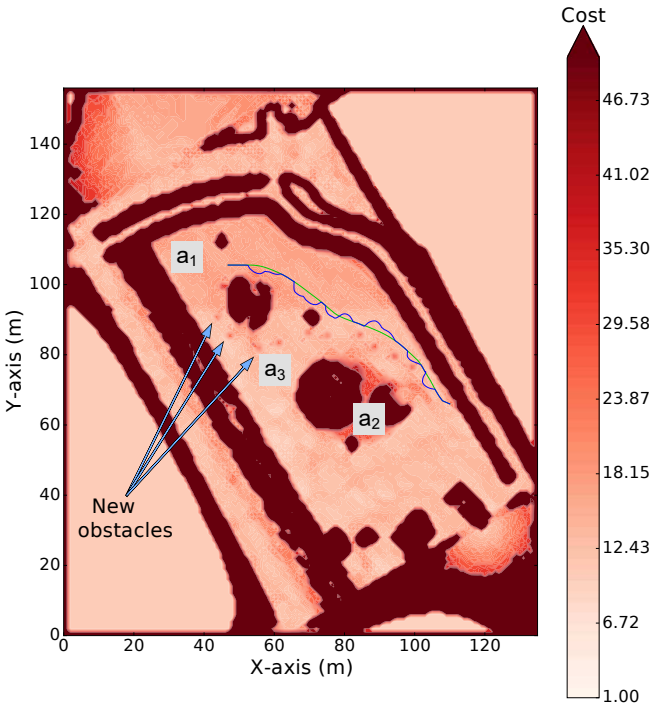
Regarding the aforementioned parameters, the initial cost of the terrains were $c_1 = 30$, $c_2 = 25$ and $c_3 = 10$. Due to that cost distribution, the initially planned path (green) took a longer way to reach the objective. This initial path was repaired continuously during the traverse (blue) due to the newfound hazards. As a result, the total length of the initial path was 90.13 m, while the repaired one was 99.06 m, and the rover spent 16 min 24 s to reach the objective at 10 cm/s speed.

When the return traverse was planned, the cost map was already updated, as exposed in Figure 6.2b. On one hand, the detected obstacles were included in the cost map; on the other hand, the updated costs of the terrains a_1 and a_3 concluded quite similar ($c_1 = 10.24, c_3 = 10$), since their measured slip ratio and trafficability were almost identical ($\overline{\sigma_1} = 0.03$, $\overline{\sigma_3} = 0.01$; $\overline{\xi_1} = 0.91$, $\overline{\xi_3} = 0.93$). The cost of the terrain a_2 remained the same since it had not been traversed. As a consequence, the return planned path (green line in Figure 6.2b) was shorter and avoided directly the already detected obstacles, with a total length of 77.84 m. Nevertheless, the rover still encountered new hazards along this new path, which needed to be repaired (blue). It increased its length to 88.22 m with a total spent time of 14 min 18 s at the same velocity.

In summary, a reduction of 13.64 % could be observed for the return path length during the simulation tests, i.e. the path planner was able to find a better path to return after updating the cost map using the gathered information. The real reduction,



(a) Initial segmentation of the terrain within the cost map, planned and traversed paths.



(b) Updated cost map for the return traverse.

FIGURE 6.2: Round-trip simulation results, showing the initial and final cost maps with the planned and traversed path. Both images show the segmented terrains and their cost (color), the first (a) and second (b) planned (green) and followed (blue) paths.

TABLE 6.2: Results of the HDPR round-trip simulation tests.

Parameter	Outbound traverse	Return traverse	Reduction
Planned distance	90.13 m	77.84 m	13.64 %
Traveled distance	99.06 m	88.22 m	10.94 %
Time spent	16 min 24 s	14 min 18 s	12.80 %

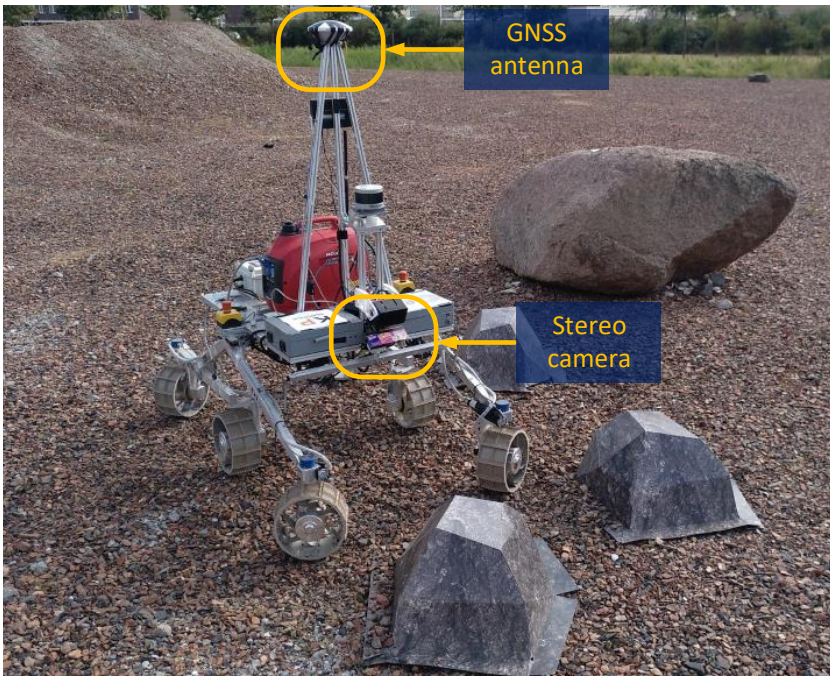


FIGURE 6.3: HDPR with the stereo cameras used for hazard avoidance and the GNSS antenna used for localization.

due to the later encountered obstacles, was 10.94 %. Regarding the time spent, there was a reduction of 12.80 %, saving 2 min 6 s during the return traverse thanks to the improvements in the cost map. Find the results of the simulation tests summed up in Table 6.2.

Field test campaign

Once the simulation confirmed the utility of the proposed methods, a field test was carried out to analyze their impact in an analogue round-trip planetary exploration mission. For this purpose, the real HDPR was equipped with a GNSS receiver for ground-truth localization, with complementary Real-Time Kinematic (RTK) corrections coming from a GNSS base station antenna. Besides, a hazard detector was included to identify local obstacles (Gerdes et al., 2020), which makes use of stereo cameras. Figure 6.3 illustrates the experimental setup for HDPR.

An aerial image of the scenario during the field test is shown in Figure 6.4, including the three patches of terrain, and dummy rocks that were placed randomly and not considered initially in the DEM. The objective of the mission was to reach

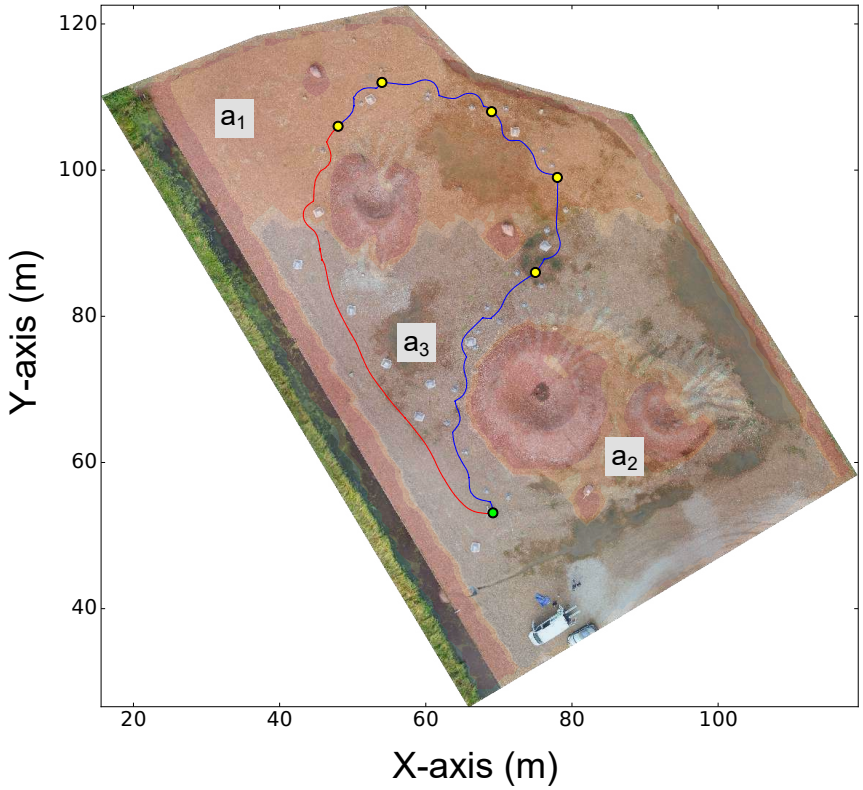


FIGURE 6.4: Aerial image of Decos and the trajectories covered by HDPR during the field test.

several key spots during the traverse, shown as yellow dots in Figure 6.4, and return to the starting point, shown as a green dot. Thus, a new path was planned each time a key spot was reached, and the cost map was also updated on the traverse. The first trajectory covered by HDPR during the field test is shown in blue, and the return trajectory is shown in red. In both paths, the hazard avoidance maneuvers can be noted. Additionally, the different terrain patches are labeled in Figure 6.4. The initial segmentation and parameters of the scenario were the same as the ones used in the simulations, hence the costs were also the same.

Analyzing the rover traverse until it reached the last key spot, the total length of the planned path on terrain a_1 was 43.19 m, and on terrain a_3 was 47.15 m. Therefore, the total planned distance was 90.34 m. Considering the initially estimated slip ratio of both terrains and a constant rover speed of 10 cm/s, the expected time needed to cover these distances was 7 min 52 s and 35 min 59 s respectively, with a total of 43 min 51 s. Nevertheless, the real performance of the rover differed significantly from what was expected. On one hand, the real distance covered by the rover was 101.45 m, which means an increase of 12.30 % (11.12 m), due to the encountered obstacles and the consequent hazard avoidance maneuvers. On the other hand, the real spent time to cover the paths was 29 min 54 s, which is 31.81 % (13 min 57 s) less than expected. This difference is caused by errors in the initial terrain parameters estimation since the real slip ratio of terrain a_1 was much lower. In summary, the initial cost map estimation was clearly not representative of the real situation of the

TABLE 6.3: Results of the HDPR round-trip field test campaign.

Parameter	Outbound traverse	Return traverse
Planned distance	90.34 m	62.19 m
Traveled distance	101.45 m	64.31 m
Absolute error	11.12 m	2.12 m
Percent error	12.30 %	3.41 %
Estimated time spent	43 min 51 s	10 min 30 s
Real time spent	29 min 54 s	12 min 22 s
Absolute error	13 min 57 s	1 min 52 s
Percent error	31.81 %	17.78 %

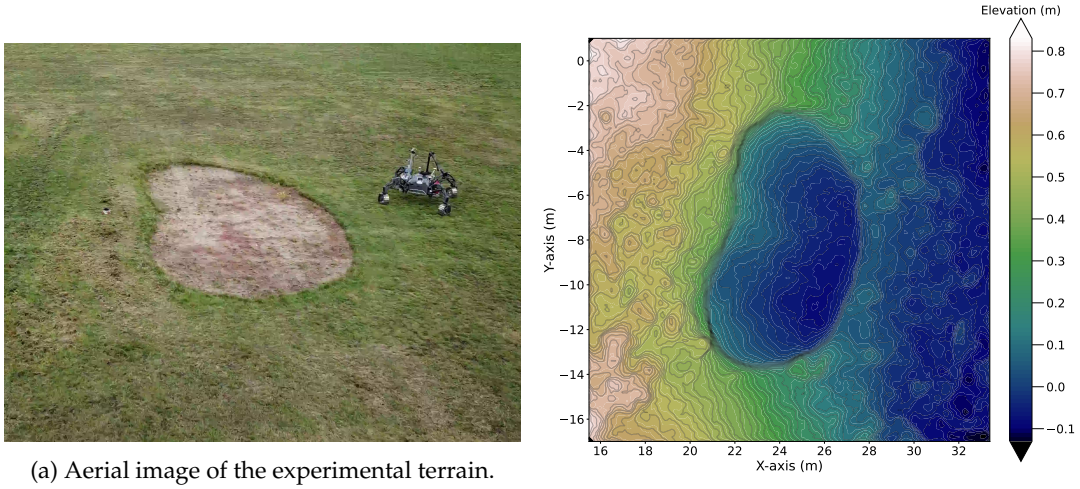
scenario, and therefore the final performance of the rover differed substantially from what was expected.

Before performing the return traverse, the cost map was updated with the already found information, i.e., slip ratio, detected obstacles, and trafficability. In consequence, the planned return path evaded the first detected obstacle close to the starting point. Additionally, as it happened in the simulations, the final costs of the terrains concluded much the same ($c_1 = 10, c_3 = 10.08$), showing that the measured slip ratio and trafficability of both terrains were almost equal ($\overline{\sigma_1} = 0.02$, $\overline{\sigma_3} = 0.01$; $\overline{\xi_1} = 0.90$, $\overline{\xi_3} = 0.88$). Once again, the terrain a_2 was not traversed hence its cost remained the same.

Finally, the total planned distance for the return path on terrain a_1 was 12.92 m, and on terrain a_3 was 49.27 m, with a total distance of 62.19 m. Considering the updated slip ratios, the expected time to cover these distances were 2 min 12 s and 8 min 18 s respectively, with a total of 10 min 30 s. Let us compare the aforementioned distances and times with the real ones, as summarized in Table 6.3. Firstly, the final distance covered by the rover was 64.31 m, i.e. 3.41 % (2.12 m) more than planned. Secondly, the total time spent during the return traverse was 12 min 22 s, which is 17.78 % (1 min 52 s) more than expected. This increase in both distance covered and time spent is caused by hazard avoidance maneuvers. Nevertheless, these errors are noticeably smaller than the ones committed during the first traverse. This improvement is thanks to the information feedback, which is raising the reliability of the cost map as regards the real situation of the scenario, i.e. the rover is more aware of the characteristics of the scenario, enhancing the consequent planned paths.

6.3 Robust goal-constrained MM in planetary exploration

Within the context of the European project ADE, as explained before in Chapter 1, a test campaign was performed with the SherpaTT rover from DFKI. Initially, the project field trials were planned in Fuerteventura, Canary Islands, Spain, but had to be disregarded due to the COVID-19 pandemic of early 2020. Consequently, several minor field test campaigns were carried out close to DFKI in Bremen, Germany,



(a) Aerial image of the experimental terrain.

(b) DEM.

FIGURE 6.5: Depiction of the experimental terrain at Galopprennbahn Bremen, Germany.

only with the team from DFKI RIC on-site, with the rest of the partners providing online support. Among the goals of ADE's tests was to perform an autonomous dummy MM operation, so SherpaTT could reach a particular spot of the scenario and perform a manipulation task, moving base and arm in a coordinated manner. This use case was the perfect application of the robust motion planner presented in Chapter 4, considering that SherpaTT was equipped with multiple scientific and engineering tools that considerably limited the free workspace of the robotic arm.

One of the aforementioned field test campaigns was in Galopprennbahn Bremen, a vast racecourse suitable for long rover traverses, which is shown in Figure 6.5a. Before these tests, the robust MM motion planner was integrated within SherpaTT's software stack and tested within a custom DFKI simulation environment with tailored models for SherpaTT and Galopprennbahn Bremen. To generate the model of the scenario, a DEM was obtained by aerial surveying of the area, which is shown in Figure 6.5b. Some of the simulation and field tests in Galopprennbahn are shown in a supplementary video[4]. Additionally, a benchmark with an off-the-shelf MM motion planner was performed, to analyze the advantages and disadvantages of the proposed method w.r.t. the state of art. Hence, the platform, the simulation tests, the benchmark, and the Galopprennbahn field test campaign are thoroughly described below.

Platform survey

The platform used within the robust MM simulation and field tests is SherpaTT. Shown in Figure 6.6, SherpaTT is a legs-on-wheels rover equipped with a 6 DoF manipulator, developed at DFKI RIC. Its hybrid locomotion enables walking and

[4] `https://youtu.be/I-cEbNgtQ9c`

FIGURE 6.6: SherpaTT rover in the Lunar crater of the SpaceHall at DFKI RIC, Bremen.

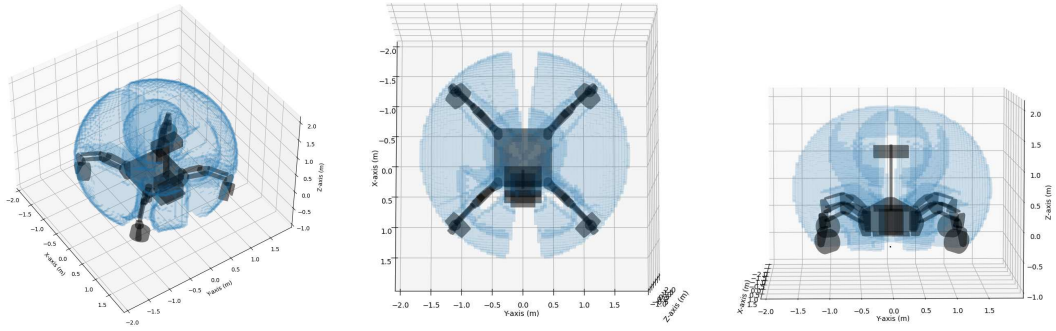
driving capabilities, i.e. an active suspension system that increases mobility on irregular terrains. Besides, the 6 DoF robotic arm holds an Electro-Mechanical Interface on its end effector, allowing the manipulator to retrieve different tools, placed in four mounts on the central body, to perform a wide variety of tasks. The rover has an Emlid Reach GPS with RTK corrections and two GNSS antennas for ground truth positioning, an IMU mainly used to estimate the rover 3D orientation, and a Velodyne HDL-32E Light Detection and Ranging (LIDAR) for mapping. Finally, the rover is also equipped with four NIR Basler acA2040-25gmNIR cameras, one stereo pair of Localization Camera (LocCam)s on top of an avionics box on the foremost side of the rover, and another stereo pair of Navigation Camera (NavCam)s on top of a PTU-E46 Pan & Tilt Unit from FLIR Systems, which is placed on a mast located also at the front of the rover. The cameras are used for perception purposes, i.e. NavCam for mapping and LocCam for localization, as the name suggests.

Concerning the robotic arm, its joint specifications are shown in Table 6.4. It is a 6 DoF manipulator with a Roll-Pitch-Pitch-Roll-Pitch-Roll configuration. The first three joints, also called position joints, are slower ($\leq 6\,^{\circ}/\mathrm{s}$) because of big gear ratios. On the other hand, the last three joints are the orientation joints that define the spherical wrist, weaker but faster and with a wide joint movement range. More details about the manipulator can be found in (Manz et al., 2012).

Regarding computational resources and software, a standard Intel-Core-i7 PC

TABLE 6.4: SherpaTT arm joints specifications.

Joint	1	2	3	4	5	6
Type	Rot.	Rot.	Rot.	Rot.	Rot.	Rot.
Orientation	Z	Y	Y	Z	Y	Z
Range (°)	+175 −175	+55 −120	+165 −100	+175 −175	+125 −125	+175 −175
Speed (°/s)	±3	±6	±6	±20	±20	±20
Gear ratio	7680:1	3000:1	3000:1	160:1	160:1	160:1
Repeatable peak torque (N m)	520	866	433	92	92	92



(a) General point of view (b) Top point of view (c) Front point of view

FIGURE 6.7: Shape of the reachability volume Ω_R of SherpaTT, where the safe and reachable volume for the wrist W_R is represented in light blue. The up elbow and right shoulder solution of the IK is used.

is installed onboard SherpaTT to run the Motion Control Subsystem (MCS), using the ROCK2 framework. Additionally, the SherpaTT API also runs on the onboard computer, which allows interchanging information between the robot and external software components, i.e. for receiving commands or sending sensor data. Besides, it is the same interface used to communicate with SherpaTT in the simulator, following the software-in-the-loop concept. The SherpaTT API makes use of the Robot Remote Control library (RRC), see (Danter et al., 2020) for more details.

Finally, the depicted characteristics of SherpaTT were analyzed for their inclusion into the motion planner, as follows. Making use of the URDF of the platform, the computed reachability volume of the SherpaTT's manipulator is shown in Figure 6.7, with the safe volume represented in light blue. Note that the wrist reachability volume was used in this paper, taking advantage of the already mentioned kinematic decoupling, seeing that the SherpaTT's robotic arm has a spherical wrist. Also note that the up elbow, right shoulder solution of the IK was used, since it keeps the arm elbow as far as possible from the body. It is remarkable that the presence of the legs, the avionics box, the mast, and the cameras restricted noticeably the available free-of-collision workspace of the robotic arm. In particular, a highly unsafe volume can be observed in the fore part of the rover due to the mast. This was a main restriction in the manipulator workspace that had to be considered when planning its movements. Besides, another non-reachable volume can be observed in the rear part

of the rover, due to position limits of the first arm joint.

Simulation campaign

The validation of the proposed motion planner was carried out, first, through simulation tests. They were performed using the DFKI Machina Arte Robotum Simulans (MARS) simulator[5], with 3D models of SherpaTT and Galopprennbahn Bremen[6], a racecourse located in Bremen near DFKI RIC. Both the model and the simulator characteristics are detailed in this section below.

The simulation tests consisted of reaching a particular goal sample in the scenario, to perform a dummy scientific task such as a scanning movement to analyze the surface. First, the motion planning library received the goal sample pose and the global DEM of the scenario, which was obtained by a commercial drone surveying the area beforehand; second, the planner processed the DEM to generate the traversability map and the cost map, to differentiate safe and unsafe areas; third, the cost map and the reachability volume of SherpaTT were used to generate a motion plan; fourth, the platform followed the motion plan by controlling both the base and the manipulator, i.e. sending commands depending on the current rover status. This motion planning procedure is performed only once, offline, before starting the motion, i.e. no replanning nor map update capabilities are included in this work. The motion planning library was launched in the same Central Processing Unit (CPU) as the simulator, communicating through the SherpaTT API, as aforementioned. The C++ source code of the motion planner is available in GitHub[7] under Massachusetts Institute of Technology (MIT) open source license.

On the one hand, the goal of the simulations was to confirm that the planned motions were suitable for the platform and able to be followed by the coupled controller. On the other hand, a deep analysis of the motion planner's performance was carried out to demonstrate the advantages of using a coupled approach. A simple decoupled motion planner and the proposed coupled approach were compared, considering different planner configurations by changing the deployment distance d (end deployment, progressive deployment, beginning deployment) and also changing the velocity of the joints of the manipulator relative to the real ones (x0.6, x0.8, x1 i.e. the real speed of the arm joints, x1.2 and x1.4). Note that the decoupled motion planner is just the proposed one but starting the arm motion only in the last waypoint of the rover trajectory, i.e. base and arm never move at the same time.

The simulation environment can be observed in Figure 6.8, where one of the mobile manipulation tests is shown. The SherpaTT 3D model was directly extracted by MARS from the robot URDF. Conversely, the terrain model was produced beforehand for the areas where the robotic tests were to be performed. A global, offline DEM of the environment was generated by commercial drone surveying hardware

[5] https://github.com/rock-simulation/mars

[6] https://goo.gl/maps/wJnPKLG2qY4QQnAt7

[7] https://github.com/spaceuma/ADE-Mobile_Manipulation

FIGURE 6.8: Simulation environment in MARS, with SherpaTT executing a coupled motion plan in the Galopprennbahn Bremen racecourse field.

and software. This approach, i.e. surveying the area beforehand, has various advantages when considering the final test deployment. First, the different reference frames can be set before arrival at the test site, including the map or the global frame, whose position and orientation can be set using the GPS coordinates. Once the integrated software has been tested in the simulated environment, no change in the reference frame definition is needed when setting the robot on the field. Second, the accuracy of the simulation can be evaluated after the rover tests, i.e. the simulated and real executions can be compared to try to improve the simulator parameters towards achieving better realism, although this is out of the scope of this work. The generated DEM was also used in simulations by the motion planner for the rover base path planning, as aforementioned.

Regarding the robotic simulator MARS, its software core is the Open Dynamics Engine (ODE) [8]. The physics engine uses optimization techniques to compute the contact points and forces to be applied to the simulated rigid body objects. Tuning of different parameters (e.g. material softness) and the robot models (e.g. simplifications through groupings) are needed to avoid instabilities and an acceptable simulation speed. MARS wraps the ODE with graphical tools, robotic sensors, and actuator models, as well as several plugins with a great variety of functions. In the presented research multiple plugins were employed, such as a plugin that manages the noise levels in sensors, or a plugin that emulates the real connection of the robotic framework to the sensor and motor drivers of the robot.

Multiple simulation experiments were conducted using the simulation environment presented above. For each motion planner configuration and arm joint speed, a total of 40 simulations were run on a single core of an Intel(R) Core(TM) i7-10750H

[8] `https://www.ode.org/`

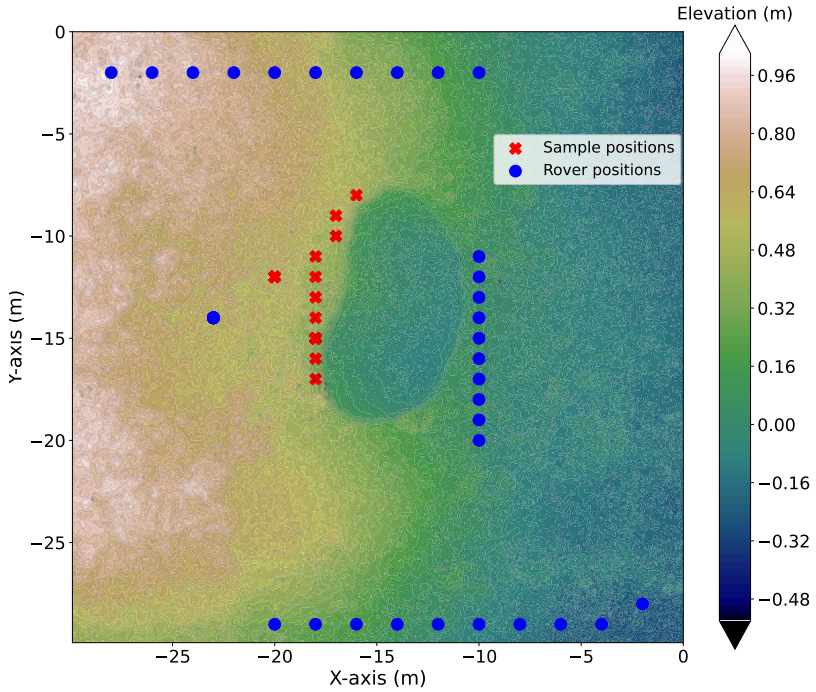


FIGURE 6.9: Rover initial positions (blue circles) and sample goal positions (red crosses) in the 40 simulation tests. Note that different combinations of starting pose and goal were used.

CPU (2.60GHz). At each simulation, the initial rover pose and the goal sample position were changed as shown in Figure 6.9, maintaining an average separation of approximately 15 m.

In the first place, Figure 6.10 shows the average computation time required to obtain the solution for every configuration of the motion planner. As can be observed, the decoupled solution is the fastest (2.85 s), since the computation is simpler. Nonetheless, the coupled approaches have comparable computation times (4.27 s progressive deployment, 5.12 s beginning deployment, 10.80 s end deployment), being slightly higher for the end deployment configuration (10.80 s) due to the bigger size of the 3D tunnel cost volume, which increases noticeably the number of nodes to be checked during the 3D trajectory planning phase.

The average distance to the closest self-collision during the motion is another remarkable result of the simulation tests, as a measure of the risk of self-collisions. Maintaining a high distance to self-collisions increases the manipulability of the robotic arm, and reduces the possibilities of self-colliding in case of a positioning error, e.g. due to a joint failure or de-calibration. As shown in Figure 6.11, the fact that the arm and base move at the same time does not increase the risk of self-collisions, which is similar in the decoupled solution (0.23 m) and in the coupled ones (0.26 m progressive deployment, 0.21 m beginning deployment, 0.34 m end deployment). On top of that, the end deployment configuration is considerably safer than the other ones (0.34 m), thanks to maintaining the arm in a comfortable and

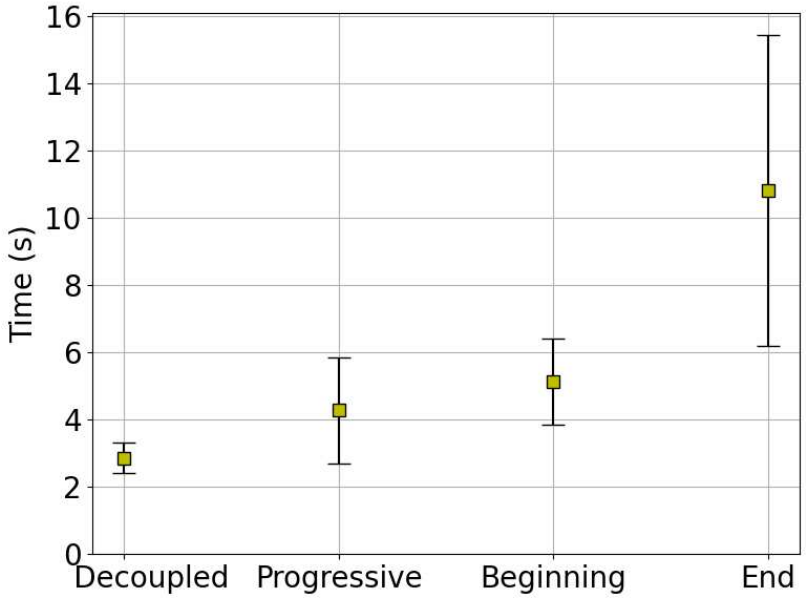


FIGURE 6.10: Motion plan computation time. For every configuration of the motion planner, 40 different simulations were carried out. Showing results for arm joints relative speed x1.

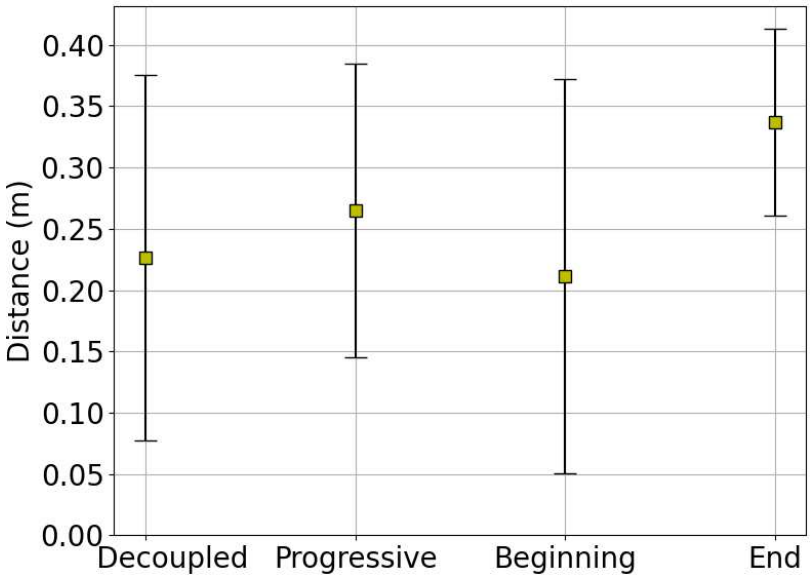


FIGURE 6.11: Average distance to closest unsafe wrist position. For every configuration of the motion planner, 40 different simulations were carried out. Showing results for arm joints relative speed x1.

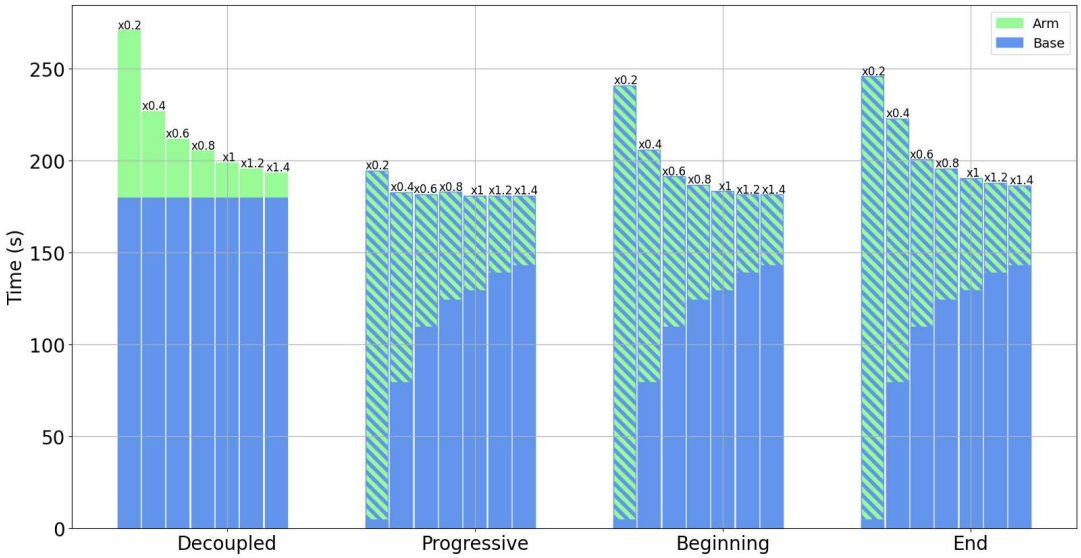


FIGURE 6.12: Average required execution time of the motion plan for the manipulator (green) and the rover base (blue). A blue and green hatch shows when both systems are moving coordinately, at the same time. 40 different simulations were carried out for different configurations of the motion planner (decoupled, progressive, beginning, end) and different manipulator relative speeds (x0.2, x0.4, x0.6, x0.8, x1, x1.2, x1.4).

safer configuration during the motion, deploying mainly afterward in the last waypoints of the trajectory.

As mentioned in Chapter 4, the main contribution of the proposed coupled motion planner is the increased efficiency in the motion plan execution, mainly in time terms, meanwhile ensuring the safety of the platform. This can be observed in Figure 6.12, which depicts the average required execution time of the solution for every configuration of the motion planner and speed of the arm joints, differentiating whenever the manipulator (green) or the base (blue) are moving. Clearly, the decoupled solution is the slowest, since the base movements are performed first (179.70 s on average), then the rover stops, and finally the arm moves. The decoupled results can be used as a reference for the worst possible motion in efficiency terms, with an average total required time 19.54 % bigger than the rover base required time. On the other hand, the coupled solutions coordinate both the base and arm movements, thus reducing noticeably the total execution time. In particular, the progressive deployment solution is the most efficient for every speed of the arm joints, actually with almost maximized efficiency since the total spent time is similar to the time required by the rover base to reach the sample vicinity, only 2.16 % bigger on average, i.e. the rover base does not need to wait or slow down for the arm to move. This is thanks to the fact that the arm joints' movement profile is the smoothest, distributing the arm joints' motion throughout the whole rover base trajectory. Only for speed x0.2 the required time is slightly higher (194.58 s), but this is due to the rover trajectory not being always long enough to give the arm time to move, since the arm joints are very slow. Anyhow, progressive deployment is still significantly faster

than the other configurations for speed x0.2, spending only 194.58 s in comparison to the 271.03 s of the decoupled solution, the 241.10 s of the beginning deployment or the 245.93 s of the end deployment.

Conversely, the end deployment solution is the slowest among the coupled solutions, since it resembles the decoupled approach, i.e. the arm mainly moves when the rover is already close to the goal. On average, the end deployment total required time is 13.72 % bigger than the rover base required time. The beginning deployment solution is also quite efficient with an average total required time 9.06 % bigger than the base one, although not as much as the progressive deployment since the main motion of the arm is performed at the beginning, which slows down the rover base when the arm starts to move. It is remarkable in Figure 6.12 how the beginning and end deployments noticeably increase their average execution times when the arm is slow since the arm joints profile is not properly distributed throughout the trajectory, which provokes the coupled controller to slow down the rover base motion quite often during the trajectory. This is not the case for the progressive deployment, which is completely unaffected by the arm joints' velocity as long as the rover trajectory is long enough for the arm to deploy completely.

The aforementioned simulation results demonstrate that the motion planner finds solutions with comparable computational cost to the decoupled solution, for all of its configurations. These solutions are equally safe, if not safer, than the standard decoupled ones, meanwhile noticeably increasing the mission efficiency in execution time terms by coordinating the arm and base motions. The progressive deployment is, conclusively, the best configuration for the motion planner, having maximized efficiency and maintaining sufficient distance to self-collisions during the motion, with the smallest computational cost among the coupled approaches. Remark that the slower the arm w.r.t. the rover base, the greater the importance of using a coupled arm-base motion with a progressive deployment, since the execution of the motion plan will be noticeably shorter. On the other hand, the faster the arm w.r.t. the rover base, the less the room for improvement left for the coupled approach, until it could slow down the motion considering the increased computational time w.r.t. the decoupled solution of the coupled motion planner. As a result, coupled arm-base motion planners, like the one proposed in this work, are recommended for systems with a slow manipulator speed w.r.t. the base movement speed. And, in particular, a progressive deployment is the best approach for a coupled arm-rover motion to maximize its efficiency.

Benchmark with an off-the-shelf motion planner

As a way to highlight the strengths and weaknesses of the presented motion planner, a comparison with an off-the-shelf mobile manipulation motion planner was performed. The selected motion planner was the one included within MoveIt[9], an

[9] `https://moveit.ros.org/`

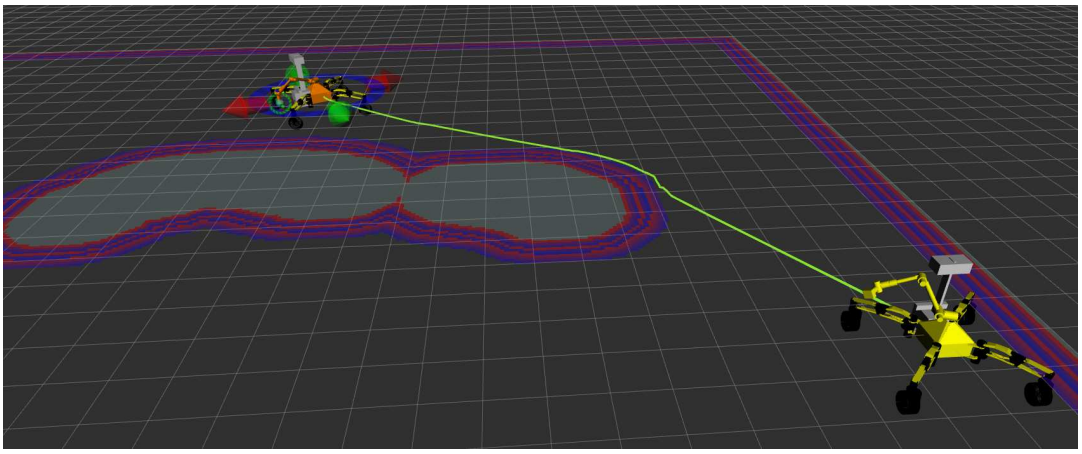


FIGURE 6.13: Motion plan generated by MoveIt for SherpaTT during one of the benchmark tests, with the planned path in green on top of the Galopprennbahn Bremen cost map.

open motion planning and control library mainly for robotic arms, used by a great variety of companies and research institutions worldwide. MoveIt was selected due to its popularity and its easiness of including new robots, thanks to the direct use of URDFs.

Remark that there are significant differences between our mobile manipulation motion planner and the one included within MoveIt. First, MoveIt does not generate a coupled motion plan for the base and arm, but rather a path for the mobile base and an independent motion plan for the manipulator, and afterward it executes both plans at the same time. Our motion planner considers both systems in a coupled manner during the motion planning stage, and also meanwhile executing the plan. Second, MoveIt does not include a Cartesian planner for mobile manipulation, but a path planner for the mobile platform based on Nav2[10] and a joint space planner for the manipulator. Therefore, MoveIt considers self-collisions of the manipulator with the robot itself, but not collisions with the environment. Third, the path planner does not guarantee the rover base will reach a position in which the sample is inside the manipulator workspace, thus additional maneuvers may be required. Fourth, MoveIt uses A* (Hart, Nilsson, and Raphael, 1968) for the mobile base path planning, and RRT Connect (Kuffner and La Valle, 2000) for the manipulator joint space motion planning. These are sub-optimal algorithms that do not always find the globally optimal solution. Conversely, FMM does find the globally optimal solution, maximizing the motion plan efficiency, which is the reason why it is used within the presented motion planner.

As before within the MARS simulation campaign, 40 simulation experiments were carried out using the simulation environment included within MoveIt, based on Gazebo. For a balanced comparison, the same cost map of Galopprennbahn Bremen generated in our simulations was directly fed to MoveIt. The 40 tests were run on a single core of an Intel(R) Core(TM) i7-10750H CPU (2.60GHz), using the same

[10] `https://navigation.ros.org/`

rover and goal poses as in the MARS simulations campaign. An example of the motion plan generated during one of the tests is shown in Figure 6.13, where the planned path for the mobile base, highlighted in green, avoided an obstacle on the Galopprennbahn Bremen cost map.

The simulations showed that the MoveIt motion planner is computationally fast, with an average planning time of 0.47 s in comparison to the 2.87 s of our progressive deployment planner. This is without considering the average 1.51 s required for the cost map generation (in total, the progressive deployment planner takes 4.38 s to compute the cost map and generate the motion plan), which is directly provided to MoveIt, and without any computational optimization of our planner code. The faster computation of MoveIt is also thanks to, as aforementioned, the generation of the manipulator motion plan in the joint space, without a Cartesian trajectory, and the use of sub-optimal planners. Nevertheless, the average distance from self-collisions of MoveIt is low in comparison to ours, 0.12 m vs 0.26 m, i.e. MoveIt motion plans get closer to possible self-collisions. Additionally, MoveIt does not consider collisions of the manipulator with the scenario when planning the movements of a mobile manipulator, as aforementioned. Therefore, it can not guarantee that the planned motion will not lead to a self-collision during its execution. Regarding the required time for execution of the motion plan, the comparison is not evenhanded, since MoveIt does not generate coupled motion plans but two independent ones executed at the same time, as stated above. Keeping that in mind, MoveIt requires slightly more time to execute the motion plan, on average 186.53 s vs. 181.05 s of the progressive deployment, due to the lower quality of the base path. Both planners have maximized manipulation efficiency, i.e. the whole manipulator motion is performed meanwhile the base is reaching the sample.

This comparison with an off-the-shelf motion planner shows, in conclusion, that the presented motion planner generates safer motion plans, with guaranteed avoidance of self-collisions, with a globally optimal path for the mobile base and maximized efficiency of the manipulator plan execution. The slightly higher computational cost of the proposed method is a drawback, which can be solved by optimizing the motion planner's current implementation, or by running some of the planning tasks in parallel to accelerate the computations, which would require to switch from FMM to a similar path planner such as FSM (Zhao, 2004).

Field test campaign and lessons learned

An ultimate validation of the presented algorithm was carried out through a field test campaign, integrating the motion planner into a real platform. The campaign consisted of several analogue tests to the simulations, with the real SherpaTT in Galopprennbahn Bremen as shown in Figure 6.14, executing the planned motions to reach different samples and performing a dummy scientific analysis on them. Due to time constraints within the field test campaign, only qualitative results could be extracted from the tests, which are presented in the following as lessons learned.

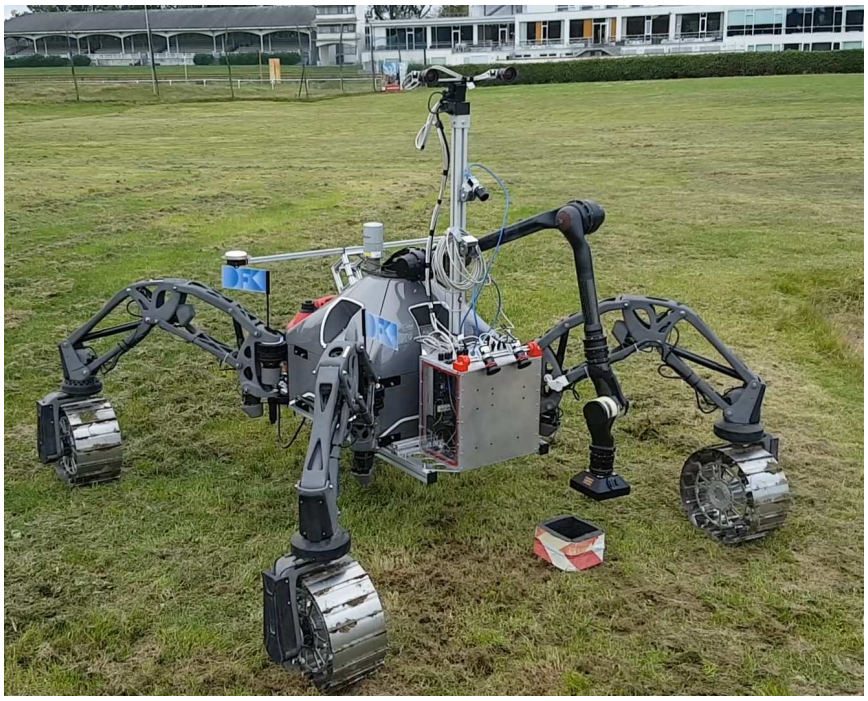
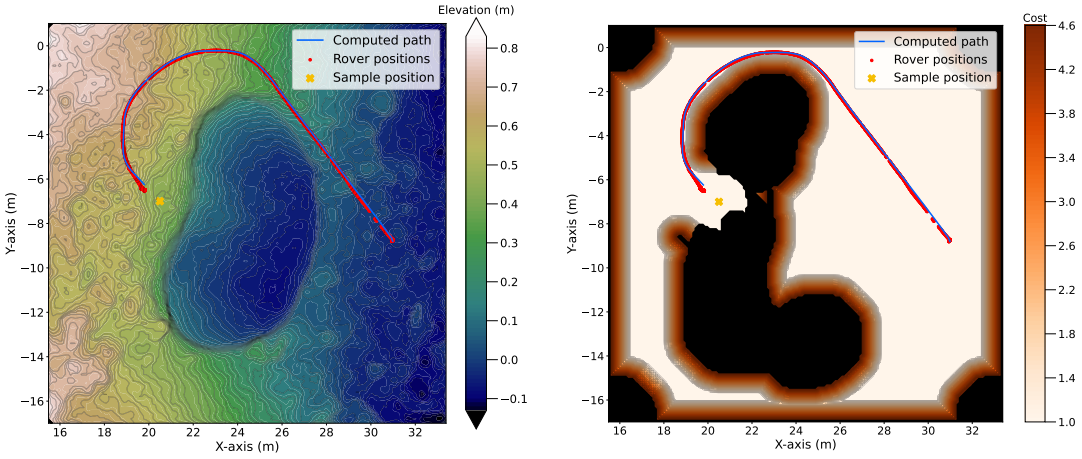


FIGURE 6.14: SherpaTT during the field test campaign in Galopprennbahn Bremen. The rover is equipped with a robotic arm, NavCams with a Pan and Tilt Unit on top of the mast, LocCams on the avionics box, two Global Navigation Satellite System antennas, and a Velodyne LIDAR.

Several sensors were required within the field test campaign. For localization, a Visual Odometry algorithm was used, merging the information received by the LocCams, the wheel odometry, and the Inertial Measurement Unit. Additionally, a GPS with RTK corrections was used to get the initial pose and log the ground-truth positions. The cameras were calibrated before the tests using standard distortion estimation methods. Lastly, the rover joints were manually re-calibrated in case of a significant mismatch between the position given by the proprioceptive sensors and the real ones.

The field tests proceeded similarly to the simulations, as follows. The motion planning library was launched in a separate Intel-Core-i7 CPU running Ubuntu 18.04, which communicated to the robot through the SherpaTT API using wireless communication (WiFi). Then, the motion planning library received the DEM of the scenario and the goal pose, always located inside the DEM, to generate the motion plan. Lastly, the library controlled the platform (rover base and manipulator) to properly follow the generated motion plan, reading the rover pose and the joint states and commanding the base motion commands as well as the arm joints' position and velocity. As in the simulations, no updates of the map or replans of the motion were performed, i.e. the motion plan was generated once, offline, and thoroughly followed. Whenever the goal was reached, the library launched the dummy scanning measurement, and the execution was finished.

The results of one of the field tests are shown in Figure 6.15. The base path



(a) Planned path and rover covered positions on top of the DEM.

(b) Planned path and rover covered positions on top of the cost map.

FIGURE 6.15: Results of one of the MM tests with SherpaTT at Galopprennbahn Bremen, Germany.

planner avoided the central pit in the scenario, as depicted in Figure 6.15a since the planner considered the left limits of the pit as non-traversable, as can be observed in the cost map of Figure 6.15b. Later on, the rover and manipulator were able to accurately follow the planned trajectory until the last stretch, where the LSC was activated to leave the sample within the manipulator workspace and the manipulator performed the scanning task, as shown in the appended video[11].

During the field tests, several issues appeared that led to some lessons learned. On the one hand, the arm presented some repeatability inaccuracies in the position of the end effector. This was caused by a small dead zone in the first joint and some occasional calibration errors on the joints. Although this was partially solved with a re-calibration of the arm before every test, the mismatch between the robot model and the real system added some uncertainty w.r.t. the safety of the motion plans to be executed. This danger was enlarged considering the highly restricted manipulator workspace on SherpaTT, i.e. the tight space between the legs, the mast, and the avionic box, since any small error could lead to a collision. Therefore, it was decided to be more conservative, i.e. maximize the security of the platform during the field tests. The motion planner was configured to use the beginning deployment since its motions were the most repetitive (thus predictable) and it maintained the arm far enough from collisions. Besides, the goal pose of the end effector above the sample was modified to be farther from the rover body and the avionics box.

On the other hand, an occasional lack of accuracy was observed in the position of the end effector w.r.t. the sample. This had two main causes. In the first place, the already mentioned positioning error of the manipulator. In the second place, small errors in the localization of the rover (fluctuations of the initial GPS position and the accumulated drift of the Visual Odometry localization) and errors in the position

[11] See footnote 4.

of the sample w.r.t. the actual commanded goal. This could not be solved during the field test campaign, since it required a separate component to detect the sample online and estimate its position, to feedback continuously the sample position to the motion planning library and replan the motion, thus, removing the positioning errors.

Conclusively, even with the aforementioned issues, the motion planning library was correctly deployed, tested, and verified in the real rover SherpaTT during the field test campaign. The platform autonomously generated a motion plan to reach a goal sample, and correctly followed the generated trajectory meanwhile the arm was deploying, finally leaving the sample within the reachability area of the manipulator and performing successfully the scanning measurement on the sample.

6.4 Optimal retrieval of Martian sample tubes

A sample tube retrieving rover is a highly over-actuated mobile platform, composed of a mobile base with multiple actuators (mainly driving and steering joints), and a robotic arm with several DoF. Considering the energy and time efficiency requirements of a planetary exploration mission, a Martian sample tube retrieval is the perfect use case to demonstrate the advantages of the optimal motion planning methodology proposed in Chapter 5. On the one hand, a deep performance analysis of the motion planner was carried out with a benchmark between different layouts of the approach, i.e. using different combinations of the already explained stages. This comparison confirmed that the proposed warm-start sequence is the most convenient, with a path planning warm start, a first unconstrained stage, and a final constrained stage.

On the other hand, several laboratory tests were performed with ExoTeR in the Martian Analogue Testbed of the ESA-PRL. Using MWMP and the proposed replanning procedure, ExoTeR was capable of successfully reaching a Martian sample tube and retrieving it with its manipulator. The source code of the MWMP library used within these tests is available in MatLab[12] and C++[13], under MIT open source license.

Below, prior to showing the test results, a complete description of the platform characteristics, the developed state-space model of the mobile manipulator, the motion controller, the experimental setup, the scenario, and the motion planner configuration are presented. Later on, the performance benchmark and the laboratory tests are thoroughly analyzed.

Mobile platform description

Within the research and development carried out at the ESA-PRL, the design and testing of planetary rover testbeds stands out. This is the case of ExoTeR (Azkarate et

[12] `https://github.com/spaceuma/MWMP-MatLab`
[13] `https://github.com/spaceuma/MWMP-Cpp`

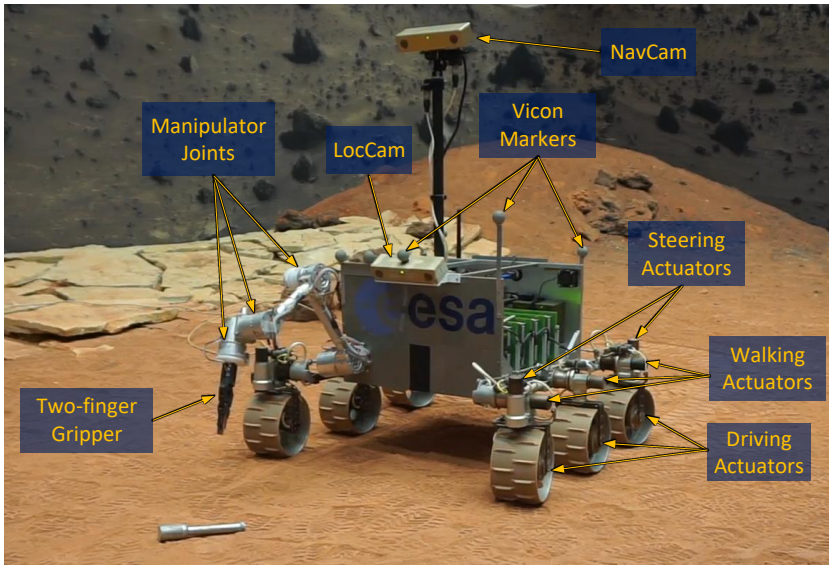


FIGURE 6.16: Detail of the experimental setup with ExoTeR approaching a sample tube, including its actuators (driving, walking, steering and manipulator's joints, two-fingers gripper) and its exteroceptive sensors (LocCam, NavCam). Besides, ExoTeR is equipped with an IMU to estimate its orientation and several Vicon Markers to precisely locate it inside the Martian Analogue Testbed of the ESA-PRL at ESTEC.

al., 2022), which conceptually mimics the early model of the Rosalind Franklin ExoMars rover, with a scaled-down concept. ExoTeR is a triple-bogie, double-Ackermann rover, with a locomotion system of 6 x 6 x 4 + 6. This means 6 wheels with 6 driving actuators, 4 of them steerable (the front and rear ones), which permits double-Ackermann steering or spot turns. Additionally, all 6 wheels include a walking actuator, as depicted in Figure 6.16, where ExoTeR is shown at the Martian Analogue Testbed at ESA-PRL. ExoTeR is also equipped with a five DoF manipulator, called MA5-E. Its five joints are rotational, with a Roll-Pitch-Pitch-Pitch-Roll configuration, being the first joint placed looking toward the movement direction of the platform. Its end effector has attached a two-fingered gripper for sample retrieval purposes. For localization and perception, ExoTeR has two stereo cameras, a close-range LocCam, and a long-range NavCam. Finally, ExoTeR has also an IMU for sensing the platform 3D orientation and has appended several Vicon markers for ground-truth localization inside the Martian testbed.

Focusing first on the mobile platform base, ExoTeR is a double-Ackermann rover with steering joints at the front and rear wheels. The central wheels do not steer, which implies that the rover can not move in every direction depending on the system orientation, i.e. non-holonomic constraints. This is a significant non-linearity, which is tackled inside the system model. The platform minimum turn radius is 0.6 m, due to the geometric distribution of the wheels and the range limit of $\pm 50^{\circ}$ in the steering joints, although it can perform point turns (change its orientation with zero linear velocity). Finally, its nominal translational speed is 5 cm/s, with a

TABLE 6.5: MA5-E joints characteristics.

Joint	1	2	3	4	5
Type	Rot.	Rot.	Rot.	Rot.	Rot.
Orientation	Z	Y	Y	Y	Z
Range (°)	±45	±170	±170	±170	±170
Speed (°/s)	0.57	0.57	0.57	0.57	0.57
Power (W)	0.75	0.75	0.75	0.75	0.75
Gear ratio	83200:1	83200:1	83200:1	83200:1	83200:1
Efficiency	0.5	0.5	0.5	0.5	0.5

2.85 N m maximum torque of the driving actuators.

Regarding the robotic arm, MA5-E, its main characteristics are outlined in Table 6.5. As can be observed, MA5-E joints have huge gear ratios, which allows the manipulator to handle heavy payloads, 2 kg, in comparison to the arm weight, 2.4 kg, and considering the joint motor's power (6 W). The dynamic effect of external disturbances is consequently negligible, i.e. gravity and the rover base movements. Nevertheless, the gears also imply an important drawback: the joints move very slowly, with a maximum rotational speed of 0.57 °/s.

The fully extended arm length is 0.527 m, with an additional 0.14 m taking into account the gripper. The arms end effector reachability is restricted by each joint position limit, as can be observed in Table 6.5. Doubtlessly, the arm movements are also limited by the rover body itself and the ground. Regarding the end effector, it can not reach any orientation because of the limitation of the five DoF configuration. This issue hinders any manipulation task, especially a sample retrieval operation, since the end effector can not always approach the sample completely perpendicular to the ground, with the appropriate gripper yaw w.r.t. the sample tube.

Note that ExoTeR has a series of system constraints, as explained above, which raise the necessity of using an optimal planner like MWMP. First, a coupled arm-base motion solves the arm joints' velocity issues, generating optimal motions where the arm is already prepared to perform the desired task once the base has reached the objective. Second, the rover DoF can be used to place the manipulator in a certain manner to effectively retrieve the sample, i.e. aligning the manipulator's first joint and the sample tube, performing completely perpendicular retrieval operations. This is achieved by properly tuning the costs associated with the goal pose of the end effector, including its orientation, as later clarified.

Double-Ackermann mobile manipulator model

As a mobile manipulator, ExoTeR is modeled with two different kinematic chains: the full-Ackermann mobile base and the robotic arm. The dynamic coupling between them is ignored, seeing that the movements of both the platform and the manipulator are very slow, generating negligible dynamic effects between them.

Additionally, the effect of gravity on the manipulator's joints is also disregarded, considering the huge gear ratios as aforementioned.

Following the generic state space model explained before, the state vector $x(n)$ for ExoTeR is defined in (6.1).

$$x(n) = \left[{}^{w}P_1 \; {}^{w}\dot{P}_1 \; {}^{w}P_2 \; {}^{1}P_2 \; {}^{1}\dot{P}_2 \; q_1 \; \dot{q}_1 \; \ddot{q}_1 \; q_2 \; \dot{q}_2 \; \ddot{q}_2 \right]^T \tag{6.1}$$

Where the kinematic chain 1 represents the mobile base and 2 the manipulator, thus, q_1 corresponds to the mobile base joints, i.e. the wheel driving θ_d and steering θ_s joints, and q_2 corresponds to the manipulator joints θ_m, which are all rotational as aforementioned. As a result, the state transition matrix $A(n)$ is extracted straightforwardly from the generic one, but specifically for a platform with two kinematic chains. In particular, I_1, V_1 and ${}^{w}\mathcal{J}_1$ refer to the inertia, Coriolis/centrifugal, and Jacobian matrices of the full-Ackermann non-holonomic mobile base, as well as I_2, V_2 and ${}^{1}\mathcal{J}_2$ refer to the five DoF manipulator. Remark that, as aforementioned, the mobile base Jacobian ${}^{w}\mathcal{J}_1$ is linearized employing a TSL considering the notable non-linearity that appears due to the non-holonomic constraints.

The actuators of the system are the six driving and four steering joints of the wheels of the mobile base and the five rotational joints of the manipulator. As external disturbances, gravity acts on the mobile platform as a constant acceleration. Thus, the actuation vector $u(n)$ is defined in 6.2.

$$u(n) = \left[\tau_d \; \tau_s \; \tau_m \; g \right]^T \tag{6.2}$$

Where τ_d, τ_s and τ_m are the actuation torques to the driving, steering, and manipulator joints respectively, and g is the gravity acceleration. Note that ExoTeR joints only receive position and velocity commands, nevertheless, using the whole dynamics model allows to generate torque-efficient motions. Later on, the joint position and speed commands are directly extracted from the state vector $x(n)$.

Once more, the input distribution matrix $B(n)$ is directly obtained using the generic one presented in Chapter 5.2 but with only one external disturbance, the gravity g. On one hand, the effect of g into the mobile base, γ_1^1, generates wheel-soil friction, which is modeled in a simplified way through the rolling resistance of the terrain as expressed in (6.3), with ρ the rolling resistance coefficient of the terrain, d_w the diameter of the wheels, m the mass of the vehicle and N_w the number of wheels of the rover. On the other hand, the effect of g into the manipulator, γ_1^2, is ignored as aforementioned, considering the huge gear ratio of the arm joints.

$$\gamma_1^1 = \rho \, \frac{d_w}{2} \, \frac{m}{N_w} \tag{6.3}$$

Finally, several constraints have been defined to consider ExoTeR limits. On the one hand, the maximum actuation torque for the driving (τ_d), steering (τ_s), and manipulator (τ_m) joints are included as state-input constraints. On the other hand, the limits on the velocity and acceleration of the driving ($\dot{\theta}_d$, $\ddot{\theta}_d$), the steering ($\dot{\theta}_s$, $\ddot{\theta}_s$) and

the manipulator ($\dot{\theta}_m$, $\ddot{\theta}_m$) joints are defined as pure state constraints. Additionally, the position limits of the steering θ_s and manipulator θ_m joints are also included as pure state constraints.

Error-based replanning methodology

Given a feasible motion plan, i.e. the state $x(n)$ and actuation $u(n)$ vectors for the complete planning horizon $\mathbb{T}$, a separate component, the motion controller, needs to bring it to the mobile platform, ensuring it is properly followed until reaching the goal. If there are deviations from what was planned, then this component has to make the right decisions to ensure that the goal is reached. These deviations can be caused by different means as previously stated in Chapter 2.4, like the model intrinsic errors because of the discretization and the linearization. But it is also pertinent to consider the effect of other agents on the system, such as external disturbances not considered initially in the model, e.g. the platform localization error, the goal pose estimation error, or the non-ideal behavior of the actuators. Additionally, the motion controller needs to prioritize different sub-objectives depending on the use case, such as robustness and predictability in case of a long-term trajectory following, or in fast convergence and energy efficiency in reactive tasks.

Focusing on the sample tube retrieval use case, the motion planner has generated a mid- or long-term trajectory for the robot to follow to reach a goal sample and retrieve it with the manipulator. In such a case, the system is not affected by high-frequency disturbances. Conversely, such an autonomous navigation rover can suffer disturbances such as new local obstacles, localization drift, or errors in the system joints. These inconveniences add up to the target positioning error, induced by the sample detection and localization subsystem. The sample positioning error is expected to be higher as farther the rover is from the sample, raising the necessity of replanning the motion as the positioning error gets smaller, i.e. as the system gets closer to the goal.

As a result, the motion-following algorithm does not need to be fast, but robust and predictable to regenerate the motion only in the particular cases when it is needed, what can be called an error-based replanning methodology. Thus, during the sample retrieval tests a replanning capability was used, similar to the Event-triggered one proposed in (Luis, Vukosavljev, and Schoellig, 2020), as follows. First, the motion plan follower sends sequentially the next actuation command to the platform (or the first one initially). Second, it checks if the goal pose has changed. If this is the case, then the system replans the whole motion. If not, a third step checks if there is too much drift in any of the controlled states. This would lead to returning to the initial stages of the motion planner, using the current platform pose to replan the motion. Fourth, if no replan is needed and the goal is reached, the follower finishes the execution. Otherwise, it continues sending actuation commands in accordance with the already generated motion plan and starts again the sequence.

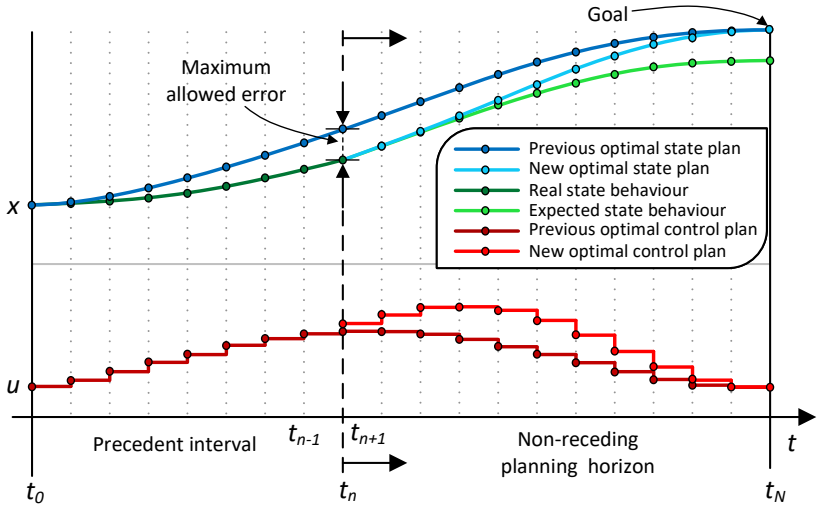
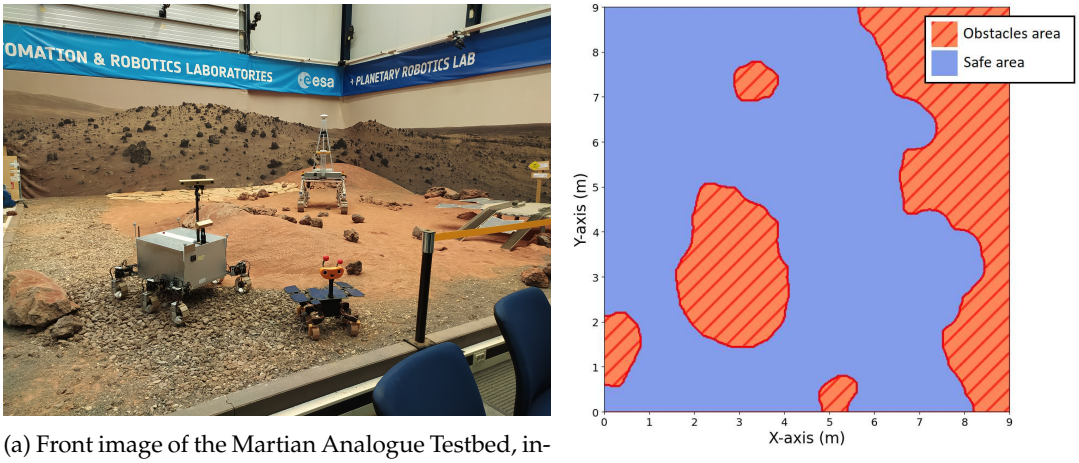


FIGURE 6.17: Graph exemplifying the replanning capability. The behavior of a controlled state $x(n)$ is continuously checked (dark green). If a considerable deviation from the previous plan (dark blue) is detected, a new global motion plan (light blue) is computed from the time step t_n onwards. The new state and actuation plans (light red) allow the system to smoothly reach the goal, correcting the previously accumulated error.

In case one of the states drifts from the planned motion, the behavior of the replanning capability is exemplified in Figure 6.17. Starting with a planned motion (dark blue) from t_0 to t_N, with N number of time steps of Δt size, and given the time evolution of a controlled state $x(n)$ (dark green) in accordance to a given actuation plan $u(n)$ (dark red), this evolution may differ from the plan, increasingly accumulating error. When this error surpasses a certain threshold at time step t_n, the predicted behavior of $x(n)$ is completely undesired (light green), thus, a replan is launched using the previous motion plan, from t_n onwards (dark blue), as a warm start. Thus, the replanned motion (light blue) compensates the accumulated drift in $x(n)$ by slightly modifying the previous optimal actuation plan $u(n)$ (dark red), generating a new one (light red) in the neighborhood of the previous solution. In this way, the state $x(n)$ will still reach the goal as long as the new motion plan is properly followed.

Experimental setup

The goal of the performed laboratory tests was to demonstrate that ExoTeR can reach and retrieve a Martian sample tube in a completely autonomous way. The tests were carried out in the Martian Analogue Testbed at the ESA-PRL, which can be observed in Figure 6.18a. This is a 9×9 m experimental terrain that is highly representative of a real Martian environment, including different types of soil (sandy, rocky), rocks, or small slopes. After processing a 2 cm resolution DEM of the ESA-PRL, a traversability map was generated, which is depicted in Figure 6.18b. This traversability map considers obstacles, slopes, and roughness, and is used to generate the cost map Ω. For more information about the cost map generation see Chapter 3.3.



(a) Front image of the Martian Analogue Testbed, including some of the lab's rovers.

(b) Traversability map.

FIGURE 6.18: Depiction of the Martian Analogue Testbed of the ESA-PRL at ESA-ESTEC, Noordwijk, The Netherlands.

The tests include real Martian autonomous navigation restrictions to perform an illustrative emulation of a sample tube retrieval mission. Therefore, two additional subsystems were integrated into the platform, apart from the presented MWMP and replanning algorithms. First, an autonomous sample detection and localization subsystem based on a Convolutional Neural Network (CNN), which uses the LocCam stereo images to locate the sample tube with an average under 5 cm position and 5° orientation errors (Castilla-Arquillo et al., 2022). Second, a visual odometry algorithm for the platform localization, using the LocCam stereo camera and the IMU, with 7.50 % average localization drift in position and less than 2° orientation error (Geiger, Ziegler, and Stiller, 2011). The Vicon markers were also used to obtain the ground-truth localization, not online but for data logging and post-processing purposes.

It was necessary to properly configure the motion planner for the tests. The test's total time t_N was 160 s, with a time step Δt of 0.8 s. For the motion planner to converge, the maximum allowed position error was set up to 1 cm and the orientation error to 10°. It was considered that the algorithm had converged if the norm of the stepped actuation $\bar{u}(n)$ was lower than 1 % of the norm of the whole actuation vector $u(n)$. The particular costs that configured the LQR cost matrices $Q(n), R(n)$ are defined in Table 6.6. Note that the cost of modifying the input gravity disturbance g in $R(n)$ is the largest, to ensure that it remains as a constant gravity acceleration of 9.81 m/s^2 precisely following the reference u^0. This gravity disturbance is ignored by the manipulator, as aforementioned. Additionally, to ensure that the sample was retrieved perpendicularly to the ground, the goal pose orientation ${}^w\phi_2(N)$ was filled with roll ${}^w\varphi_2(N) = 0$, and the pitch ${}^w\vartheta_2(N)$ and yaw ${}^w\psi_2(N)$ were computed depending on the estimated sample orientation.

Lastly, the replanning was launched if the platform deviated in certain aspects when following the planned motions. In the first place, if the platform drifted more than 4 cm from the planned path. In the second place, if any of the controlled joints

TABLE 6.6: Quadratic costs configuration.

Type	Variable	Cost
Goal state $Q(N)$	EE pose ${}^{w}P_2$	10^{11}
	Platform speed ${}^{w}\dot{P}_1$	10^{6}
	End effector speed ${}^{1}\dot{P}_2$	10^{6}
State full motion $Q(n)$	Platform pose ${}^{w}P_1$	20
	Driving wheels speed $\dot{\theta}_d$	100
	Driving wheels acc. $\ddot{\theta}_d$	10^{4}
	Arm joints speed $\dot{\theta}_m$	$3 \cdot 10^{5}$
	Arm joints acceleration $\ddot{\theta}_m$	$3 \cdot 10^{5}$
Input full motion $R(n)$	Wheels driving torque τ_d	10^{5}
	Steering joints torque τ_s	$8 \cdot 10^{4}$
	Arm joints torque τ_m	10^{11}
	Gravity g	10^{15}

deviated more than 2.29° from the plan. Additionally, every time the goal sample pose differed more than 3 cm or 17.19° from the previous estimation, a complete motion replan was launched.

Performance evaluation

The performance of the proposed motion planning approach was analyzed using the aforementioned use case and setup, to showcase its advantages w.r.t. any possible layout of the stages, for instance, cold-started or single-staged versions of the motion planner. For that purpose, six different layouts of the motion planner were defined, which include every possible combination of the three stages (USLQ, PPWS + USLQ, CSLQ, PPWS + CSLQ, USLQ + CSLQ) and the complete approach (PPWS + USLQ + CSLQ = MWMP). For every layout, the same 21 motion plans were launched, using the ESA-PRL scenario with the ExoTeR model and the sample tube retrieval use case, changing the initial rover pose and the goal sample pose.

Three main parameters were measured within the tests. First, the success rate, as a percentage. This represents the ratio of finding a successful motion plan in the 21 tests, i.e. when the algorithm converges in less than 100 iterations, which is shown in Figure 6.19. Second, the feasibility rate, which represents the percentage of constraint-compliant motion plans in the 21 tests, which is also shown in Figure 6.19. Remark that the feasibility rate encompasses the success rate, given that a motion plan can only be feasible if the solver converges, i.e. if the motion plan is successful. Hence, every layout that makes use of the CSLQ stage has equal success and feasibility rates, since the algorithm only converges if the constraints are fulfilled. Third, the average number of iterations until convergence. If several stages are established, then the total number of iterations is used. The extracted results regarding the number of iterations are shown in Figure 6.20. Note that the number of iterations is employed in the following as a measure of the convergence speed,

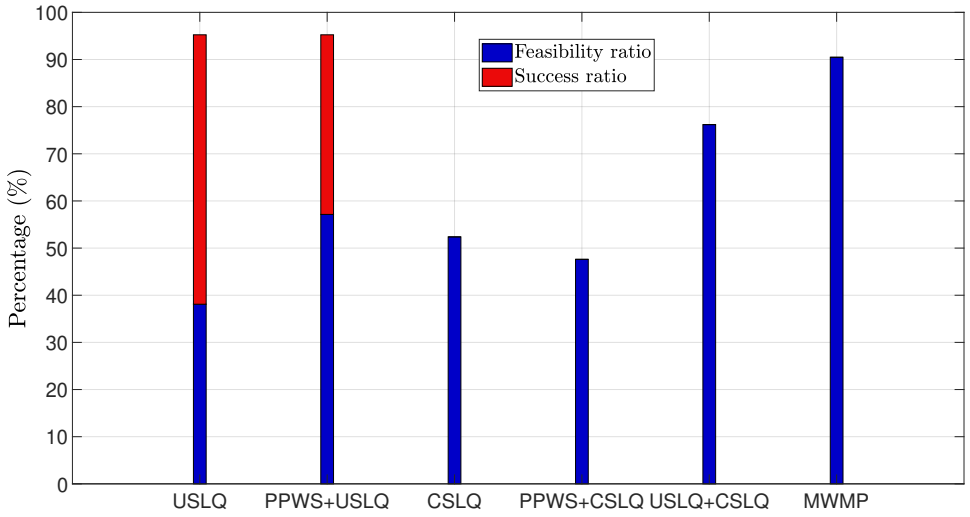


FIGURE 6.19: Measured success and feasibility ratios on the performance tests. For every layout, 21 different motion plans were launched. Note that the layouts that include the constrained stage (CSLQ) have the same percentage of successful and feasible motion plans since this stage ensures feasibility if the algorithm converges.

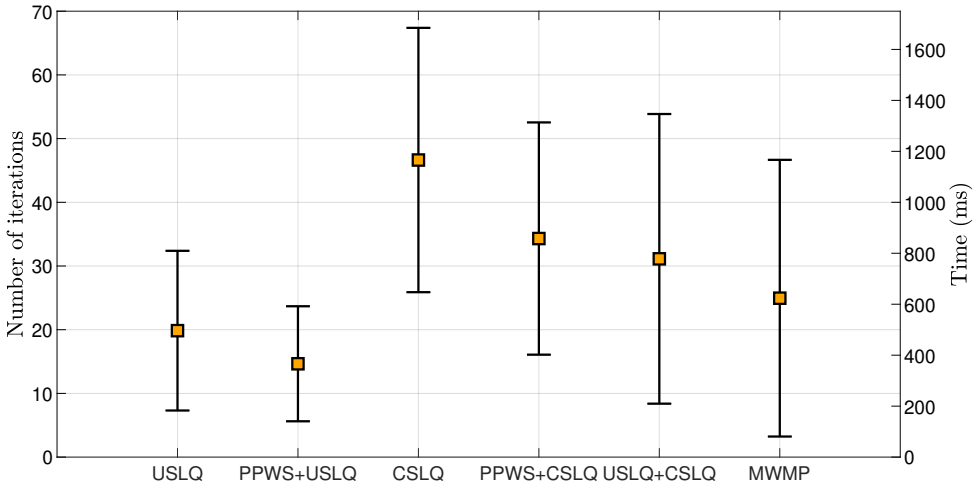


FIGURE 6.20: Measured iterations and execution time on the performance tests. For every layout, 21 different motion plans were launched. Average measurements are shown, including the standard deviation on the samples.

since the computational time spent is not representative due to its dependency on external factors, such as the hardware, the quality of the software implementation, or the CPU usage. Besides, the computational time spent is nearly proportional to the number of iterations, as an example, spending approximately 25 ms/it in these tests, run on a single core of an Intel(R) Core(TM) i7-10750H CPU (2.60 GHz).

The results of the performance tests are summarized in Figures 6.19 and 6.20. The path planner warm start (PPWS + USLQ, PPWS + CSLQ, PPWS + USLQ + CSLQ), as can be observed, always reduces the average number of iterations w.r.t. the cold started versions (USLQ, CSLQ, USLQ + CSLQ), reducing also the convergence speed variability. This means that the motion planner's behavior is more predictable. Additionally, the fastest layouts are the unconstrained ones (USLQ, 19.85 it; PPWS + USLQ, 14.65 it), as expected and aforementioned. Although these layouts have high success ratios (USLQ, 95.24 %; PPWS + USLQ, 95.24 %), they can not guarantee constraints compliance, thus they also have the lowest feasibility ratios (USLQ, 38.10 %; PPWS + USLQ, 57.14 %). The slowest layouts are the constrained ones (CSLQ, 46.64 it; PPWS + CSLQ, 34.31 it), and they still do not reach high feasibility ratios (CSLQ, 52.38 %; PPWS + CSLQ, 47.62 %), due to convergence difficulties considering the high over-actuation and number of constraints. Finally, the unconstrained-constrained layouts have an intermediate convergence speed (USLQ + CSLQ, 31.13 it; MWMP, 24.95 it), having the complete approach (MWMP) a comparable convergence speed to the unconstrained layouts. Besides, these layouts also have the highest feasibility ratios (USLQ + CSLQ, 76.19 %; MWMP, 90.48 %), thanks to the successive warm start procedure.

In summary, the performance tests demonstrate that the proposed multi-staged approach (MWMP) improves the behavior of the optimal motion planner, increasing noticeably the average number of feasible motion plans (90.48 %) and maintaining a considerably low average number of iterations until convergence (24.95 it).

Laboratory test campaign

Regarding the laboratory tests campaign, four of the most representative tests are analyzed in this paper, and one of them is shown in a summarizing video[14] of the lab tests campaign. Each test starts from a different rover location, also with a different pose of the sample tube. Additionally, an example of the evolution of the tests is also shown in Figure 6.21.

The experiments were run as follows. First, it was assumed that the sample was inside ExoTeR's LocCam Field of View (FoV). Therefore, the sample detection and localization subsystem was launched at the beginning to provide the initial estimation of the sample pose. Then, the sample pose was translated into an end effector goal pose, just above the sample and approaching the ground perpendicularly, and this goal pose was fed to MWMP to compute a global initial motion plan. This was

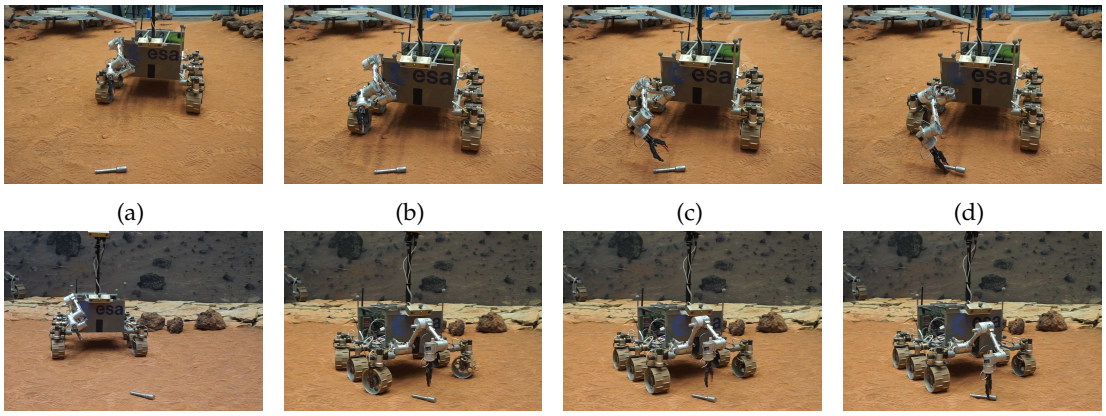
[14] `https://youtu.be/xDFv4Ho4KZs`

TABLE 6.7: Sample tube retrieval lab test results.

Sample Tube Retrieval Test Case	1	2	3	4
First plan number of iterations	17	20	29	13
Number of replans (after goal changed)	6 (6)	5 (3)	7 (2)	7 (5)
Average number of iterations in the test case	13.43	16.60	25.62	60.33
Average arm joints position error (°)	0.2521	0.2235	0.9167	0.1719
Average steering joints position error (°)	2.6471	0.0521	0.0997	0.0555
Rover base final pose error (m, °)	0.0023, 2.0798	0.036, 0.4183	0.02, 0.0	0.0361, 0.0630
End effector final pose error (m, °)	0.0942, 15.8996	0.0221, 12.4332	0.0441, 0.1833	0.0332, 24.9007
Sample pose estimation error (m, °)	0.0908, 15.7964	0.0150, 12.0035	0.0220, 0.6303	0.0030, 24.8893

sent to the motion plan follower, which started to send the control commands at each time step and receive the robot's measured state. As explained in Section 6.4, the follower continuously checked if any of the controlled states was accumulating too much drift, given the defined thresholds. Then, if necessary, the motion was replanned using the last motion plan as a warm start to accelerate the computation. Additionally, the sample detection and localization subsystem was launched repeatedly with a frequency of 0.1 Hz, to keep improving the sample pose estimation and filtering the localization error due to the use of visual odometry, eventually triggering additional motion replans. Note that even though the sampling frequency was low (0.1 Hz), the rover's nominal speed was quite slow too, of 5 cm/s. Thus, a sample detection was launched every 50 cm, which is sufficient to effectively detect the sample. Once the end effector had reached its goal pose, the execution was finished, and a separate sample retrieval component was launched just to perform the final sample grasping movement.

The carried out laboratory tests are examined in detail in the following. The results of four of the lab tests are summarized in Table 6.7. First, the initial motion plan number of iterations (20 it. avg) matches what is expected, seeing the results of the performance tests (25 it. avg). Furthermore, the average number of iterations in each test case, i.e. mean of iterations required by the planner to converge in each particular test case considering the initial plan and the replans, is generally lower than the first plan iterations. This confirms that using the last motion plan as a warm start accelerates the motion planning procedure, although it did not happen in case 4, since a sharp turn was required and the steering joints were repeatedly reaching their limits. It is remarkable that the average errors w.r.t. the planned motion of the controlled states, i.e. the arm joints (0.3896° avg) and the steering joints (0.7105° avg), are negligible, which means that the predicted motion was accurate and the linearization errors do not severely impact the system behavior, thanks to a sufficiently small time step Δt. Besides, most of the required replans were performed due to the low accuracy of the sample pose estimator (16 out of 25), which changed the goal pose substantially several times during the tests, as can be observed in Table 6.7. Regarding the non-directly controlled states, the errors of the rover base (0.0236 m, 0.6417° avg) and the end effector (0.0484 m, 13.3556° avg) final poses are acceptable, considering that the end effector final pose error is caused mainly by the



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE 6.21: Motion evolution of ExoTeR during two different sample tube retrieval tests, using a decoupled motion planning approach (a-d) or MWMP (e-h). The decoupled solution is not prepared to retrieve the sample once it reaches it (b), and can not retrieve the sample perpendicularly to the ground (c), with the result of a defective grasp (d). On the other hand, MWMP generates an optimal motion, leaving the arm prepared for the retrieval operation as soon as the base stops (f), being it placed in a certain pose which allows the manipulator to retrieve the sample perpendicularly (g), with a high-quality grasp (h).

sample pose estimator, being minimal the errors induced by the motion planner itself. Therefore, it is confirmed that the system model is representative despite of the linearization and that the motion planner is accurate enough to ensure a successful sample retrieval, being the 7 cm full opened gripper width enough to cope with these errors.

Finally, the evolution of Test Case 2 is shown in Figures 6.21e-6.21h, in comparison to another sample retrieval test in Figures 6.21a-6.21d, performed with a standard non-optimal and decoupled motion planning approach (Mantoani et al., 2022). As can be observed, although both approaches start from similar situations (Figures 6.21a and 6.21e), MWMP leaves the manipulator prepared for the retrieval operation as soon as the rover reaches the sample (Figure 6.21f), meanwhile the decoupled solution yet requires to move the arm once the rover stops (Figure 6.21b). Additionally, the decoupled solution does not place the rover base in a good position considering the posterior retrieval operation, thus, the gripper orientation is not perpendicular to the ground and does not match the sample yaw (Figure 6.21c), which, in the end, generates a defective grasp (Figure 6.21d). Conversely, the optimal motion planner uses all the system joints (rover and manipulator), placing the base to leave the arm in a perfectly perpendicular pose w.r.t. the sample. Therefore, the gripper is orientated perpendicularly to the ground and matches the sample yaw (Figure 6.21g), being the quality of the grasp, thus, much higher (Figure 6.21h).

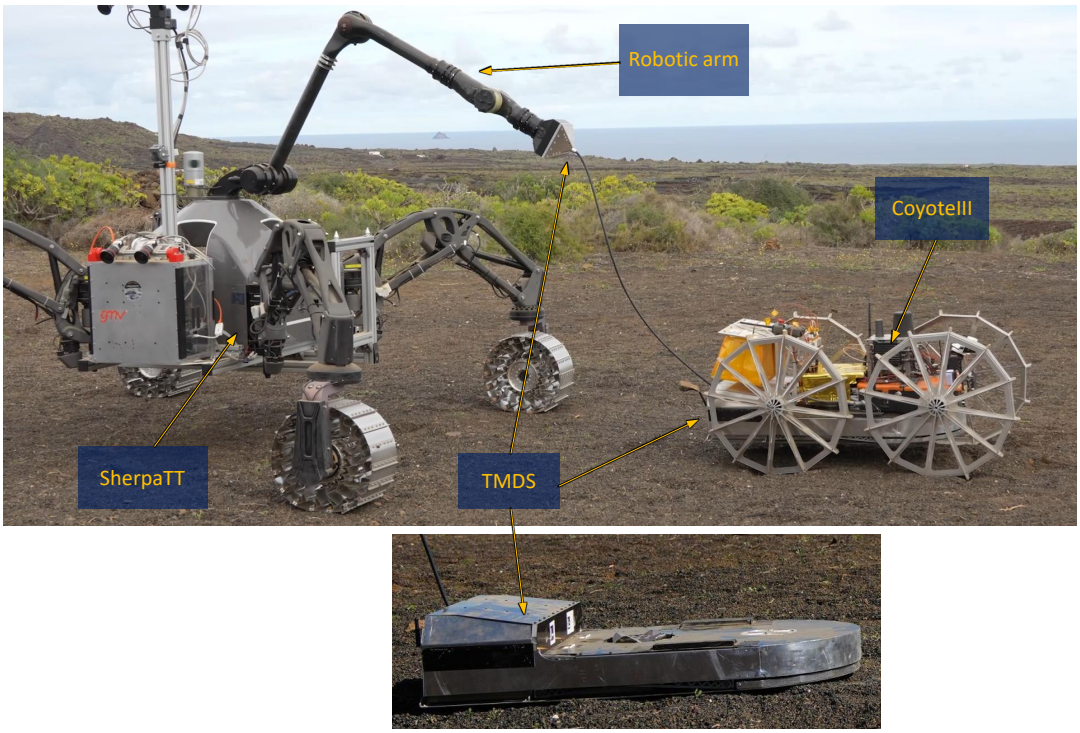


FIGURE 6.22: Systems involved in the rappelling mission. SherpaTT's robotic arm is attached to CoyoteIII through the TMDS, prepared to start a rappelling operation into a lava tube in Lanzarote, Canary Islands, Spain.

6.5 Multi-robot rappelling into Lunar lava tubes

The Moon has been the main focus of worldwide space agencies ever since they existed, as already presented in the first chapters of this thesis. Our satellite could provide answers to a better understanding of planetary processes and the origin of our solar system, or it could be used as a space technology testbed, before trying to reach more distant space bodies. But, more importantly, the Moon could be eventually used as a "space station", to provide shelter to humans when traveling through our solar system. Among the exploration targets on the Moon, the lava tubes are particularly interesting, since they are a natural access to the Lunar subsurface. Isolated from space radiation, the Lunar subsurface is supposed to keep intact materials from the formation of our solar system and can be directly used by humankind as a natural shelter from radiation when staying there. Nevertheless, getting to the Moon and exploring a Lunar lava tube is still expensive, challenging, and dangerous for humans, which could be solved by robots. As investigated during the European CoRob-X project, a multidisciplinary team of robots could coordinate to autonomously explore a Lunar lava tube. First, mapping the skylight where to access the lava tube. Second, rappelling a small robot into it, using a big robot as a mobile anchor. Third, employing the small robot to explore the lava tube interior.

Focusing on the rappelling mission, it involved three main systems as shown in Figure 6.22. First, the SherpaTT rover, which acts as a mobile anchor that supports the rappelling with its robotic arm. Second, CoyoteIII, as the small rover able to

rappel down the skylight to later explore the lava tube. Third, the Tether Management and Docking System (TMDS), which is the system that connects the two rovers, where CoyoteIII can dock to perform the rappelling or receive power and data from outside the tube.

As can be understood, this mission is quite challenging, considering that a failure would severely damage CoyoteIII, and probably also SherpaTT. Particularly critical is, thus, controlling the tension that the tether applies to the rovers, in two main senses. On the one hand, given the considerable weight of CoyoteIII and the TMDS, it is essential to properly distribute the tension on SherpaTT's robotic arm, absorbing the main efforts with the bigger joints to avoid damaging the smaller ones. On the other hand, insofar as CoyoteIII would eventually be completely hanging from the tether, it is crucial to maintain a sufficient amount of tension on the tether when the rover is approaching the skylight. Otherwise, CoyoteIII could briefly, but abruptly, fall free when transitioning to a vertical hang, which would result in a sudden pull from the tether that could harm all of the systems involved.

Consequently, the technology developed within this doctoral work was key to controlling the tension during the rappelling mission of the CoRob-X project, enabling an autonomous rappelling operation. The optimal motion planner presented in Chapter 5 was integrated into a MPC scheme to control SherpaTT, dynamically reacting to the forces that the tether applies to the manipulator's end effector. As a result, SherpaTT was able to perform MM motions to compensate and distribute those efforts among the stronger joints of the robotic arm. Additionally, the guided movements of CoyoteIII were coordinated with the deployment of the tether through a simple proportional controller, which always maintained an acceptable level of tension on the tether to avoid sudden pulls. Let us present the platforms and the tension controllers in the following, before showing the results of the simulation campaign and the real field trials carried out in an analogue Lunar lava tube in Lanzarote, Canary Islands, Spain.

Platforms description

Three main systems were involved in the autonomous rappelling test, shown in Figure 6.22. First, SherpaTT, which has already been presented before in this chapter. It is a four-wheeled, full-Ackermann rover that weighs approximately 230 kg when fully equipped, i.e. including the power generator and the external avionics box. Its size makes it capable of carrying heavy payloads, being the perfect system to act as a mobile anchor during the rappelling operation. SherpaTT has also onboarded a robotic arm, whose characteristics were summarized before in Table 6.4. It has six DoF and a spherical wrist, the spherical wrist joints (4, 5, and 6) are the weaker and smaller ones, due to small gear ratios. Conversely, the shoulder (1 and 2) and elbow (3) joints have considerable gear ratios, being thus stronger and bigger, capable of withstanding considerable efforts. The maximum payload of the manipulator is 25 kg, limited by the spherical wrist joints. The manipulator is also equipped with

a Mini45 force/torque sensor, located at its end effector, and an electro-mechanical interface, which is able to retrieve different tools and transmit power and data.

The second rover involved in the rappelling is CoyoteIII. Also developed at DFKI RIC, Bremen, this is a micro-rover with low weight (approximately 20 kg) and high mobility, which makes it suitable for rappelling and exploring narrow lava tubes. It has skid-steering locomotion with four wheels, which were initially star-shaped but later substituted for standard, round ones. This change was required to improve the locomotion capabilities during the rappelling operation, to carry heavier loads (the TMDS), and to increase the ground clearance.

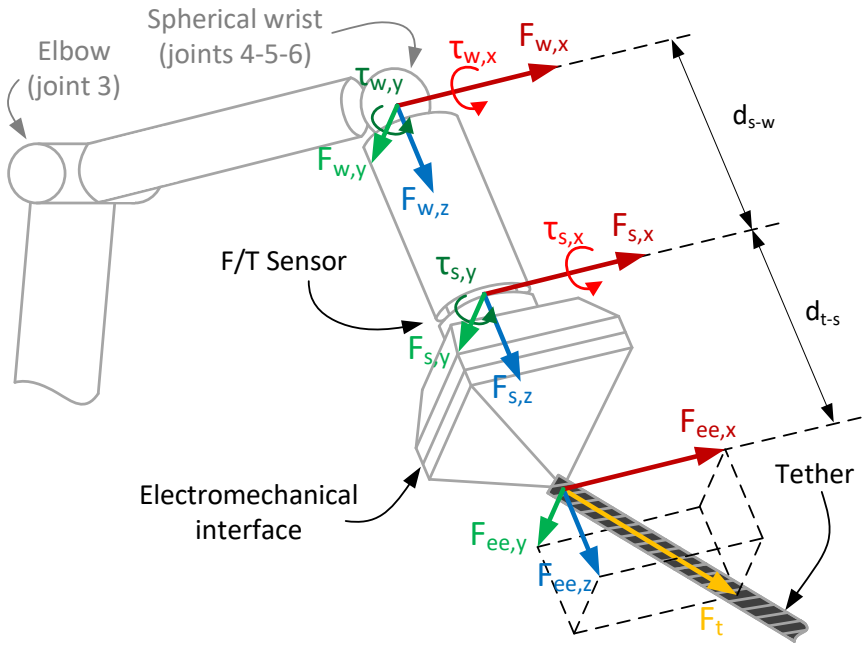
A third robotic system is crucial for the rappelling: the TMDS. The TMDS contains the tether that connects both SherpaTT and CoyoteIII. The tether is rolled up on an actuated spool inside the TMDS, to control the winding (and unwinding) velocity. The TMDS is also equipped with a force sensor, to measure the tension. If the tension on the tether is too small, then the spool motor is blocked to avoid entanglements inside the TMDS. Besides being used as a rope to rappel CoyoteIII down the skylight, the tether also transmits power and data from SherpaTT to the TMDS to command its actuators. Additionally, this link can be used to provide energy to CoyoteIII inside the lava tube, and for communication between the rovers. The TMDS is connected to the rovers with two different interfaces. On the one hand, the tether connects directly to SherpaTT's manipulator end effector through the electro-mechanical interface. On the other hand, the TMDS has a hot-dock interface that allows CoyoteIII to quickly dock and carry the TMDS before rappelling, or undocking once the rover is inside the lava tube to start the exploration. Remark that with this TMDS design, the tether unwinds from CoyoteIII meanwhile it descends, similarly to standard human rappelling. This is a fundamental feature, otherwise, the tether could suffer damage from being dragged on the surface.

Manipulator's joints tension

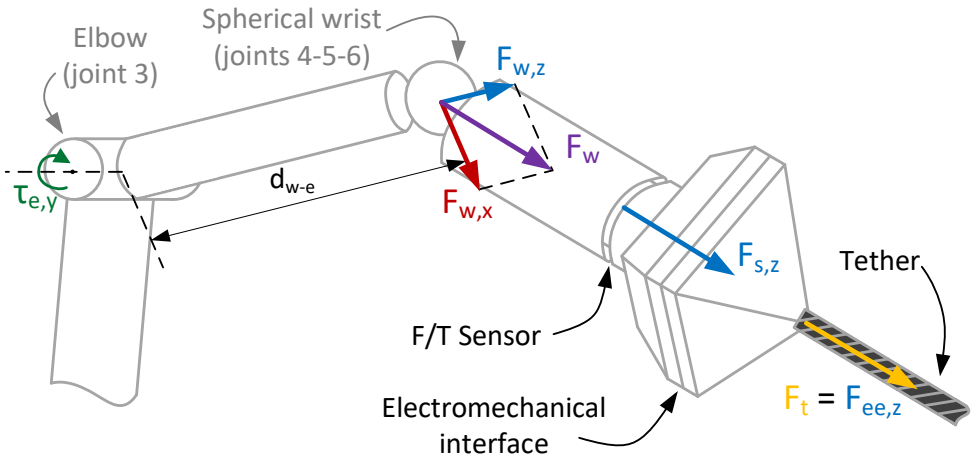
For the rappelling mission to be successful it is crucial to control the tension that the tether applies to both rovers, SherpaTT and CoyoteIII. Otherwise, as aforementioned, the systems would suffer efforts that they may not be able to withstand, jeopardizing the mission.

Let us start analyzing the efforts applied to SherpaTT's robotic arm, which are depicted in Figure 6.23a. Being F_t (N) the tension on the tether, it generates a force $F_{ee} = F_t$ into the tip of the arm's end effector, which can be decomposed in each axis according to the direction of the last link of SherpaTT's arm, i.e. $F_{ee} = [F_{ee,x}\ F_{ee,y}\ F_{ee,z}]$. These forces are endured by the whole link, i.e. by the spherical wrist and by the force/torque sensor, as in (6.4).

$$[F_{w,x}\ F_{w,y}\ F_{w,z}] = [F_{s,x}\ F_{s,y}\ F_{s,z}] = [F_{ee,x}\ F_{ee,y}\ F_{ee,z}] \tag{6.4}$$



(a) The tether and the arm's last link are not aligned, which generates torques on the spherical wrist.



(b) The tether and the arm's last link are aligned, hence, the torques do not affect the spherical wrist. Instead, they are transferred to the elbow joint, which is stronger.

FIGURE 6.23: Efforts that the tether applies to SherpaTT's robotic arm during the rappelling mission.

Being $[F_{w,x}\ F_{w,y}\ F_{w,z}]$ the forces endured by the wrist and $[F_{s,x}\ F_{s,y}\ F_{s,z}]$ the forces measured on the force/torque sensor.

Regarding the torques, $F_{ee,z}$ is aligned with the last link, thus, it does not generate any torque. Conversely, $F_{ee,x}$ and $F_{ee,y}$ are perpendicular to the last link, therefore, they generate some torques alongside the robotic arm. The force/torque sensor experiences the torques $\tau_{s,x}$ and $\tau_{s,y}$ (N m), as shown in (6.5-6.6).

$$\tau_{s,x} = F_{ee,y}\, d_{ee-s} \tag{6.5}$$

$$\tau_{s,y} = F_{ee,x}\, d_{ee-s} \tag{6.6}$$

Where d_{ee-s} (m) is the distance between the tip of the arm's end effector and the force/torque sensor. Additionally, $F_{ee,x}$ and $F_{ee,y}$ generate torques in the spherical wrist of the manipulator, as expressed in (6.7-6.8).

$$\tau_{w,x} = F_{ee,y}(d_{ee-s} + d_{s-w}) \tag{6.7}$$

$$\tau_{w,y} = F_{ee,x}(d_{ee-s} + d_{s-w}) \tag{6.8}$$

Where d_{s-w} (m) is the distance between the force/torque sensor and the spherical wrist.

The weakest point of the robotic arm during the rappelling operation is the spherical wrist. The wrist joints are capable of enduring forces but are not designed to resist high torques. As a result, these joints yield if $\tau_{w,x}$ and $\tau_{w,y}$ are big, which only happens if $F_{ee,x}$ and $F_{ee,y}$ are considerable, i.e. if there is a significant misalignment between the directions of the tether and the last link of the arm.

These torques can be eliminated, thus, by aligning the arm with the tether, as demonstrated in Figure 6.23b. When the tether and the last link of the arm are completely aligned the tether tension generates only axial forces into the arm, $F_{ee} = F_{ee,z} = F_t$. Hence, the force/torque sensor measures directly the tension on the tether, $F_{s,z} \simeq F_t$, being zero the rest of the measured forces and torques ($[F_{s,x}, F_{s,y}, \tau_{s,x}, \tau_{s,y}] \simeq 0$). Similarly, the spherical wrist only endures the axial force $F_w = F_t$, no torques, which are transferred to the elbow joint as in (6.9).

$$\tau_{e,y} = F_{w,x}\, d_{w-e} \tag{6.9}$$

Where $\tau_{e,y}$ is the torque endured by the elbow joint, $F_{w,x}$ is the transversal force acting on the spherical wrist ($F_w = [F_{w,x}\ 0\ F_{w,z}]$) according to the elbow-wrist link direction, and d_{w-e} is the distance between the spherical wrist and the elbow joint. Consequently, the stronger elbow joint is the one withstanding the torques as desired. Meanwhile, the spherical wrist joints only experience axial forces.

Considering the hazard of torques acting on the spherical wrist, it is necessary to include a controller that aligns the arm's last link and the tether to ensure the safety of the robotic arm during the rappelling operation, since this alignment eliminates the torques on the wrist, as demonstrated above. In this regard, the over-actuated

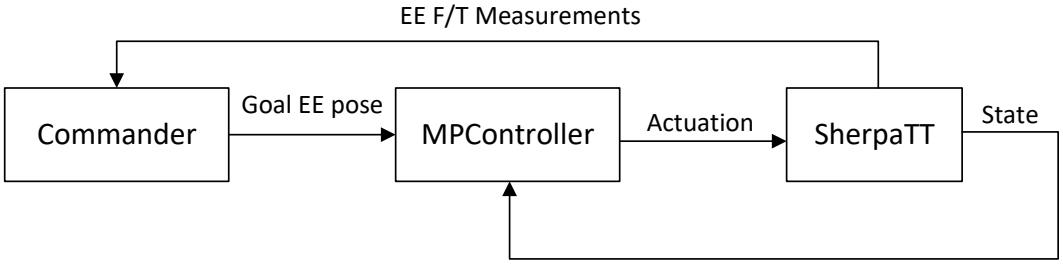


FIGURE 6.24: Pipeline to control the tension that the tether applies to SherpaTT's arm, including a commander for the end effector poses and an MPC.

MM capabilities of SherpaTT give the system a lot of possibilities to actively compensate for the measured forces and torques, by moving the mobile base and/or the manipulator. A MM efforts compensation algorithm was, thus, developed, with two main components performing two control loops, as shown in Figure 6.24. Let us detail these two components, the end effector goal pose commander and the MPC whole-body controller, in the following.

End effector goal pose commander

The first component is the commander, which receives the measurements of the force/torque sensor to compute a goal end effector pose that would compensate them. The measured transversal forces, $F_{s,x}$ and $F_{s,y}$, generate a relative linear displacement of the goal pose in the direction of the corresponding force. To do so, they are scaled to translate the forces to displacements as if a spring were located in the end effector, following (6.10-6.11).

$$\Delta x = F_{s,x}/k_x \tag{6.10}$$

$$\Delta y = F_{s,y}/k_y \tag{6.11}$$

Where Δx and Δy (m) are the relative displacements in the x and y axes of the end effector respectively, and k_x and k_y are their scaling factors, i.e. spring constants (N/m). Similarly, the measured torques, $\tau_{s,x}$ and $\tau_{s,y}$, are translated into a relative angular displacement of the goal pose, as expressed in (6.12-6.13).

$$\Delta\varphi = \tau_{s,x}/k_\varphi \tag{6.12}$$

$$\Delta\vartheta = \tau_{s,y}/k_\vartheta \tag{6.13}$$

With $\Delta\varphi$ and $\Delta\vartheta$ (rad) the relative rotations for the x and y axes of the end effector respectively, and k_φ and k_ϑ their scaling factors (N m/rad). Note that the measured axial force, $F_{s,z}$, is ignored, since it does not generate torques on the wrist, and it will always be present as long as there is tension on the tether. Also, note that all the force/torque measurements are filtered to erase high-frequency noises affecting the sensor.

MPC for enhanced reactiveness

The second component of the efforts compensation algorithm is an MPC. Briefly, the controller takes the goal end effector pose as input, computes an optimal actuation for the system's joints to reach the goal, and actuates the real system. In this regard, the MPC actuates both the mobile base joints and the manipulator ones to reach the goal end effector pose, although the mobile base movements are restricted to be parallel to the skylight to avoid the rover getting too close to it. Besides, the MPC continuously measures the system's state to filter disturbances, and checks for new goals even if the previous one has not been reached yet, hence quickly reacting to changes in the tension of the tether.

Let us provide more details about the controller. As stated in Chapter 2.4, MPC is one of the most common controllers in the robotics community for its high, fast performance. MPC controls the robot to reach a desired goal state while predicting the robot's evolution in the immediately following period. This prediction, based on a model of the system, serves as a comparison to continuously evaluate how the real system evolves w.r.t. what is expected. Feeding this information back to the controller allows it to wisely compensate for mismatches and disturbances, modifying its initial plan to still get to the desired goal.

In the rappelling support use case, it is not relevant if the controller fails to find a suitable solution for the problem from time to time. The priority is to quickly plan the motion to find new solutions for the next few seconds and keep commanding the system, so it reacts as fast as possible to any high-frequency disturbances, therefore, MPC is the best option. The developed MPC makes use of MWMP, the optimal motion planner presented in Chapter 5.3. The integration of MWMP into an MPC scheme is quite straightforward since MWMP is based on optimal control techniques, as follows. In the first place, MWMP receives a Cartesian high-level goal for the manipulator end effector, from the end effector goal commander. Using the current state of the system, MWMP generates a motion plan to reach that goal within the given prediction horizon, if possible. The prediction horizon, usually smaller than one second, should be tuned depending on the computational complexity of the motion planning problem and the frequency of the external disturbances, to be able to react to them. In the second place, the controller gets this motion plan and starts sending motion commands to the system. MPC keeps sending the commands until it hits the configured maximum number of consecutive commands (which is usually one in most state-of-art MPCs). Remark that a higher number of consecutive commands means smoother movements and less computational cost, but also slower reactiveness. After sending the configured consecutive commands, the prediction horizon recedes, MWMP generates a new motion plan according to the current state of the system, and MPC starts commanding again. These steps are sequentially repeated until the goal is reached. Find a summary of the main parameters configuring the MPC and the arm's tension controller in Table 6.8.

Note that there are a few issues that could arise given this MPC procedure, as

TABLE 6.8: Configuration of the arm's tension control components.

Component	Parameter	Value
Commander	Stiffness k_x (N/m)	3000
	Stiffness k_y (N/m)	100
	Stiffness k_φ (N m/rad)	inf
	Stiffness k_ϑ (N m/rad)	10
MPC	Prediction time t_N (s)	0.4
	Time step Δt (s)	0.029
	Max consecutive commands	14

follows. If a new goal is requested but the previous one has not been reached yet, the controller will cancel the current plan and substitute the old goal for the new one. Also, if MWMP eventually fails to find a solution, MPC will keep the previous one until MWMP manages to generate a new motion plan. If MWMP keeps failing, it will retry several times until finally rejecting the goal, which indicates that the goal is not reachable for some reason, e.g. obstacles or system limitations.

Tether tension control

SherpaTT actively moves to compensate for the efforts applied to its manipulator with the algorithm presented above. But, at the same time, CoyoteIII is rappelling the skylight at a constant linear velocity, and the TMDS is unwinding the tether. The relative speed between the two systems, the tether and CoyoteIII, defines the tension that the tether finally undergoes. A small tension could lead to abrupt pulls from the rover once the vertical descents starts, as aforementioned. On the contrary, high tether tension could induce slippage or even sinkage of the rover wheels when approaching the skylight. As a result, it is essential to control the tether unwinding speed to induce a reasonable tension to the rover.

In this regard, a simple tether tension controller component was included, as follows. Intuitively, if the tether unwinds at the same speed as CoyoteIII is advancing, then the tension on the tether is maintained. If the tether unwinds slower than CoyoteIII, then the tension increases, and vice versa. According to these rules, the tension can be directly controlled by slightly modifying the unwinding speed of the tether depending on the measured and desired tether tensions, as in (6.14).

$$v_u = v_c + \mathcal{P}(F_t - F_r) \tag{6.14}$$

Where v_u (m/s) is the unwinding velocity of the tether, v_c (m/s) is the linear speed of CoyoteIII, $\mathcal{P}$ (m/(s N)) is a configurable proportional gain of the controller, F_t (N) is the measured tension of the tether and F_r (N) is the desired or reference tension.

Thus, the measured tension on the TMDS F_t and the current velocity of CoyoteIII v_c are fed back to the controller to modify the tether unwinding speed, therefore

FIGURE 6.25: Snapshot of a MARS simulated rappelling into a lava tube (bottom right), with SherpaTT (left) attached to CoyoteIII (center) by a simulated tether.

maintaining adequate tension on the tether. The reference tension was selected under the rappelling state, i.e. low tension when the rover is still approaching the skylight and high tension when the rover is about to start descending vertically. The proportional gain $\mathcal{P}$ was tuned experimentally, being $0.0005\,\mathrm{m}/(\mathrm{s}\,\mathrm{N})$ the final configured value during the tests.

Simulation campaign

Before the real field trials, a simulation campaign was carried out to verify and validate the viability of the rappelling operation, to confirm the usefulness of the MM MPC tension controller, as well as debug and improve the involved algorithms as much as possible. To do so, the MARS simulator was used. This is a robotics simulator developed by DFKI RIC, as already presented before. MARS makes use of the ODE, including optimization techniques to compute the physics, i.e. the contact points and forces that are applied to the different bodies. Additionally, MARS wraps sensors, actuators, and several plugins to cope with various functionalities. The autonomous architecture stack was also integrated within the simulations using the software-in-the-loop concept, with ROCK as the real-time operating system for both SherpaTT and CoyoteIII. Within this stack, the newly developed tension controllers were included.

A snapshot of the MARS simulation environment during one of the rappelling tests is shown in Figure 6.25. Here, the models of SherpaTT and CoyoteIII were generated utilizing their URDFs, with the TMDS model included in the URDF of CoyoteIII. The model of SherpaTT includes, as required, a force/torque sensor on the manipulator's end effector. On the other hand, the scenario replicates a real lava tube located in Lanzarote, Canary Islands, Spain. An aerial survey was performed

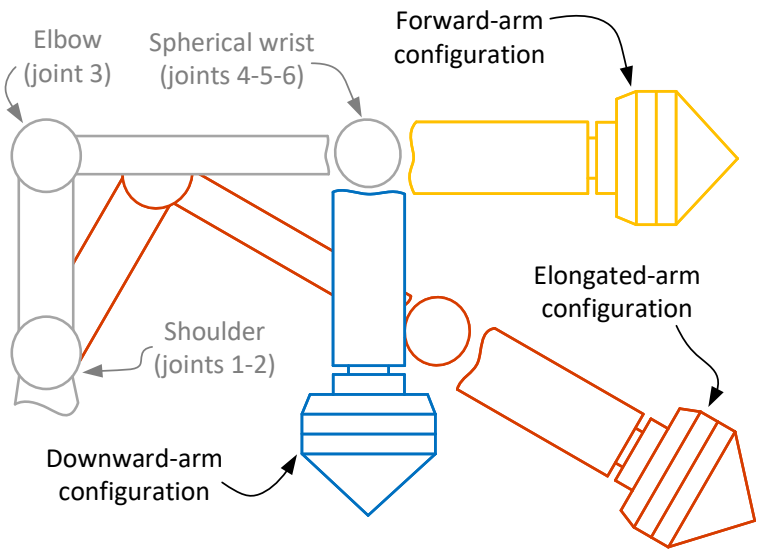


FIGURE 6.26: Depiction of the different initial configurations (downward, elongated, forward) of SherpaTT's robotic arm during the rappelling tests in simulation.

to obtain the 3D model of the surroundings of the lava tube's skylight. Then, a rover with a LIDAR was introduced inside the cave to 3D map its interior. Finally, as can be observed in Figure 6.25, both models (lava tube's exterior and interior) were merged and included in the MARS simulator. As a result, the models of the scenario and the rovers replicated the later field trials with a good level of precision, which made the simulations accurate and useful enough to improve the operation before the real test campaign in Lanzarote.

Apart from the rovers and the scenario, there was another system included in the rappelling simulation: the tether. The tether is a flexible, non-stretchable system with variable and controllable length. Due to its physical complexity, a tailored plugin was developed by DFKI to model it, as follows. The virtual tether is composed of a line of particles. Each particle applies forces to the adjacent particles depending on the distance between them, as if a spring were placed in between. The constant k of the spring determines the rigidness of the tether: high k generates a strong, rigid rope, meanwhile low k generates a loose, stretchable rope. At each timestep of the simulation, these forces between the particles are computed and applied. Afterward, the particles' positions are modified considering their mass, inertia, and additional external factors, such as contact points or gravity acceleration. Additionally, the tether can be enlarged or shortened by adding or removing particles respectively. In this regard, the tether plugin includes as an input the wind/unwind velocity. According to the simulation timestep and the size of the particles, the plugin continuously generates new particles at the tether's end if the commanded velocity is positive, or erases particles if the velocity is negative.

After putting together all of these models, the rappelling simulations were carried out. Four types of tests were arranged to confirm the usefulness of the SherpaTT's arm MM MPC tension controller. The goal was to analyze the torques applied to each joint of the manipulator during the rappelling and see if the MM component can redistribute and reduce those torques w.r.t. maintaining the arm still. Thus, three of these tests had MM disabled and kept the arm in a particular initial configuration throughout the whole test. These initial configurations are shown in Figure 6.26, which include a first one with the arm looking downward, perpendicular to the ground (which is also the one shown in Figure 6.25), a second one called elongated, with the arm slightly inclined looking towards the skylight (which is the same starting configuration when MM is enabled, as shown in Figure 6.22), and a third one with the arm looking forward, parallel to the ground.

As can be observed in the CoRob-X rappelling video[15], the simulations were as follows. SherpaTT and CoyoteIII began looking towards the skylight, already attached by the tether. CoyoteIII started moving in a straight line at a constant velocity to the skylight, meanwhile, the tether got unwinded at a controlled velocity according to its tension. In those tests where the MM MPC component was activated, SherpaTT compensated for the tether tension by aligning the arm's last link with the tether. Once CoyoteIII was close to the skylight it slowed down, and transitioned to a completely vertical hang from the tether. At this point, the tether started to contact the ground at the edge of the skylight, and the rover descended gradually until it touched the bottom of the skylight. Then, the rover kept advancing until it was sufficiently horizontal inside the cave, when the rover and the tether stopped, and the rappelling operation was concluded.

The results of four of the rappelling simulation tests are shown in Figure 6.27, where the applied torques on each joint of SherpaTT's arm throughout the rappelling operation are depicted, alongside the joints' torque limits. As can be observed, maintaining the arm looking downward is unsafe, since joints 2, 3, and 5 overcome the torque limits continuously during the rappelling. Maintaining the arm looking forward is not safe either, with joint 5 also surpassing its maximum torque for most of the rappelling operation. Conversely, the elongated arm configuration (the arm's last link already looking towards the skylight) shows quite good results, with only high torques in the second joint but within the limits. Nevertheless, beware that the elongated configuration is a very particular case since it is by chance already orientated to the skylight. Such a good arm configuration cannot be easily estimated before a rappelling execution, and would not be as valid for any scenario (for instance, if CoyoteIII has to ascend to get to the skylight rim).

Regarding the simulation test where the MM MPC tension controller was enabled, the torques acting on all arm joints are considerably reduced throughout the rappelling and completely maintained within the limits, as shown in Figure 6.27.

[15] `https://youtu.be/zJE6d-LOpcI`

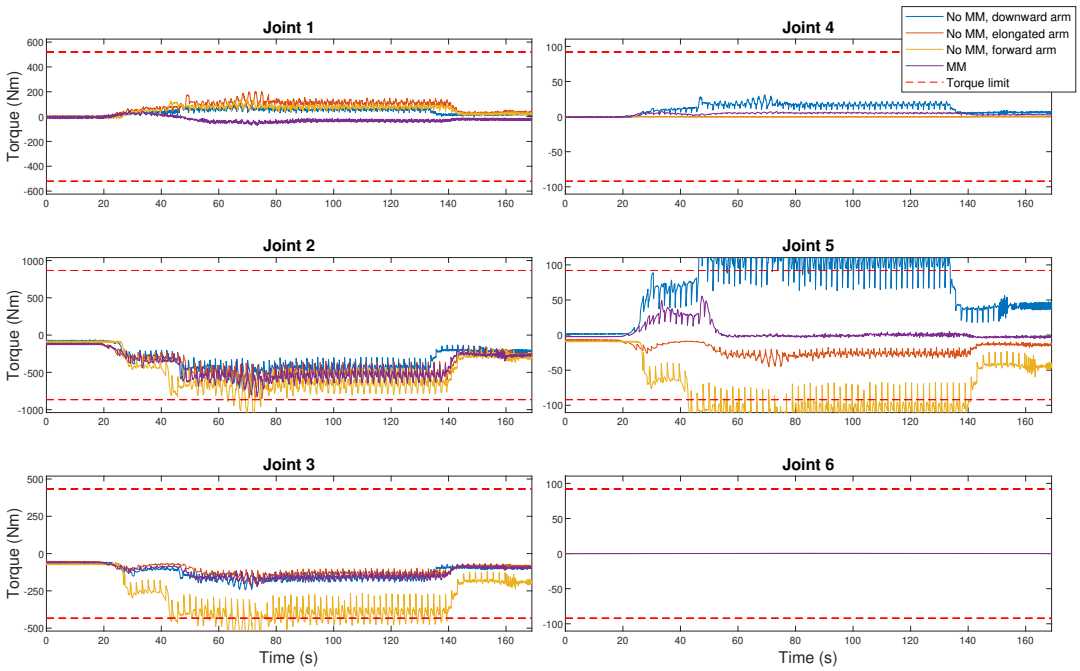


FIGURE 6.27: Torques experienced by SherpaTT's arm during the rappelling tests performed in simulation with MARS. Four different simulations were performed: one with the MM MPC tension controller enabled, and three without it, maintaining the arm still in different initial configurations.

This is thanks to the dynamic movements of the arm and the mobile base, continuously reacting to the measured forces and torques as intended. Particularly, with MM the spherical wrist joints (4, 5, and 6) suffer almost no torque during the operation, with just a small increase in joint 5 when the rappelling started until the MPC was able to compensate it properly. Also, the first joint does not endure almost any torques, which are mainly absorbed by joint 2, the strongest one. Given these results, it is confirmed that the MM MPC tension controller increases the safety of the rappelling operation, by reducing and distributing the torques on SherpaTT's manipulator joints.

In summary, these simulations helped debug and improve the algorithms, tuning the different configuration parameters like the speed of CoyoteIII, the expected tensions on the tether, or the controllers' gains. They also demonstrated the usefulness of including the MM MPC controller to increase the safety of the operation. Finally, some unexpected events lead to a series of lessons learned. First, the simulations showed that CoyoteIII could continuously turn axially during the vertical descent. Although not critical itself, this turning sometimes led to undesirable touchdowns where the rover ended upside down. As a workaround, it was decided to keep moving the rover's wheels even during the vertical descent, so the rover advances as soon as it touches the ground, hence, ensuring a proper landing. Second, in some cases, the tether kept some tension after CoyoteIII landed, leading to undesirable pulls even when the systems were already stopped. This issue was directly solved

(a) (b)

FIGURE 6.28: Skylight (a) and lava tube (b) to be explored during the CoRob-X field trials in Lanzarote, Canary Islands, Spain.

by slightly increasing the unroll speed of the tether after the touchdown, gently depositing the rover at the cave, and reducing the tension to a safer level. Third, the MM MPC for SherpaTT's arm was occasionally causing small pulls from the tether meanwhile trying to align the manipulator's last link. Even though these pulls are not critical for the success of the operation, they increase the tension on the tether and induce some slip into the wheels of CoyoteIII, facts that could eventually be undesirable if happening in a critical stage, such as the transition to the vertical descent. Therefore, the MPC was configured to avoid as much as possible changing the position of the manipulator's end effector, instead mainly modifying its orientation.

Lanzarote field trials

The CoRob-X field trials lasted for three weeks, to autonomously explore an analogue Lunar lava tube with a team of three different rovers: SherpaTT, CoyoteIII, and LUVMI-X. The selected scenario, located in Lanzarote, Canary Islands, Spain, was composed of a narrow, sharp skylight, shown in Figure 6.28a, which conducted into a considerably big lava tube, shown in Figure 6.28b. This scenario was selected since the lava tube's interior was smooth and flat, ideal for demonstrating an exploration with CoyoteIII.

Regarding the equipment, SherpaTT and CoyoteIII had onboard a CPU running their control subsystems within ROCK, including the tension controllers as in the

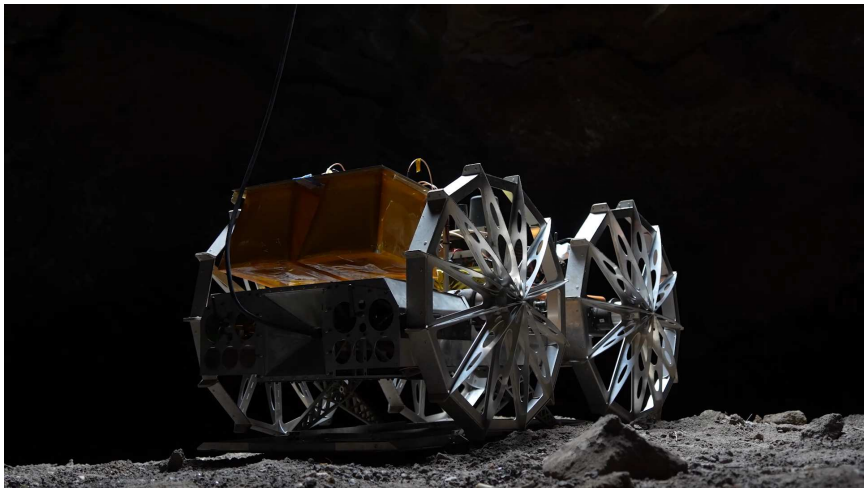
FIGURE 6.29: Final configuration of SherpaTT's robotic arm at the end of a rappelling operation. After the forces are compensated, the arm's last link is perfectly aligned with the tether, thus, the efforts are transferred to the shoulder (joints 1-2) and elbow (joint 3), which are stronger, relieving the spherical wrist (joints 4-5-6).

simulations. The rovers were equipped with WiFi antennas to communicate between them and with the ground monitoring and control station. Additionally, SherpaTT communicated with the TMDS through the electro-mechanical interface on the end effector of its robotic arm. Also, CoyoteIII could establish communication with the TMDS, but through the hot-dock interface placed on its body. Communicating through the TMDS was key to sending and receiving information to/from CoyoteIII once the rover was inside the lava tube since the WiFi couldn't reach its interior.

The rappelling tests were conducted as follows, as can be observed in the complementary rappelling video[16]. First, the most suitable spot for the rappelling was manually selected by human operators (an autonomous selection of the descent spot was out of the scope of the project). Second, both rovers were placed looking towards that descent spot, in a straight line. Third, SherpaTT deployed the TMDS to the ground using its robotic arm, CoyoteIII walked on top of the TMDS and docked to it. Then, the rappelling operation began, in a similar manner as the previous simulations. CoyoteIII started moving to the skylight at the same time that the tether was unwinded. The unwinding velocity of the tether was computed by the tether tension controller presented above. In parallel, SherpaTT continuously modified the configuration of its robotic arm and the position of its body to compensate for the tether tension, by using the MM MPC component. Whenever CoyoteIII reached the skylight and transitioned to a vertical hang, the MM component was stopped, since the efforts wouldn't change from then on, and the manipulator was already aligned

[16]See footnote 15

(a) CoyoteIII vertically descending through the skylight (DFKI RIC).



(b) CoyoteIII undocking from the TMDS after successfully landing inside the lava tube (DFKI RIC).

FIGURE 6.30: CoyoteIII rappelling sequence.

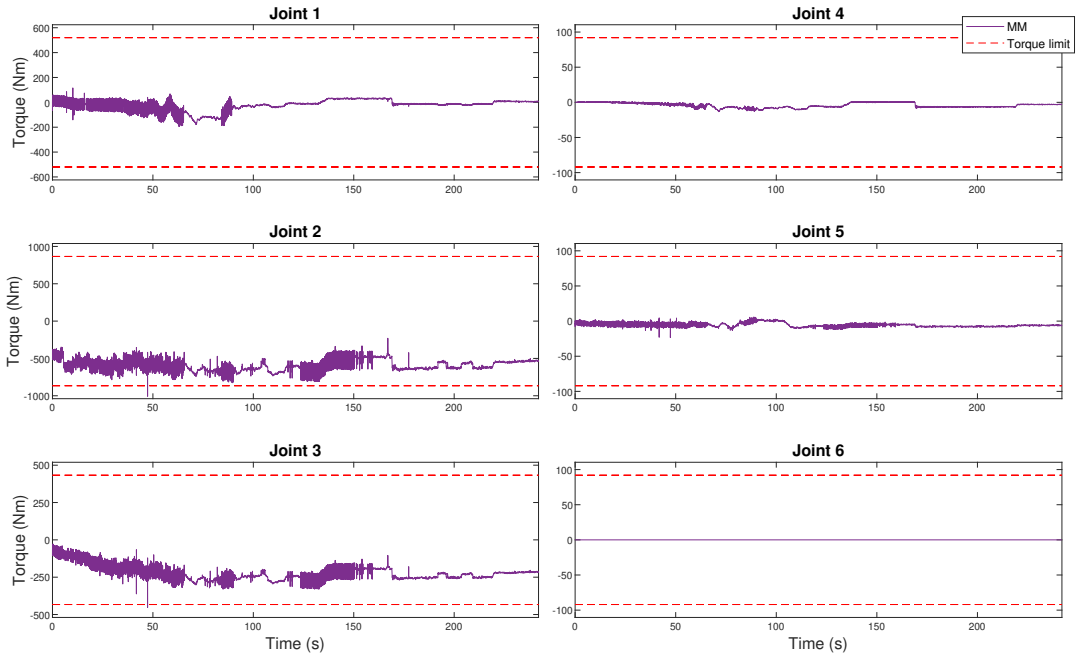


FIGURE 6.31: Torques experienced by SherpaTT's arm during one of the rappelling operations performed in a Lunar analogue lava tube in Lanzarote, Canary Islands, Spain. Since the MM MPC tension controller was enabled, the torques on the arm joints are maintained within the limits throughout the rappelling.

with the tether as can be observed in Figure 6.29, where it is shown the final state of SherpaTT in one of the rappelling tests. Besides, moving the manipulator once the tether is touching the ground is completely undesirable, since it could lead to dragging the tether on the surface and eventually damaging it. Finally, CoyoteIII kept descending vertically into the lava tube, as depicted in Figure 6.30a, until touchdown at the bottom of the lava tube. After landing, the rover kept its wheels rolling until its body was horizontal, as shown in Figure 6.30b. Then, CoyoteIII undocked from the TMDS, the rappelling operation was considered successful and the execution stopped.

Multiple rappelling operations were performed during the field trials, not all of them successful. Initially, a security rope was attached to CoyoteIII. During the first tests, CoyoteIII got stuck several times in the rim of the skylight due to a pointy rock touching its belly. Consequently, in one of the tests the docking between CoyoteIII and the TMDS failed, and both systems had to be manually retrieved from the skylight using this security rope. To avoid this rock, the descent spot was later changed. After that, the rappelling operations started to be mostly successful, despite some smaller issues due to communication drops between the rovers. Therefore, the security rope was removed, and multiple successful rappelling operations were performed, all of them with the MM MPC component enabled to ensure the robotic arm's safety.

One of the main issues during the field trials was related to the force/torque sensor placed on the end effector of SherpaTT's robotic arm. This sensor experienced

some de-calibration issues in the axial force (Z axis) that made the measurements slightly jump from time to time, which could not be solved during the field trials. Though this measurement error was not critical, and did not affect in any sense the MM MPC tension controller since the axial forces were ignored, the later logged data from the rappelling tests was not completely reliable for extracting quantitative conclusions. Anyhow, this data allowed us to perform a qualitative estimation of the torques experienced by the joints of SherpaTT's manipulator during one of the rappelling tests, with similar results to the simulations. As can be observed in Figure 6.31, the torques acting on all the different joints were kept within their limits thanks to the MM component. In particular, the spherical wrist (joints 4, 5, and 6) did not experience substantial torques, which were mainly absorbed by the shoulder (joints 1-2) and the elbow (joint 3), due to the alignment of the arm's last link and the tether.

6.6 Summary and conclusions

This chapter encompasses the results of this doctoral dissertation, i.e. the proofs that confirm the usefulness of all the proposed methods in the previous chapters. These results are in the form of, on the one hand, simulation tests, which try to replicate real scenarios and robots with high accuracy. Since a simulation can be easily performed without the need for big investments in equipment or transport, simulating is key in robotics to debug, improve, and analyze the performance of the software (and even hardware) components of the system. On the other hand, the results include laboratory and field tests with real robots in analogue planetary scenarios. Performing these real tests is time-consuming, expensive, and especially challenging for autonomous robots due to the high complexity of the system components and their interactions. Nevertheless, field tests are essential to demonstrate that the proposed methodologies can be of use in real systems and scenarios, and eventually be included in future planetary missions. Consequently, four main simulation and field test campaigns have been depicted in this chapter, each one focusing on one contribution of this doctoral research.

The first step for an autonomous MM rover is to autonomously reach the area of scientific interest in the scenario. Enhancing these autonomous navigation capabilities is the purpose of the first contribution, the dynamic cost map update procedure to enhance path planning in autonomous rovers, which was validated firstly in a simulation environment within CoppeliaSim, and afterward, in a field test with HDPR, a planetary rover from ESA-PRL. Results demonstrate that the proposed approach improves the rover's path planning performance continuously during the traverse since the rover's awareness of the scenario is increased, in two main senses. First, the estimated terramechanic properties of the terrains (slippage, obstacles density) are more accurate. Second, the obstacles newly found are included in the cost map to be later considered when path planning. These cost map update procedure particularly enhances path planning for round-trip missions as demonstrated in the

simulation and field tests, although it is also useful to traverse unknown terrains after extrapolating the gathered information to similar areas of the scenario, making the most of a terrain segmentation map. It is worth mentioning that the used experimental terrain in both the simulation and field tests was limited to only one type of terrain with an initial, intentionally wrong segmentation. So, a new field test in a more relevant terrain is foreseen. Moreover, it is proposed as future work the use of long-range obstacle detection systems, such as panoramic stereo cameras or a LIDAR, to improve the rover hazard detection capability, being able to feedback more information to enrich the cost map.

Once the global path planner, which is the one that gets the rover to the target MM area, was improved with the first contribution, the focus switched to planning the motion of the mobile manipulator to perform the scientific task. Using the same path planner as a base, a robust path and motion planner for MM was researched, which is the second contribution. This algorithm was validated through both simulation and field tests with the SherpaTT rover of DFKI RIC in Galopprennbahn Bremen. During the tests the rover was able to reach the goal avoiding obstacles in the scenario and smoothly orient itself to leave the target inside the manipulator workspace, meanwhile, the arm was performing its particular motion plan to perform a scanning measurement on the sample goal. Besides, a benchmark with an off-the-shelf motion planner was performed, demonstrating the increased efficiency of the coupled approach, and the guarantee of self-collision avoidance without a significant increase in the computational effort. In particular, among the different possible configurations for the motion planner progressive deployment is the one that maximizes the efficiency, since the manipulator movements are completely performed meanwhile the rover is still reaching the goal, and the arm joints' motion is equally distributed throughout the whole rover base trajectory. Although the proposed motion planner has been demonstrated to be comparable to off-the-shelf planners in computational terms, it could be improved with a parallel implementation of some of the motion planning stages, which is foreseen as future work. Additionally, for its use in future space exploration missions, it would be necessary to test the algorithm with real space conditions, i.e. a space-qualified processing unit running an autonomous navigation architecture and including communication delays. Finally, it is foreseen to include on the arm end effector a scientific tool, such as a sample localization subsystem or a spectrometer, to perform a real scientific task on interesting areas of the scenario. This tool, altogether with the inclusion of real space conditions, would allow us to perform a more representative and complete field test campaign, to confirm that the proposed algorithm could be used in future space exploration missions.

Though the robust motion planner generates efficient motions for MM with low computational cost, it is still a sub-optimal method. Therefore, the third contribution explored an optimal methodology for MM motion planning, able to optimize parameters like the energy spent. Called MWMP, this whole-body optimal motion

planner for over-actuated platforms was validated through a performance campaign through simulations and laboratory tests. The proposed generic model for over-actuated platforms including several kinematic chains was particularized for the ExoTeR rover, composed of a mobile base and a robotic arm. This MM rover was later used to perform the laboratory tests. For that purpose, a tailored event-triggered replanning capability was included, which allows the system to precisely follow the generated motion plans. The laboratory tests were performed at the Martian Analogue Testbed of the ESA-PRL at ESTEC, to emulate a Martian sample tube retrieval mission. The tests showcased the advantages of the presented motion planner to generate accurate motions for over-actuated and constrained platforms, in this case, allowing ExoTeR to retrieve a sample tube with a high-quality grasp, even considering the rover and sample localization errors. The proposed multi-staged warm start sequence has been demonstrated to improve the motion planner convergence speed and the feasibility of its solutions w.r.t. any cold-started version of the planner or any other combination of the warm start stages. Nevertheless, it has only been tested with the SLQ solver. Testing if the proposed sequence also boosts other optimization solvers is planned as future work. Furthermore, as done in related works, the inclusion of the constraints sequentially inside the CSLQ stage can boost further the convergence speed, especially for highly constrained and cluttered use cases, which remains to be tested.

MWMP is lightweight enough to solve complex control tasks as demonstrated with the fourth contribution, which is the MM MPC tension controller to support a rappelling rover. This method was validated through a simulation and field test campaign. On the one hand, a realistic simulation environment was developed, which included all of the involved systems: the SherpaTT rover as the MM mobile anchor to support the rappelling, the CoyoteIII rover as the lightweight rappelling rover, and the TMDS as the tethering interface between both rovers. Additionally, the scenario of the field test campaign, located in Lanzarote, was accurately 3D modeled and included in the simulator. The results of the simulations showed that using the MM MPC tension controller ensures the safety of SherpaTT's robotic arm, conversely, maintaining the arm still usually leads to overcoming the arm joints' torque limits during the rappelling operation. This statement was confirmed during the later field trials in Lanzarote, where the rappelling mission was successfully performed several times with the tension controller enabled, without surpassing the torque limits in any of the arm joints. It is expected to use this MM MPC in further use cases, including 3D platforms with faster dynamics and multiple external disturbances. Thus, a benchmark of the optimal motion planner and the MPC with other algorithms, such as an off-the-shelf MPC, is planned as future work.

Chapter 7

Conclusions and future work

"Qui fortis est liber" -
"He who is brave is free"

Lucius Annaeus Seneca
Epistulae Morales ad Lucilium,
1st century A.D.

7.1 Introduction

The capacity to manipulate objects was one of the key biological achievements of human evolution to create our civilization, able to dominate the Earth. Now, humanity is trying to reach new worlds to further expand our civilization, though humans are not biologically prepared to strive in the extreme conditions of extraplanetary surfaces. Robots are, conversely, capable of reaching and exploring other planets in a much safer and cheaper manner. Nevertheless, robots' manipulation capabilities in space are still immature. Manipulation requires not only the proper hardware but also the intelligence to perform the movements autonomously and efficiently. For a robot, intelligence is intimately related to computational capacity, however, space-qualified processors are computationally limited due to the harsh conditions of space.

Within the indicated context, this dissertation exposes the achievements of the author's doctoral research when pushing the boundaries of autonomous MM applied to planetary exploration vehicles. To do so, advancements in motion planning and control for planetary MM rovers were investigated, with four main contributions. In summary, Contribution 1 is a dynamic cost map update procedure, which enhances the global path planner of a planetary rover when approaching an area of scientific interest. Contribution 2 is a robust and efficient motion planner to perform MM scientific tasks in a target area, which generates safe, coupled motions for the robotic arm and the mobile base in a computationally lightweight manner. To compensate for several weaknesses of Contribution 2, an optimal control-based motion planner was explored in Contribution 3. This planner can obtain motions for the whole system (base and arm) that optimize any desired parameter, e.g. minimize the overall energy spent or maximize the manipulability of the platform. Finally, the

knowledge acquired during this doctoral research was applied to solve a complex MM task: a multi-robot rappelling into a Lunar lava tube. To do so, the optimal motion planner of Contribution 3 was integrated into a fast MPC loop to control the tension during the rappelling, which is Contribution 4.

This thesis demonstrates the outcomes of the author's doctoral studies in seven different chapters. Chapter 1 introduces the conducted research and puts it in context. Chapter 2 analyzes the state of the technology regarding autonomous MM. Chapters 3, 4, and 5 depict the methodologies behind Contributions 1, 2, and 3 respectively. Chapter 6 shows the experimental campaigns carried out during this doctoral research and their results, which demonstrate the validity and usefulness of all the contributions, including the rappelling tension controller of Contribution 4. Lastly, this chapter concludes the dissertation with an overview of each contribution in Section 7.2, and a proposal of future research lines in Section 7.3, to continue investigating the topic of autonomous MM in space exploration.

7.2 Conclusions

As thoroughly explained throughout this dissertation, autonomy in space exploration is a huge advantage in terms of time and energy efficiency seeing that the delays caused by teleoperation are mostly removed. Since a great amount of the tasks that a planetary rover executes involve MM, performing autonomous MM motions implies increasing the number and complexity of the tasks that the rover can carry out during its lifetime. Hence, this thesis explores different ways to enable autonomous MM in space robots, by answering how to plan the motion of planetary rovers efficiently, to fulfill MM tasks.

To answer this question, technology from on-Earth applications has been extracted and tailored for the planetary use case, in several steps that establish some contributions to the field. First, by increasing the awareness of the rover regarding the scenario, enhancing the global path planning process to reach the area of scientific interest. Second, by including collision avoidance in an efficient yet robust manner to the motion planning stage, increasing the safety when performing the MM scientific tasks. Third, by optimizing the energy spent without a heavy computational effort within the motion planner, maximizing the overall execution efficiency of the MM motion plan. Fourth, by optimally controlling the motion in multi-robot MM operations, improving the safety of the systems involved. Let us summarize and draw some conclusions from each of these four contributions in the following.

Contribution 1: Dynamic cost map update to enhance global path planning.

Before reaching a spot of scientific interest and performing the MM tasks, a planetary rover usually drives long distances. Thus, achieving an efficient and autonomous MM in space requires first achieving an efficient and autonomous global navigation.

Considering that these traverses usually cover areas with great similarities or even the same areas in round-trip trajectories, the first question of this doctoral research arises: how to increase the rover's awareness of the characteristics of the surrounding scenario to improve future plans?

Contribution 1 is, therefore, a dynamic cost map update procedure to continuously improve the quality of the global planned paths during the traverse. Integrated within the DyMu path planner, this feedback update methodology enhances the rover performance in two main senses. On the one hand, the locally detected obstacles are dynamically included in the cost map, so the rover does not encounter them again seeing that they are directly avoided in the global planned paths. On the other hand, the soil properties estimation is continuously refined during the traverse, modifying the cost of the terrain patches in the cost map. This way, the path planner is more likely to generate trajectories on the areas of the scenario that are easier to traverse.

This dynamic cost map update methodology was validated with an experimental campaign with both simulations and field tests, with the rover HDPR from ESA-PRL. The conclusion from these experiments is that the dynamic feedback to the cost map enhances the global path planner since the rover is more aware of the characteristics of the scenario where it is traversing. It is particularly helpful for round-trip missions, i.e. going to a zone of scientific interest, performing a MM task, and returning to the initial spot. During the return, the rover would cover the same areas, and all the information gathered in the outbound traverse can be used to improve the quality of the return paths. Anyhow, this methodology also enhances the path planner when traversing unknown terrains, since the gathered information can be extrapolated to similar areas if using a global segmentation of the scenario.

Contribution 2: Robust motion planning for MM.

Once the rover has reached the target scientific area, it has to autonomously perform a combined MM motion to fulfill the goal scientific task. However, planning such a motion is computationally expensive given the difficulty of considering collisions if moving both systems (mobile base and robotic arm) simultaneously. This is the second question of this doctoral research: how to consider collision avoidance robustly and efficiently in the motion planning stage of MM?

Contribution 2 is, hence, a robust and efficient motion planning algorithm for planetary exploration MM. This algorithm plans first a path for the mobile base that avoids the obstacles in the scenario until reaching the goal. Then, it plans the motion for the manipulator according to the mobile base path. The generated motion plans always leave the target inside the manipulator workspace since the planner includes a tailored plan-and-control approach in the last stretch of the mobile base path. Additionally, the proposed method is quite lightweight even in the case of a highly restricted arm workspace, since the self-collisions are avoided by using an offline-computed reachability volume of the manipulator.

The robust motion planner was validated within another experimental campaign in simulation and field tests with the rover SherpaTT from DFKI RIC. These experiments demonstrated that the proposed planner is efficient thanks to the coordinated motions of the mobile base and the manipulator. A progressive deployment is, among the possible configurations of the motion planner, the most efficient approach, seeing that it distributes the manipulator movements equally along the rover base trajectory. Besides, the motion planner intrinsically guarantees that the complete plan is safe, with a thorough avoidance of obstacles and self-collisions. This safety is achieved in a computationally efficient manner, thanks to the offline computation of the reachability volume.

Contribution 3: Optimal motion planning for over-actuated mobile platforms.

Although the MM motion planner of Contribution 2 solves the problem robustly and efficiently, it is still a sub-optimal method. As such, it cannot optimize the energy spent during the MM operation, i.e. the motion efficiency is not maximized. Besides, though it coordinates the mobile base and the manipulator, the motion is generated separately. This approach leaves out the possibility of using the DoFs of one of the systems (mobile base or robotic arm) to compensate for the limitations of the other one. Nevertheless, these issues can be directly solved with a whole-body, optimal control-based motion planner, which brings to focus the third question of this doctoral research: can optimal control algorithms be applied to MM motion planning in planetary exploration with admissible computational costs?

Contribution 3 is, thus, an optimal control-based motion planner for over-actuated mobile platforms. Called MWMP, this planner considers the whole system dynamics and its constraints meanwhile maintaining an acceptable computational cost. To do so, several stages of warm-started planning are included, i.e. subsequently initializing the next optimization with a feasible solution to accelerate the convergence speed. Particularly, three warm start stages are implemented in MWMP, a path planner (making the most of the first contribution of this doctoral thesis), an unconstrained SLQ-based motion planner, and a constrained SLQ-based motion planner. To use this motion planner a state space model of the system is required, therefore, a generic model for over-actuated mobile platforms is also presented, which considers systems with multiple kinematic chains.

Before the experimental campaign, the proposed method was evaluated, testing the motion plan quality of MWMP w.r.t. any other combination of the warm starts. These tests proved that the proposed multi-staged warm start sequence improves the performance and reduces the computational cost, with the fastest convergence and the highest success rate. Afterward, the viability of MWMP was confirmed with a laboratory test campaign with the rover ExoTeR from ESA-PRL. In this regard, a motion plan follower was developed to bring the planned motions to the platform. This follower checks for errors in the current state of the platform w.r.t. the planned

motions and requests a new motion plan in case of significant deviations. The laboratory tests confirmed that MWMP generates accurate, constraint-compliant motions for over-actuated platforms, tackling the non-holonomic constraint of the mobile base, filtering the rover and the goal localization errors, and fulfilling the task in an efficient, optimized manner.

Contribution 4: Optimal tension control for multi-robot rappelling.

Not all the MM tasks in space are goal-constrained, i.e. with the rover traversing a scenario to reach a spot of scientific interest and perform a manipulation task. Some operations require a certain level of reactiveness. A clear example is the case of a multi-robot rappelling into a lava tube. Briefly, a big rover acting as a mobile anchor uses its manipulator to support the rappel of a smaller rover through the skylight of the lava tube. Therefore, the manipulator withstands the weight of the small rover as it descends. The rover's weight produces a high tension in the tether that connects both rovers, and this tension generates high torques into the robotic arm's joints, which may damage them if the system does not react on time. Within this context, the fourth question of this doctoral research arises: can optimal MM control increase the safety of multi-robot missions in space?

Contribution 4 is, hereby, an optimal MM controller. Applied to a multi-robot rappelling operation, this controller can compensate for the tension that the tether applies to the robotic arm of the mobile anchor rover as the small rover rappels down to the lava tube. This optimal controller makes use of MWMP from Contribution 3 since it is lightweight enough to be integrated into a MPC, high-frequency loop. To increase the safety of the operation, the proposed MM MPC algorithm continuously measures the efforts (forces and torques) acting on the manipulator's end effector and tries to reduce them by aligning the arm's last link and the tether. This alignment eliminates the torques that the tension generates in the wrist of the manipulator, which has weaker joints, redistributing them to the shoulder and elbow, which have stronger joints.

The MM MPC methodology for tension control was validated with simulations and a field test campaign with the rovers SherpaTT and CoyoteIII from DFKI RIC. The simulation tests showed that enabling the MM MPC component notably reduces the torques endured by the manipulator during the rappelling. On the contrary, maintaining the arm still could lead to a mission failure, since the arm joints suffered torques that surpassed their limits. The usefulness of the proposed controller was confirmed during the field test campaign, seeing that multiple rappelling operations were successfully performed without ever exceeding the torque limits of the robotic arm joints.

7.3 Future work

From a general point of view, the research carried out throughout this doctoral research has enhanced the technology until demonstrating autonomous MM with rover prototypes in analogue environments. Nevertheless, as an academic research work the maximum reached Technology Readiness Level (TRL) is TRL 5 (validating the components with a prototype in a relevant environment). In the future, getting the proposed methods to a real space exploration mission would require reaching at least TRL 6, qualifying the code, and testing the algorithms with real space conditions, i.e. a space-qualified processing unit running an autonomous navigation architecture and including communication delays, localization errors, and energy constraints. In this regard, the algorithms should be adapted to space-grade CPUs like the LEON III or the LEON IV. Additionally, the motion planners should be tested to solve a broader scope of real MM scientific tasks, demanding scientific tools to be attached to the robotic arm end effector, such as a spectrometer, a sample acquisition tool, or a close-range imager. These enhancements would lead to a more representative validation of the proposed methodologies, confirming that they could be brought into future planetary vehicles.

On the other hand, regarding the contributions themselves, there are several potential advancements to the algorithms and alternative research lines to keep improving the autonomous capabilities of MM rovers. Focusing first on the dynamic cost map for global path planning, only two parameters have been used to model the terramechanics of the scenario: the slip ratio and the trafficability. Using a broader amount of parameters would increase the accuracy of the model w.r.t. the real conditions of the terrain, hence, the path planner would find better ways to traverse the scenario. This would be demonstrated in a more relevant field test campaign, on a scenario with multiple types of terrain for the rover to estimate their properties. Moreover, the local obstacle detection should not only rely on the LocCams. Panoramic cameras or laser detectors could also be included to increase the number of detected obstacles, thereby improving the quality of the cost map. Lastly, the cost map used within this work is isotropic, which cannot take into account the movement direction of the rover but only the terrain shape and properties. A cost map that considers anisotropy would allow the inclusion of further aspects that are critical for the rover's safety, such as the risk of overturns or lateral slippage (Sánchez-Ibáñez et al., 2023).

Regarding the MM motion planners, although they are already quite efficient in computational terms (which was the main goal of this doctoral research), their computational cost could be further reduced with parallel computing devices like FPGAs. This has been demonstrated with Perseverance since it has onboarded an FPGA that carries out most of the computer vision tasks (Verma et al., 2023). The vision tasks used to demand considerable processing efforts in the previous rovers, but

are significantly more lightweight in Perseverance thanks to a parallel implementation within the FPGA. Hence, it is foreseen that the proposed robust MM motion planner and the optimal control-based one could also be accelerated with parallel implementations.

Furthermore, the optimal MM motion planner for over-actuated mobile platforms has some room for improvement. In computational terms, though it is not common, there are cases where the constraints are too restrictive and the planner falls into local minima or is not able to converge at all. On-Earth trajectory-constrained optimization algorithms have solved this issue by including the constraints sequentially in small steps (Kabir et al., 2021), i.e. adding even more warm start intermediate stages to the MWMP pipeline. Moreover, the proposed warm start sequence has only been tested with FMM and SLQ as the optimization solvers. Nevertheless, this warm-start approach could accelerate other optimization algorithms too, which remains to be tested. Last of all, more parameters and constraints could be added to the cost function of the optimization, such as the instantaneous power consumption, which, given the available power in the platform, may limit the possible motions to be performed at a certain time.

Finally, though all the existent rovers, to date, use wheels as their main locomotion, new systems are lately taking the spotlight: walking robots like quadrupeds (Arm et al., 2023), or flying robots like the Ingenuity helicopter (Balaram, Aung, and Golombek, 2021). These platforms will also require autonomous MM capabilities when sent to extraterrestrial surfaces. Hereby, a very interesting research line is to generalize the proposed motion planners and the MPC methodology to cope with multiple types of locomotion (wheeled, legged, flying...). To do so, the generic state space model could be further developed to include a wider variety of systems, for instance, closed kinematic chains.

Bibliography

Ali, Khaled S., C. Anthony Vanelli, Jeffrey J. Biesiadecki, Mark W. Maimone, Yang Cheng, A. Miguel San Martin, and James W. Alexander (2005). "Attitude and position estimation on the Mars Exploration Rovers". In: *IEEE International Conference on Systems, Man and Cybernetics*. Vol. 1, pp. 20–27. DOI: 10.1109/ICSMC.2005.1571116.

Almaeeni, Sara, Sebastian Els, Hamad Almarzooqi, Sara Almaeeni, Sebastian Els, and Hamad Almarzooqi (2021). "The Rashid rover: to guide the way for the next generation lunar missions and solar system exploration". In: *EGUGA*, pp. 21–14732. DOI: 10.5194/EGUSPHERE-EGU21-14732.

Anderson, R C, L Jandura, A B Okon, D Sunshine, C Roumeliotis, L W Beegle, J Hurowitz, B Kennedy, D Limonadi, S Mccloskey, M Robinson, C Seybold, and K Brown (2012). "Collecting Samples in Gale Crater, Mars; an Overview of the Mars Science Laboratory Sample Acquisition, Sample Processing and Handling System". In: *Space Sci Rev* 170, pp. 57–75. DOI: 10.1007/s11214-012-9898-9.

Araguz, Carles, Elisenda Bou-Balust, and Eduard Alarcón (Apr. 2018). "Applying autonomy to distributed satellite systems: Trends, challenges, and future prospects". In: *Systems Engineering* 21.5, pp. 401–416. ISSN: 1520-6858. DOI: 10.1002/SYS.21428.

Arm, Philip, Gabriel Waibel, Jan Preisig, Turcan Tuna, Ruyi Zhou, Valentin Bickel, Gabriela Ligeza, Takahiro Miki, Florian Kehl, Hendrik Kolvenbach, and Marco Hutter (July 2023). "Scientific exploration of challenging planetary analog environments with a team of legged robots". In: *Science Robotics* 8.80. ISSN: 2470-9476. DOI: 10.1126/SCIROBOTICS.ADE9548.

Autonomous DEcision making in very long traverses (ADE) (2021). URL: https://www.h2020-ade.eu/.

Avanzini, Giovanni Buizza, Andrea Maria Zanchettin, and Paolo Rocco (Jan. 2018). "Constrained model predictive control for mobile robotic manipulators". In: *Robotica* 36.1, pp. 19–38. ISSN: 0263-5747. DOI: 10.1017/S0263574717000133.

Azkarate, Martin, Levin Gerdes, Luc Joudrier, and Carlos J Perez-del Pulgar (Apr. 2020). "A GNC Architecture for Planetary Rovers with Autonomous Navigation". In: Institute of Electrical and Electronics Engineers (IEEE), pp. 3003–3009. DOI: 10.1109/icra40945.2020.9197122.

Azkarate, Martin, Levin Gerdes, Tim Wiese, Martin Zwick, Marco Pagnamenta, Javier Hidalgo Carrio, Pantelis Poulakis, and Carlos Perez-del Pulgar (Apr. 2022). "Design, Testing, and Evolution of Mars Rover Testbeds: European Space Agency

Planetary Exploration". In: *IEEE Robotics & Automation Magazine*, pp. 2–15. ISSN: 1070-9932. DOI: 10.1109/MRA.2021.3134875.

Azkarate, Martin, Martin Zwick, Javier Hidalgo-Carrio, Robin Nelen, Tim Wiese, Pantelis Poulakis, Luc Joudrier, and Gianfranco Visentin (2015). "First Experimental Investigations on Wheel-Walking for Improving Triple-Bogie Rover Locomotion Performances". In: *13th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA)*.

Balaram, J., Mi Mi Aung, and Matthew P. Golombek (June 2021). "The Ingenuity Helicopter on the Perseverance Rover". In: *Space Science Reviews* 217.4, pp. 1–11. ISSN: 15729672. DOI: 10.1007/S11214-021-00815-W.

Biesiadecki, Jeffrey J. and Mark W. Maimone (2006). "The mars exploration rover surface mobility flight software: Driving ambition". In: *IEEE Aerospace Conference*. Vol. 2006. ISBN: 0780395468. DOI: 10.1109/AERO.2006.1655723.

Bora, Leonardo, Richard Lancaster, Ben Nye, Chris Barclay, Sergio Rubio, and Matthias Winter (2017). "Exomars rover control, localisation and path planning in a hazardous and high disturbance environment". In: *14th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA)*.

Brinkmann, Wiebke, Thomas M Roehr, Sankaranarayanan Natarajan, Florian Cordes, Roland U Sonsalla, Roman Szczuka, Sebastian Bartsch, and Frank Kirchner (Apr. 2018). "Design and evaluation of an end-effector for a reconfigurable multi-robot system for future planetary missions". In: *IEEE Aerospace Conference*. Vol. 2018-March. IEEE Computer Society, pp. 1–10. ISBN: 9781538620144. DOI: 10.1109/AERO.2018.8396415.

Burgess-Limerick, Ben, Chris Lehnert, Jürgen Leitner, and Peter Corke (May 2023). "An Architecture for Reactive Mobile Manipulation On-The-Move". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1623–1629. DOI: 10.1109/ICRA48891.2023.10161021.

Burget, Felix, Maren Bennewitz, and Wolfram Burgard (Apr. 2016). "BI2RRT: An efficient sampling-based path planning framework for task-constrained Mobile manipulation". In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*. Vol. 2016-Novem. Institute of Electrical and Electronics Engineers Inc., pp. 3714–3721. ISBN: 9781509037629. DOI: 10.1109/IROS.2016.7759547.

Camacho, E F and C Bordons (2007). *Model predictive control*. Springer. ISBN: 1852336943.

Carsten, Joseph, Arturo Rankin, Dave Ferguson, and Anthony Stentz (Apr. 2009). "Global planning on the Mars Exploration Rovers: Software integration and surface testing". In: *Journal of Field Robotics* 26.4, pp. 337–357. ISSN: 1556-4967. DOI: 10.1002/ROB.20287.

Castilla-Arquillo, Raúl, Carlos J. Perez-del Pulgar, Gonzalo J. Paz-Delgado, and Levin Gerdes (2022). "Hardware-accelerated Mars Sample Localization via deep transfer learning from photorealistic simulations". In: *IEEE Robotics and Automation Letters*, pp. 1–8. ISSN: 2377-3766. DOI: 10.1109/LRA.2022.3219306.

Colaprete, A., D. Andrews, W. Bluethmann, R. C. Elphic, B. Bussey, J. Trimble, K. Zacny, J. E. Captain, A. Colaprete, D. Andrews, W. Bluethmann, R. C. Elphic, B. Bussey, J. Trimble, K. Zacny, and J. E. Captain (2019). "An Overview of the Volatiles Investigating Polar Exploration Rover (VIPER) Mission". In: *AGUFM* 2019, P34B–03.

CoRob-X - Cooperative Robots for Extreme Environments (2023). URL: https://www.corob-x.eu/.

Danter, Leon, Steffen Planthaber, Alexander Dettmann, Wiebke Brinkmann, and Frank Kirchner (Apr. 2020). "Lightweight and Frameowrk-Independent Communication Library to Support Cross-Platform Robotic Applications and High-Latency Connections". In: *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*. Virtual.

Dettmann, Alexander, Thomas Vögele, Jorge Ocón, Iulia Dragomir, Shashank Govindaraj, Matteo De-Benedetti, Valerie Ciarletti, Rafik Hassen-Khodja, Thierry Germa, Raphael Viards, Gonzalo J. Paz-Delgado, and Laura M Mantoani (2022). "CoRob-X: A cooperative robot team for the exploration of lunar skylights". In: *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*.

Dietrich, Alexander, Thomas Wimböck, Alin Albu-Schäffer, and Gerd Hirzinger (2012). "Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom". In: *IEEE Robotics and Automation Magazine* 19.2, pp. 20–33. ISSN: 10709932. DOI: 10.1109/MRA.2012.2191432.

Dijkstra, E. W. (1959). "A Note on Two Problems in Connexion with Graphs". In: *Numerische Mathematik 1*, pp. 269–271. DOI: 10.1145/3544585.3544600.

Ding, L., R. Zhou, T. Yu, H. Gao, H. Yang, J. Li, Y. Yuan, C. Liu, J. Wang, Y. Y.S. Zhao, Z. Wang, Xiyu Wang, G. Bao, Z. Deng, L. Huang, N. Li, X. Cui, X. He, Y. Jia, B. Yuan, G. Liu, H. Zhang, R. Zhao, Z. Zhang, Z. Cheng, F. Wu, Q. Xu, H. Lu, L. Richter, Z. Liu, F. Niu, H. Qi, S. Li, W. Feng, C. Yang, B. Chen, Z. Dang, M. Zhang, L. Li, Xiaoxue Wang, Z. Huang, J. Zhang, H. Xing, G. Wang, L. Niu, P. Xu, W. Wan, and K. Di (Mar. 2022). "Surface characteristics of the Zhurong Mars rover traverse at Utopia Planitia". In: *Nature Geoscience* 15.3, pp. 171–176. ISSN: 1752-0908. DOI: 10.1038/s41561-022-00905-6.

Dominguez-Durante, Salvador, Carlos J. Perez-Del-Pulgar, Gonzalo J. Paz-Delgado, and Martin Azkarate (2022). "Experimental analysis of slip ratio using the wheel walking locomotion mode in reconfigurable rovers". In: *30th Mediterranean Conference on Control and Automation (MED)*, pp. 749–754. DOI: 10.1109/MED54222.2022.9837116.

Duncan, Robert Clifton and Alf S. Gunnersen (May 1964). "Inertial guidance, navigation, and control systems". In: *Journal of Spacecraft and Rockets* 1.6, pp. 577–587. ISSN: 00224650. DOI: 10.2514/3.27706.

Egerstedt, Magnus and Xiaoming Hu (2000). "Coordinated trajectory following for mobile manipulation". In: *IEEE International Conference on Robotics and Automation (ICRA)* 4, pp. 3479–3484. ISSN: 10504729. DOI: 10.1109/ROBOT.2000.845268.

Farshidian, Farbod, Michael Neunert, Alexander W Winkler, Gonzalo Rey, and Jonas Buchli (Apr. 2017). "An efficient optimal planning and control framework for quadrupedal locomotion". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Institute of Electrical and Electronics Engineers Inc., pp. 93–100. ISBN: 9781509046331. DOI: 10.1109/ICRA.2017.7989016.

Ferguson, Dave and Anthony Stentz (Feb. 2006). "Using interpolation to improve path planning: The Field D* algorithm". In: *Journal of Field Robotics* 23.2, pp. 79–101. ISSN: 1556-4967. DOI: 10.1002/ROB.20109.

Filip, Jan, Martin Azkarate, and Gianfranco Visentin (2017). "Trajectory control for autonomous planetary rovers". In: *14th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*.

Fnadi, Mohamed, Wenqian Du, Frédéric Plumet, and Faïz Benamar (Apr. 2021). "Constrained Model Predictive Control for dynamic path tracking of a bi-steerable rover on slippery grounds". In: *Control Engineering Practice* 107, p. 104693. ISSN: 0967-0661. DOI: 10.1016/J.CONENGPRAC.2020.104693.

Francis, R., T. Estlin, G. Doran, S. Johnstone, D. Gaines, V. Verma, M. Burl, J. Frydenvang, S. Montaño, R. C. Wiens, S. Schaffer, O. Gasnault, L. DeFlores, D. Blaney, and B. Bornstein (June 2017). "AEGIS autonomous targeting for chemcam on mars science laboratory: Deployment and results of initial science team use". In: *Science Robotics* 2.7, p. 4582. ISSN: 24709476. DOI: 10.1126/SCIROBOTICS.AAN4582.

Gao, Yang and Steve Chien (Apr. 2017). *Review on space robotics: Toward top-level science through space exploration*. DOI: 10.1126/scirobotics.aan5074.

García, Carlos E., David M. Prett, and Manfred Morari (May 1989). "Model predictive control: Theory and practice—A survey". In: *Automatica* 25.3, pp. 335–348. ISSN: 0005-1098. DOI: 10.1016/0005-1098(89)90002-2.

Geiger, Andreas, Julius Ziegler, and Christoph Stiller (2011). "StereoScan: Dense 3d reconstruction in real-time". In: *IEEE Intelligent Vehicles Symposium*, pp. 963–968. ISBN: 9781457708909. DOI: 10.1109/IVS.2011.5940405.

Gerdes, Levin, Martin Azkarate, J. Ricardo Sánchez-Ibáñez, Luc Joudrier, and Carlos J Perez-del Pulgar (Apr. 2020). "Efficient autonomous navigation for planetary rovers with limited resources". In: *Journal of Field Robotics* 37.7, pp. 1153–1170. ISSN: 15564967. DOI: 10.1002/rob.21981.

Geromichalos, Dimitrios, Martin Azkarate, Emmanouil Tsardoulias, Levin Gerdes, Loukas Petrou, and Carlos Perez-Del-Pulgar (Apr. 2020). "SLAM for autonomous planetary rovers with global localization". In: *Journal of Field Robotics* 37.5, pp. 830–847. ISSN: 15564967. DOI: 10.1002/rob.21943.

Giftthaler, Markus, Farbod Farshidian, Timothy Sandy, Lukas Stadelmann, and Jonas Buchli (Apr. 2017). "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3411–3417. ISBN: 9781509046331. DOI: 10.1109/ICRA.2017.7989388.

Gomez, Javier V, David Alvarez, Santiago Garrido, and Luis Moreno (2019). "Fast Methods for Eikonal Equations: An Experimental Survey". In: *IEEE Access* 7, pp. 39005–39029. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2906782.

Gonzalez, Ramon, Mirko Fiacchini, and Karl Iagnemma (2018). "Slippage prediction for off-road mobile robots via machine learning regression and proprioceptive sensing". In: *Robotics and Autonomous Systems* 105, pp. 85–93.

Goswami, Namrata (2020). "India's Space Program, Ambitions, and Activities". In: *National Bureau of Asian Research* 15.2, pp. 43–49.

Govindaraj, Shashank, Jeremi Gancet, Diego Urbina, Wiebke Brinkmann, Nabil Aouf, Simon Lacroix, Mateusz Wolski, Francisco Colmenero, Michael Walshe, Cristina Ortega, and Balazs Bodo (2019). "PRO-ACT: Planetary Robots Deployed For Assembly And Construction Of Future Lunar ISRU And Supporting Infrastructures". In: *15th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*.

Grotzinger, John P., Joy Crisp, Ashwin R. Vasavada, Robert C. Anderson, Charles J. Baker, Robert Barry, David F. Blake, Pamela Conrad, Kenneth S. Edgett, Bobak Ferdowski, Ralf Gellert, John B. Gilbert, Matt Golombek, Javier Gómez-Elvira, Donald M. Hassler, Louise Jandura, Maxim Litvak, Paul Mahaffy, Justin Maki, Michael Meyer, Michael C. Malin, Igor Mitrofanov, John J. Simmonds, David Vaniman, Richard V. Welch, and Roger C. Wiens (Sept. 2012). "Mars Science Laboratory mission and science investigation". In: *Space Science Reviews* 170.1-4, pp. 5–56. ISSN: 00386308. DOI: 10.1007/S11214-012-9892-2.

Gwinner, K., F. Scholten, F. Preusker, S. Elgner, T. Roatsch, M. Spiegel, R. Schmidt, J. Oberst, R. Jaumann, and C. Heipke (June 2010). "Topography of Mars from global mapping by HRSC high-resolution digital terrain models and orthoimages: Characteristics and performance". In: *Earth and Planetary Science Letters* 294.3-4, pp. 506–519. ISSN: 0012-821X. DOI: 10.1016/J.EPSL.2009.11.007.

Hart, Peter E., Nils J. Nilsson, and Bertram Raphael (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". In: *IEEE Transactions on Systems Science and Cybernetics* 4.2, pp. 100–107. ISSN: 21682887. DOI: 10.1109/TSSC.1968.300136.

Haviland, Jesse, Niko Sunderhauf, and Peter Corke (Apr. 2022). "A Holistic Approach to Reactive Mobile Manipulation". In: *IEEE Robotics and Automation Letters* 7.2, pp. 3122–3129. ISSN: 23773766. DOI: 10.1109/LRA.2022.3146554.

Hedrick, Gabrielle, Nicholas Ohi, and Yu Gu (2020). "Terrain-Aware Path Planning and Map Update for Mars Sample Return Mission". In: *IEEE Robotics and Automation Letters*, p. 1. ISSN: 2377-3766. DOI: 10.1109/LRA.2020.3005123.

Helmick, Daniel M., Stergios I. Roumeliotis, Yang Cheng, Daniel S. Clouse, Max Bajracharya, and Larry H. Matthies (2006). "Slip-compensated path following for planetary exploration rovers". In: *Advanced Robotics* 20.11, pp. 1257–1280. ISSN: 15685535. DOI: 10.1163/156855306778792470.

Hentout, A. (2011). "Operational trajectory following for wheeled mobile manipulators". In: *Journal of Electrical Engineering* 11.1, pp. 9–9.

Herdt, Andrei, Holger Diedam, Pierre Brice Wieber, Dimitar Dimitrov, Katja Mombaur, and Moritz Diehl (Apr. 2010). "Online Walking Motion Generation with Automatic Footstep Placement". In: *Advanced Robotics* 24.5-6, pp. 719–737. ISSN: 01691864. DOI: 10.1163/016918610X493552.

Hewitt, Robert A, Evangelos Boukas, Martin Azkarate, Marco Pagnamenta, Joshua A Marshall, Antonios Gasteratos, and Gianfranco Visentin (Apr. 2018). "The Katwijk beach planetary rover dataset". In: *The International Journal of Robotics Research* 37.1, pp. 3–12. ISSN: 0278-3649. DOI: 10.1177/0278364917737153.

Huang, Jiunn-Kai and Jessy W. Grizzle (Jan. 2023). "Efficient Anytime CLF Reactive Planning System for a Bipedal Robot on Undulating Terrain". In: *IEEE Transactions on Robotics*, pp. 1–18. ISSN: 1552-3098. DOI: 10.1109/TRO.2022.3228713.

Huang, Sheng Wei, Edward Chen, and Jenhwa Guo (Apr. 2018). "Efficient Seafloor Classification and Submarine Cable Route Design Using an Autonomous Underwater Vehicle". In: *IEEE Journal of Oceanic Engineering* 43.1, pp. 7–18. ISSN: 03649059. DOI: 10.1109/JOE.2017.2686558.

Ichnowski, Jeffrey, Yahav Avigal, Vishal Satish, and Ken Goldberg (Apr. 2020). "Deep learning can accelerate grasp-optimized motion planning". In: *Science Robotics* 5.48. ISSN: 24709476. DOI: 10.1126/scirobotics.abd7710.

Ide, Satoshi, Tomohito Takubo, Kenichi Ohara, Yasushi Mae, and Tatsuo Arai (2011). "Real-time trajectory planning for mobile manipulator using model predictive control with constraints". In: *8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 244–249. DOI: 10.1109/URAI.2011.6145970.

Jiao, Ziyuan, Zeyu Zhang, Xin Jiang, David Han, Song Chun Zhu, Yixin Zhu, and Hangxin Liu (2021). "Consolidating Kinematic Models to Promote Coordinated Mobile Manipulations". In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 979–985. ISBN: 9781665417143. DOI: 10.1109/IROS51168.2021.9636351.

Kabir, Ariyan M, Shantanu Thakar, Rishi K Malhan, Aniruddha V Shembekar, Brual C Shah, and Satyandra K Gupta (Apr. 2021). "Generation of synchronized configuration space trajectories with workspace path constraints for an ensemble of robots". In: *International Journal of Robotics Research* 40.3, pp. 651–678. ISSN: 17413176. DOI: 10.1177/0278364920988087.

Kalakrishnan, Mrinal, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal (2011). "STOMP: Stochastic trajectory optimization for motion planning". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4569–4574. ISSN: 10504729. DOI: 10.1109/ICRA.2011.5980280.

Kassel, Simon (1970). *Lunokhod-1 Soviet Lunar Surface Vehicle*. Tech. rep. Advanced Research Projects Agency.

Kavraki, Lydia E., Petr Švestka, Jean Claude Latombe, and Mark H. Overmars (1996). "Probabilistic roadmaps for path planning in high-dimensional configuration

spaces". In: *IEEE Transactions on Robotics and Automation* 12.4, pp. 566–580. ISSN: 1042296X. DOI: 10.1109/70.508439.

Klamt, Tobias, Max Schwarz, Christian Lenz, Lorenzo Baccelliere, Domenico Buongiorno, Torben Cichon, Antonio DiGuardo, David Droeschel, Massimiliano Gabardi, Malgorzata Kamedula, Navvab Kashiri, Arturo Laurenzi, Daniele Leonardis, Luca Muratore, Dmytro Pavlichenko, Arul S. Periyasamy, Diego Rodriguez, Massimiliano Solazzi, Antonio Frisoli, Michael Gustmann, Jürgen Roßmann, Uwe Süss, Nikos G. Tsagarakis, and Sven Behnke (Aug. 2020). "Remote mobile manipulation with the centauro robot: Full-body telepresence and autonomous operator assistance". In: *Journal of Field Robotics* 37.5, pp. 889–919. ISSN: 1556-4967. DOI: 10.1002/ROB.21895.

Kratky, Vit, Alfonso Alcantara, Jesus Capitan, Petr Stepan, Martin Saska, and Anibal Ollero (Apr. 2021). "Autonomous Aerial Filming with Distributed Lighting by a Team of Unmanned Aerial Vehicles". In: *IEEE Robotics and Automation Letters* 6.4, pp. 7580–7587. ISSN: 23773766. DOI: 10.1109/LRA.2021.3098811.

Kuffner, James J. and Steven M. La Valle (2000). "RRT-connect: an efficient approach to single-query path planning". In: *IEEE International Conference on Robotics and Automation (ICRA)* 2, pp. 995–1001. ISSN: 10504729. DOI: 10.1109/ROBOT.2000.844730.

Lacroix, S., I. K. Jung, and A. Mallet (Nov. 2002). "Digital elevation map building from low altitude stereo imagery". In: *Robotics and Autonomous Systems* 41.2-3, pp. 119–127. ISSN: 0921-8890. DOI: 10.1016/S0921-8890(02)00275-0.

Lai, Jialong, Yi Xu, Roberto Bugiolacchi, Xu Meng, Long Xiao, Minggang Xie, Bin Liu, Kaichang Di, Xiaoping Zhang, Bin Zhou, Shaoxiang Shen, and Luyuan Xu (July 2020). "First look by the Yutu-2 rover at the deep subsurface structure at the lunar farside". In: *Nature Communications 2020 11:1* 11.1, pp. 1–9. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17262-w.

Latombe, Jean-Claude (1991). *Robot Motion Planning*. Boston, MA: Springer US. ISBN: 978-0-7923-9206-4. DOI: 10.1007/978-1-4615-4022-9.

LaValle, S M (1998). *Rapidly-Exploring Random Trees: A New Tool for Path Planning*. Tech. rep.

Lee, Jeongseok, Michael X. Grey, Sehoon Ha, Tobias Kunz, Sumit Jain, Yuting Ye, Siddhartha S. Srinivasa, Mike Stilman, and C Karen Liu (Apr. 2018). "DART: Dynamic Animation and Robotics Toolkit". In: *The Journal of Open Source Software* 3.22, p. 500. ISSN: 2475-9066. DOI: 10.21105/joss.00500.

Lehner, Peter, Sebastian Brunner, Andreas Dömel, Heinrich Gmeiner, Sebastian Riedel, Bernhard Vodermayer, and Armin Wedler (Apr. 2018). "Mobile manipulation for planetary exploration". In: *IEEE Aerospace Conference*. Vol. 2018-March. IEEE Computer Society, pp. 1–11. ISBN: 9781538620144. DOI: 10.1109/AERO.2018.8396726.

Lembono, Teguh Santoso, Antonio Paolillo, Emmanuel Pignat, and Sylvain Calinon (Apr. 2020). "Memory of Motion for Warm-Starting Trajectory Optimization".

In: *IEEE Robotics and Automation Letters* 5.2, pp. 2594–2601. ISSN: 23773766. DOI: 10.1109/LRA.2020.2972893.

Lester, Dan and Harley Thronson (Apr. 2011). "Human space exploration and human spaceflight: Latency and the cognitive scale of the universe". In: *Space Policy* 27.2, pp. 89–93. ISSN: 02659646. DOI: 10.1016/j.spacepol.2011.02.002.

Li, Qinghua, Yaqi Mu, Yue You, Zhao Zhang, and Chao Feng (Apr. 2020). "A Hierarchical Motion Planning for Mobile Manipulator". In: *IEEJ Transactions on Electrical and Electronic Engineering* 15.9, pp. 1390–1399. ISSN: 1931-4973. DOI: 10.1002/tee.23206.

Li, Rongxing, Juwon Hwangbo, Yunhang Chen, and Kaichang Di (July 2011). "Rigorous photogrammetric processing of HiRISE stereo imagery for mars topographic mapping". In: *IEEE Transactions on Geoscience and Remote Sensing* 49.7, pp. 2558–2572. ISSN: 01962892. DOI: 10.1109/TGRS.2011.2107522.

Li, Wei and Rong Xiong (2019). "Dynamical Obstacle Avoidance of Task- Constrained Mobile Manipulation Using Model Predictive Control". In: *IEEE Access* 7, pp. 88301–88311. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2925428.

Li, Weiwei and Emanuel Todorov (2004). "Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems". In: *1st International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, pp. 222–229.

Liao, Jianfeng, Fanghao Huang, Zheng Chen, and Bin Yao (Apr. 2019). "Optimization-based motion planning of mobile manipulator with high degree of kinematic redundancy". In: *International Journal of Intelligent Robotics and Applications* 3.2, pp. 115–130. ISSN: 2366598X. DOI: 10.1007/s41315-019-00090-7.

Liu, Haiqiang, Meibao Yao, Xueming Xiao, and Yonggang Xiong (2023). "RockFormer: A U-Shaped Transformer Network for Martian Rock Segmentation". In: *IEEE Transactions on Geoscience and Remote Sensing* 61. ISSN: 15580644. DOI: 10.1109/TGRS.2023.3235525.

Luis, Carlos E, Marijan Vukosavljev, and Angela P Schoellig (Apr. 2020). "Online Trajectory Generation with Distributed Model Predictive Control for Multi-Robot Motion Planning". In: *IEEE Robotics and Automation Letters* 5.2, pp. 604–611. ISSN: 23773766. DOI: 10.1109/LRA.2020.2964159.

Maimone, Mark (2017). "The evolution of autonomous capabilities on NASA's mars rovers". In: *Southern California Robotics Symposium*.

Maimone, Mark, Yang Cheng, and Larry Matthies (Apr. 2007). "Two years of visual odometry on the Mars Exploration Rovers". In: *Journal of Field Robotics* 24.3, pp. 169–186. ISSN: 15564959. DOI: 10.1002/rob.20184.

Manning, Rob and William L. Simon (2014). *Mars Rover Curiosity: An Inside Account from Curiosity's Chief Engineer*. Ed. by Gregory McNamee. Washington, DC: Smithsonian Books.

Mantoani, Laura M, Raúl Castilla-Arquillo, Gonzalo J. Paz-Delgado, Carlos J. Pérez-Del-Pulgar, and Martin Azkarate (2022). "Samples Detection and Retrieval for

a Sample Fetch Rover". In: *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*.

Manz, Marc, Jens Hilljegerdes, Alexander Dettmann, and Frank Kirchner (2012). "Development of a lightweight manipulator arm using heterogeneous materials and manufacturing technologies". In: *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*.

Matijevic, J. (Apr. 1998). "Autonomous navigation and the Sojourner microrover". In: *Science* 280.5362, pp. 454–455. ISSN: 00368075. DOI: 10.1126/SCIENCE.280.5362.454.

Matijevic, J R, J Crisp, D B Bickler, R S Banes, B K Cooper, H J Eisen, J Gensler, A Haldemann, F Hartman, K A Jewett, L H Matthies, S L Laubach, A H Mishkin, J C Morrison, T T Nguyen, A R Sirota, H W Stone, S Stride, L F Sword, J A Tarsala, A D Thompson, M T Wallace, R Welch, E Wellman, B H Wilcox, D Ferguson, P Jenkins, J Kolecki, G A Landis, D Wilt, H J Moore, and F Pavlics (Apr. 1997). *Characterization of the martian surface deposits by the Mars Pathfinder rover, Sojourner*. DOI: 10.1126/science.278.5344.1765.

Merlo, Andrea, Jonan Larranaga, and Peter Falkner (2013). "Sample Fetching Rover (SFR) for MSR". In: *12h Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*.

Minniti, Maria Vittoria, Farbod Farshidian, Ruben Grandia, and Marco Hutter (Apr. 2019). "Whole-Body MPC for a Dynamically Stable Mobile Manipulator". In: *IEEE Robotics and Automation Letters* 4.4, pp. 3687–3694. ISSN: 23773766. DOI: 10.1109/LRA.2019.2927955.

Moeller, Robert C, Louise Jandura, Keith Rosette, Matt Robinson, Jessica Samuels, Milo Silverman, Kyle Brown, Elizabeth Duffy, Aaron Yazzie, Elizabeth Jens, Iona Brockie, Lauren White, Yulia Goreva, Torsten Zorn, Avi Okon, Justin Lin, Matthew Frost, Curtis Collins, Jeffrey B Williams, Adam Steltzner, Fei Chen, and Jeff Biesiadecki (Apr. 2021). *The Sampling and Caching Subsystem (SCS) for the Scientific Exploration of Jezero Crater by the Mars 2020 Perseverance Rover*. DOI: 10.1007/s11214-020-00783-7.

Neunert, Michael, Cédric De Crousaz, Fadri Furrer, Mina Kamel, Farbod Farshidian, Roland Siegwart, and Jonas Buchli (2016). "Fast nonlinear model predictive control for unified trajectory optimization and tracking". In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1398–1404.

Olson, Clark F., Larry H. Matthies, John R. Wright, Rongxing Li, and Kaichang Di (Jan. 2007). "Visual terrain mapping for Mars exploration". In: *Computer Vision and Image Understanding* 105.1, pp. 73–85. ISSN: 1077-3142. DOI: 10.1016/J.CVIU.2006.08.005.

Oriolo, Giuseppe and Christian Mongillo (2005). "Motion planning for mobile manipulators along given end-effector paths". In: *IEEE International Conference on Robotics and Automation (ICRA)* 2005, pp. 2154–2160. ISSN: 10504729. DOI: 10.1109/ROBOT.2005.1570432.

Pajak, Grzegorz and Iwona Pajak (2017). "Point-to-point collision-free trajectory planning for mobile manipulators". In: *Journal of Intelligent & Robotic Systems* 85.3, pp. 523–538.

Pankert, Johannes and Marco Hutter (Oct. 2020). "Perceptive model predictive control for continuous mobile manipulation". In: *IEEE Robotics and Automation Letters* 5.4, pp. 6177–6184. ISSN: 23773766. DOI: 10.1109/LRA.2020.3010721.

Paz-Delgado, Gonzalo J., Martin Azkarate, J. Ricardo Sanchez-Ibanez, Carlos J. Perez-Del-Pulgar, Levin Gerdes, and Alfonso J Garcia-Cerezo (2020). "Improving autonomous rover guidance in round-trip missions using a dynamic cost map". In: *IEEE International Conference on Intelligent Robots and Systems (IROS)*. DOI: 10.1109/IROS45743.2020.9340912.

Paz-Delgado, Gonzalo J., Carlos J. Pérez-del Pulgar, Martin Azkarate, Frank Kirchner, and Alfonso García-Cerezo (Apr. 2023a). "Multi-stage warm started optimal motion planning for over-actuated mobile platforms". In: *Intelligent Service Robotics 2023*, pp. 1–17. ISSN: 1861-2784. DOI: 10.1007/S11370-023-00461-X.

Paz-Delgado, Gonzalo J., J. Ricardo Sanchez-Ibanez, Raúl Dominguez, Carlos J. Perez-Del-Pulgar, Frank Kirchner, and Alfonso Garcia-Cerezo (2023b). "Combined path and motion planning for workspace restricted mobile manipulators in planetary exploration". In: *IEEE Access*. ISSN: 21693536. DOI: 10.1109/ACCESS.2023.3298980.

Pérez-Del-Pulgar, Carlos J., Pablo Romeo-Manrique, Gonzalo J. Paz-Delgado, José Ricardo Sánchez-Ibáñez, and Martin Azkarate (July 2019). "Choosing the Best Locomotion Mode in Reconfigurable Rovers". In: *Electronics* 8.7, p. 818. ISSN: 2079-9292. DOI: 10.3390/electronics8070818.

Pérez-Del-Pulgar, Carlos J., J. Ricardo Sanchez-Ibáñez, Andrés J. Sánchez, Martin Azkarate, and Gianfranco Visentin (Apr. 2017). "Path planning for reconfigurable rovers in planetary exploration". In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. Institute of Electrical and Electronics Engineers Inc., pp. 1453–1458. ISBN: 9781509059980. DOI: 10.1109/AIM.2017.8014223.

Petrović, Luka, Juraj Peršić, Marija Seder, and Ivan Marković (Nov. 2020). "Cross-entropy based stochastic optimization of robot trajectories using heteroscedastic continuous-time Gaussian processes". In: *Robotics and Autonomous Systems* 133, p. 103618. ISSN: 0921-8890. DOI: 10.1016/J.ROBOT.2020.103618.

Pilania, Vinay and Kamal Gupta (Apr. 2018). "Mobile manipulator planning under uncertainty in unknown environments". In: *The International Journal of Robotics Research* 37.2-3, pp. 316–339. ISSN: 0278-3649. DOI: 10.1177/0278364918754677.

Porges, Oliver, Theodoros Stouraitis, Christoph Borst, and Maximo A Roa (2014). "Reachability and capability analysis for manipulation tasks". In: *Advances in Intelligent Systems and Computing* 253, pp. 703–718. ISSN: 21945357.

Quigley, Morgan, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng (2009). "ROS: an open-source Robot Operating System". In: *ICRA Workshop on Open Source Software* 3, p. 5.

Rankin, Arturo, Tyler Del Sesto, Pauline Hwang, Heather Justice, Mark Maimone, Vandi Verma, and Evan Graser (2023). "Perseverance Rapid Traverse Campaign". In: *IEEE Aerospace Conference* 2023-March. ISSN: 1095323X. DOI: 10.1109/AERO55745.2023.10115835.

Rankin, Arturo, Mark Maimone, Jeffrey Biesiadecki, Nikunj Patel, Dan Levine, and Olivier Toupet (Aug. 2021). "Mars curiosity rover mobility trends during the first 7 years". In: *Journal of Field Robotics* 38.5, pp. 759–800. ISSN: 1556-4967. DOI: 10.1002/ROB.22011.

Rothrock, Brandon, Ryan Kennedy, Chris Cunningham, Jeremie Papon, Matthew Heverly, and Masahiro Ono (2016). "Spoc: Deep learning-based terrain classification for mars rover missions". In: *AIAA SPACE*, p. 5539.

Sánchez-Ibáñez, J. Ricardo, Raúl Dominguez, Florian Cordes, and Carlos J Pérez-del Pulgar (2020). "Enhancing Mobile Manipulation with Synchronized Arm-Locomotion Control". In: *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*. Pasadena, California, U.S (Virtual).

Sánchez-Ibáñez, J. Ricardo, Gonzalo J. Paz-Delgado, Pablo Romeo-Manrique, Carlos J. Pérez-Del-Pulgar, and Martin Azkarate (2019a). "Coupled Path and Motion Planning for a Rover-Manipulator System". In: *15th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA)*.

Sánchez-Ibáñez, J. Ricardo, Carlos J. Pérez-Del-pulgar, and Alfonso García-Cerezo (Nov. 2021). "Path Planning for Autonomous Mobile Robots: A Review". In: *Sensors* 21.23, p. 7898. ISSN: 1424-8220. DOI: 10.3390/S21237898.

Sánchez-Ibáñez, J. Ricardo, Carlos J. Pérez-Del-Pulgar, Javier Serón, and Alfonso García-Cerezo (Mar. 2023). "Optimal path planning using a continuous anisotropic model for navigation on irregular terrains". In: *Intelligent Service Robotics* 16.1, pp. 19–32. ISSN: 18612784. DOI: 10.1007/S11370-022-00450-6.

Sánchez-Ibáñez, J. Ricardo, Carlos J. Pérez-del Pulgar, Martin Azkarate, Levin Gerdes, and Alfonso García-Cerezo (Apr. 2019b). "Dynamic path planning for reconfigurable rovers using a multi-layered grid". In: *Engineering Applications of Artificial Intelligence* 86, pp. 32–42. ISSN: 09521976. DOI: 10.1016/j.engappai.2019.08.011.

Sánchez-Ibáñez, José Ricardo (2022). "Extreme Path Planning for Exploration Mobile Robots". PhD thesis. Universidad de Málaga.

Sandakalum, Thushara and Marcelo H Ang (Apr. 2022). "Motion Planning for Mobile Manipulators: A Systematic Review". In: *Machines* 10.2, p. 97. ISSN: 2075-1702. DOI: 10.3390/MACHINES10020097.

Schulman, John, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel (Aug. 2014). "Motion planning with sequential convex optimization and convex collision checking".

In: *International Journal of Robotics Research* 33.9, pp. 1251–1270. ISSN: 17413176. DOI: 10.1177/0278364914528132.

Schuster, Martin J., Christoph Brand, Sebastian G. Brunner, Peter Lehner, Josef Reill, Sebastian Riedel, Tim Bodenmüller, Kristin Bussmann, Stefan Büttner, Andreas Dömel, Werner Friedl, Iris Grixa, Matthias Hellerer, Heiko Hirschmüller, Michael Kassecker, Zoltán Csaba Márton, Christian Nissler, Felix Ruess, Michael Suppa, and Armin Wedler (Dec. 2016). "The LRU Rover for Autonomous Planetary Exploration and Its Success in the SpaceBotCamp Challenge". In: *International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pp. 7–14. DOI: 10.1109/ICARSC.2016.62.

Schuster, Martin J., Marcus G. Muller, Sebastian G. Brunner, Hannah Lehner, Peter Lehner, Ryo Sakagami, Andreas Domel, Lukas Meyer, Bernhard Vodermayer, Riccardo Giubilato, Mallikarjuna Vayugundla, Josef Reill, Florian Steidle, Ingo Von Bargen, Kristin Bussmann, Rico Belder, Philipp Lutz, Wolfgang Sturzl, Michal Smisek, M. Moritz, Samantha Stoneman, Andre Fonseca Prince, Bernhard Rebele, Maximilian Durner, Emanuel Staudinger, Siwei Zhang, Robert Pohlmann, Esther Bischoff, Christian Braun, Susanne Schroder, Enrico Dietz, Sven Frohmann, Anko Borner, Heinz Wilhelm Hubers, Bernard Foing, Rudolph Triebel, Alin O. Albu-Schaffer, and Armin Wedler (Oct. 2020). "The ARCHES Space-Analogue Demonstration Mission: Towards Heterogeneous Teams of Autonomous Robots for Collaborative Scientific Sampling in Planetary Exploration". In: *IEEE Robotics and Automation Letters* 5.4, pp. 5315–5322. ISSN: 23773766. DOI: 10.1109/LRA.2020.3007468.

Sethian, J A (1999). "Fast Marching Methods". In: *SIAM Review* 41.2, pp. 199–235. ISSN: 00361445. DOI: 10.1137/S0036144598347059.

Severny, A. B., E. I. Terez, and A. M. Zvereva (Sept. 1975). "The measurements of sky brightness on lunokhod-2". In: *The Moon* 14.1, pp. 123–128. ISSN: 00270903. DOI: 10.1007/BF00562978.

Shao, Jun, Hao Xiong, Jianfeng Liao, Wei Song, Zheng Chen, Jason Gu, and Shiqiang Zhu (Apr. 2021). "RRT-GoalBias and path smoothing based motion planning of mobile manipulators with obstacle avoidance". In: *IEEE International Conference on Real-Time Computing and Robotics (RCAR)*, pp. 217–222. DOI: 10.1109/RCAR52367.2021.9517335.

Shum, Alex, Kirsten Morris, and Amir Khajepour (Sept. 2016). "Convergence Rate for the Ordered Upwind Method". In: *Journal of Scientific Computing* 68.3, pp. 889–913. ISSN: 08857474. DOI: 10.1007/S10915-016-0163-3.

Siciliano, Bruno, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo (2009). *Robotics: Modelling, planning and control*. 1st ed. 9781846286414. Springer London. ISBN: 978-1-84628-641-4. DOI: 10.1007/978-1-84628-642-1.

Sideris, Athanasios and James E Bobrow (Apr. 2005). "An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems". In: *IEEE*

Transactions on Automatic Control 50.12, pp. 2043–2047. ISSN: 00189286. DOI: 10.1109/TAC.2005.860248.

Sideris, Athanasios and Luis A Rodriguez (Apr. 2011). "A Riccati approach for constrained linear quadratic optimal control". In: *International Journal of Control* 84.2, pp. 370–380. ISSN: 0020-7179. DOI: 10.1080/00207179.2011.555883.

Siegel, Mel (2003). "The sense-think-act paradigm revisited". In: *1st IEEE International Workshop on Robotic Sensing (ROSE): Sensing and Perception in 21st Century Robotics*, pp. 5–9. DOI: 10.1109/ROSE.2003.1218700.

Sleiman, Jean Pierre, Farbod Farshidian, Maria Vittoria Minniti, and Marco Hutter (Apr. 2021). "A Unified MPC Framework for Whole-Body Dynamic Locomotion and Manipulation". In: *IEEE Robotics and Automation Letters* 6.3, pp. 4688–4695. ISSN: 23773766. DOI: 10.1109/LRA.2021.3068908.

Song, Yunlong, Angel Romero, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza (Sept. 2023). "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning". In: *Science robotics* 8.82, eadg1462. ISSN: 24709476. DOI: 10.1126/SCIROBOTICS.ADG1462.

St. Pierre, Ryan and Sarah Bergbreiter (2019). "Toward autonomy in sub-gram terrestrial robots". In: *Annual Review of Control, Robotics, and Autonomous Systems* 2, pp. 231–252.

Stachniss, Cyrill, John J. Leonard, and Sebastian Thrun (2016). "Simultaneous Localization and Mapping". In: *Springer Handbooks*, pp. 1153–1176. ISSN: 25228706. DOI: 10.1007/978-3-319-32552-1_46.

Takei, Ryo and Richard Tsai (Apr. 2013). "Optimal trajectories of curvature constrained motion in the Hamilton-Jacobi formulation". In: *Journal of Scientific Computing* 54.2-3, pp. 622–644. ISSN: 08857474. DOI: 10.1007/s10915-012-9671-y.

Tao, Yu, Jan Peter Muller, and William Poole (Dec. 2016). "Automated localisation of Mars rovers using co-registered HiRISE-CTX-HRSC orthorectified images and wide baseline Navcam orthorectified mosaics". In: *Icarus* 280, pp. 139–157. ISSN: 0019-1035. DOI: 10.1016/J.ICARUS.2016.06.017.

Thakar, Shantanu, Rishi K Malhan, Prahar M Bhatt, and Satyandra K Gupta (Apr. 2022). "Area-Coverage Planning for Spray-based Surface Disinfection with a Mobile Manipulator". In: *Robotics and Autonomous Systems* 147. ISSN: 09218890. DOI: 10.1016/J.ROBOT.2021.103920.

Thakar, Shantanu, Pradeep Rajendran, Vivek Annem, Ariyan Kabir, and Satyandra Gupta (Apr. 2019). "Accounting for part pose estimation uncertainties during trajectory generation for part pick-up using mobile manipulators". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 2019-May, pp. 1329–1336. ISBN: 9781538660263. DOI: 10.1109/ICRA.2019.8793501.

Thakar, Shantanu, Srivatsan Srinivasan, Sarah Al-Hussaini, Prahar M. Bhatt, Pradeep Rajendran, Yeo Jung Yoon, Neel Dhanaraj, Rishi K. Malhan, Matthias Schmid, Venkat N. Krovi, and Satyandra K. Gupta (Apr. 2023). "A Survey of Wheeled Mobile Manipulation: A Decision-Making Perspective". In: *Journal of Mechanisms*

and Robotics 15.2, pp. 20801–20802. ISSN: 19424310. DOI: 10.1115/1.4054611/1141071.

Todorov, Emanuel and Weiwei Li (2005). "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems". In: *American Control Conference*. Vol. 1, pp. 300–306. DOI: 10.1109/ACC.2005.1469949.

Toupet, Olivier, Jeffrey Biesiadecki, Arturo Rankin, Amanda Steffy, Gareth Meirion-Griffith, Dan Levine, Maximilian Schadegg, and Mark Maimone (Aug. 2020). "Terrain-adaptive wheel speed control on the Curiosity Mars rover: Algorithm and flight results". In: *Journal of Field Robotics* 37.5, pp. 699–728. ISSN: 1556-4967. DOI: 10.1002/ROB.21903.

Vago, Jorge L, Frances Westall, Andrew J Coates, Ralf Jaumann, Oleg Korablev, Valérie Ciarletti, Igor Mitrofanov, Jean Luc Josset, Maria Cristina De Sanctis, Jean Pierre Bibring, Fernando Rull, Fred Goesmann, Harald Steininger, Walter Goetz, William Brinckerhoff, Cyril Szopa, François Raulin, Howell G M Edwards, Lyle G Whyte, Alberto G Fairén, John Bridges, Ernst Hauber, Gian Gabriele Ori, Stephanie Werner, Damien Loizeau, Ruslan O Kuzmin, Rebecca M E Williams, Jessica Flahaut, François Forget, Daniel Rodionov, Håkan Svedhem, Elliot Sefton-Nash, Gerhard Kminek, Leila Lorenzoni, Luc Joudrier, Viktor Mikhailov, Alexander Zashchirinskiy, Sergei Alexashkin, Fabio Calantropio, Andrea Merlo, Pantelis Poulakis, Olivier Witasse, Olivier Bayle, Silvia Bayón, Uwe Meierhenrich, John Carter, Juan Manuel García-Ruiz, Pietro Baglioni, Albert Haldemann, Andrew J Ball, André Debus, Robert Lindner, Frédéric Haessig, David Monteiro, Roland Trautner, Christoph Voland, Pierre Rebeyre, Duncan Goulty, Frédéric Didot, Stephen Durrant, Eric Zekri, Detlef Koschny, Andrea Toni, Gianfranco Visentin, Martin Zwick, Michel Van Winnendael, Martín Azkarate, and Christophe Carreau (Apr. 2017). *Habitability on Early Mars and the Search for Biosignatures with the ExoMars Rover*. DOI: 10.1089/ast.2016.1533.

Vasilopoulos, Vasileios, Yiannis Kantaros, George J. Pappas, and Daniel E. Koditschek (2021). "Reactive Planning for Mobile Manipulation Tasks in Unexplored Semantic Environments". In: *IEEE International Conference on Robotics and Automation (ICRA)* 2021-May, pp. 6385–6392. ISSN: 10504729. DOI: 10.1109/ICRA48506.2021.9561958.

Vatavuk, Ivo, Goran Vasiljevic, and Zdenko Kovacic (Mar. 2022). "Task Space Model Predictive Control for Vineyard Spraying with a Mobile Manipulator". In: *Agriculture* 12.3, p. 381. DOI: 10.3390/AGRICULTURE12030381.

Verma, Vandi, Mark W. Maimone, Daniel M. Gaines, Raymond Francis, Tara A. Estlin, Stephen R. Kuhn, Gregg R. Rabideau, Steve A. Chien, Michael M. McHenry, Evan J. Graser, Arturo L. Rankin, and Ellen R. Thiel (July 2023). "Autonomous robotics is driving Perseverance rover's progress on Mars". In: *Science robotics* 8.80, eadi3099. ISSN: 24709476. DOI: 10.1126/SCIROBOTICS.ADI3099.

Volpe, Richard, Tara Estlin, Sharon Laubach, Clark Olson, and J. Balaram (2000). "Enhanced Mars rover navigation techniques". In: *IEEE International Conference on Robotics and Automation (ICRA)* 1, pp. 926–931. ISSN: 10504729. DOI: 10.1109/ROBOT.2000.844167.

Wallace, Richard, Anthony Stentz, Charles Thorpe, Hans Moravec, William Whittaker, and Takeo Kanade (1985). "First Results in Robot Road-Following". In: *IJCAI*, pp. 1089–1095.

Wang, Cong, Yin Zhang, Yongqiang Zhang, Rui Tian, and Mingli Ding (2021). "Mars image super-resolution based on generative adversarial network". In: *IEEE Access* 9, pp. 108889–108898. ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3101858.

Williford, Kenneth H, Kenneth A Farley, Kathryn M Stack, Abigail C Allwood, David Beaty, Luther W Beegle, Rohit Bhartia, Adrian J Brown, Manuel de la Torre Juarez, Svein-Erik Hamran, Michael H Hecht, Joel A Hurowitz, Jose A Rodriguez-Manfredi, Sylvestre Maurice, Sarah Milkovich, and Roger C Wiens (Apr. 2018). "The NASA Mars 2020 Rover Mission and the Search for Extraterrestrial Life". In: *From Habitability to Life on Mars*. Elsevier, pp. 275–308. DOI: 10.1016/b978-0-12-809935-3.00010-4.

Winter, Matthias, Sergio Rubio, Richard Lancaster, Chris Barclay, Nuno Silva, Ben Nye, and Leonardo Bora (2017). "Detailed description of the high-level autonomy functionalities developed for the ExoMars rover". In: *14th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*.

Wright, Jack, Alexander M. Barrett, Peter Fawdon, Elena A. Favaro, Matthew R. Balme, Mark J. Woods, and Spyros Karachalios (2022). "Jezero crater, Mars: application of the deep learning NOAH-H terrain classification system". In: *Journal of Maps* 18.2, pp. 484–496. ISSN: 17445647. DOI: 10.1080/17445647.2022.2095935.

Xiao, Long (May 2014). "China's touch on the Moon". In: *Nature Geoscience 2014 7:6* 7.6, pp. 391–392. ISSN: 1752-0908. DOI: 10.1038/ngeo2175.

YAOKI - Japanese Lunar Rover (2023). URL: https://dymon.co.jp/en/.

Younes, Younes Al and Martin Barczyk (Apr. 2021). "Nonlinear Model Predictive Horizon for Optimal Trajectory Generation". In: *Robotics* 10.3, p. 90. DOI: 10.3390/ROBOTICS10030090.

Yuan, Wang, Yong Hua Liu, Chun Yi Su, and Feng Zhao (Mar. 2023). "Whole-Body Control of an Autonomous Mobile Manipulator Using Model Predictive Control and Adaptive Fuzzy Technique". In: *IEEE Transactions on Fuzzy Systems* 31.3, pp. 799–809. ISSN: 19410034. DOI: 10.1109/TFUZZ.2022.3189808.

Zeng, Jun, Bike Zhang, Zhongyu Li, and Koushil Sreenath (May 2021). "Safety-Critical Control using Optimal-decay Control Barrier Function with Guaranteed Point-wise Feasibility". In: *American Control Conference*. Vol. 2021-May. Institute of Electrical and Electronics Engineers Inc., pp. 3856–3863. ISBN: 9781665441971. DOI: 10.23919/ACC50511.2021.9482626.

Zhao, Hongkai (Apr. 2004). "A fast sweeping method for Eikonal equations". In: *Mathematics of Computation* 74.250, pp. 603–628. ISSN: 0025-5718. DOI: 10.1090/s0025-5718-04-01678-3.

Resumen de la tesis doctoral

El papel de la robótica en la ciencia espacial

El espacio es el principal interrogante que limita el conocimiento humano sobre el universo y el progreso tecnológico. Explorar el espacio es difícil, costoso y peligroso para los humanos. Por ello, los robots han surgido en las últimas décadas como la alternativa más adecuada, llegando más lejos, con menos recursos invertidos y sin arriesgar vidas humanas. En particular, los vehículos de exploración planetaria, también llamados rovers, se han convertido en la mejor opción para llevar los instrumentos científicos que recogen información de superficies extraterrestres como Marte (Gao and Chien, 2017) o la Luna. Normalmente instalados en un brazo robótico, estos instrumentos permiten que el rover ejecute tareas de Manipulación Móvil o Mobile Manipulation (MM), como realizar mediciones científicas en la superficie (Lehner et al., 2018), extraer muestras de terreno Marciano (Williford et al., 2018), recoger los tubos con muestras de terreno (Merlo, Larranaga, and Falkner, 2013), o incluso colaborar entre rovers (Brinkmann et al., 2018). Este tipo de robot se denomina manipulador móvil, es decir, un brazo robótico sin limitaciones en su espacio de trabajo gracias a la flexibilidad que le proporciona la plataforma móvil.

No obstante, los rovers Marcianos tienen un inconveniente principal: la teleoperación desde la Tierra. Esto se debe a la corta duración de las ventanas de comunicación (16 h al día, retrasándose 40 min cada día) y la inmensa latencia (20 min de media) entre la Tierra y Marte (Lester and Thronson, 2011). Teleoperar un rover en Marte es, por tanto, una tarea increíblemente difícil, con alto riesgo y muy baja eficiencia. Consecuentemente, la alternativa a la teleoperación es incluir en el rover un cierto nivel de autonomía, de manera que el rover reaccione a situaciones inesperadas o incluso tome decisiones de medio-alto nivel por sí mismo. Esta capacidad generalmente se denomina navegación autónoma. Aunque no tuvo mucha relevancia en las primeras misiones de exploración planetaria, la navegación autónoma ha ido poco a poco ganando importancia en las últimas décadas. De hecho, en el último rover Marciano de la National Aeronautics and Space Administration (NASA), Perseverance, el llamado Autonomous Navigation (AutoNav) está alcanzando un increíble 78 % de distancia recorrida de forma autónoma en Marte, durante su primer año Marciano (Verma et al., 2023). Gracias a este nivel tan elevado de autonomía, la distancia recorrida ha aumentado unos 144 m de media por Sol Marciano (un sol Marciano es un día solar en Marte, el cual dura 24 h y 39 min aproximadamente) en comparación con las misiones anteriores, las cuales generalmente no podían recorrer

más de 30 m/sol. Estos logros son impresionantes, los cuales han sido alcanzados gracias a un esfuerzo progresivo en investigación y experimentación en los rovers anteriores. Por ello, revisitemos la evolución de los rovers y su autonomía a lo largo de la historia de la exploración planetaria.

Navegación autónoma en superficies extraterrestres

El concepto de explorar superficies extraterrestres con robots fue propuesto inicialmente con el programa Lunokhod (Caminador Lunar) de la Unión Soviética, siendo Lunokhod 1 el primer vehículo no tripulado recorriendo una superficie fuera de la Tierra (Kassel, 1970). Este rover alunizó en 1970 y recorrió 10.54 km durante 11 días Lunares, es decir, 321 días en la Tierra. Su sucesor fue el rover Lunokhod 2 (Severny, Terez, and Zvereva, 1975), que alunizó en 1973 y recorrió aproximadamente 39 km. Ambos rovers fueron completamente teleoperados gracias a la pequeña latencia entre la Luna y la Tierra (aproximadamente 3 s). Los Lunokhod lograron captar miles de imágenes de la superficie Lunar y también realizaron múltiples análisis del terreno mediante espectrómetros, penetrómetros y telémetros láser.

No fue hasta 25 años después que otro robot alcanzó con éxito otro cuerpo celeste, con la misión de la NASA/Jet Propulsion Laboratory (JPL) Mars Pathfinder. Esta misión fue una prueba de concepto sobre la exploración Marciana con robots, concepto que fue confirmado cuando la Carl Sagan Memorial Station amartizó en 1997 con el rover Sojourner (Matijevic et al., 1997). Aunque no puedo recorrer más de 100 m, Sojourner ya incluía algunos algoritmos básicos de navegación autónoma, lo que supuso la primera demostración real de este tipo de tecnología. En particular, Sojourner incluía un método "Go to Waypoint", el cual permitía a los teleoperadores comandar una serie de puntos que el rover debía alcanzar mediante líneas rectas. Durante la ejecución, el rover utilizaba la odometría de las ruedas para localizarse en el escenario, y una serie de sensores para detectar los obstáculos cercanos y evitarlos (Matijevic, 1998).

Más adelante, los abrumadores logros de los Mars Exploration Rover (MER)s de la NASA (Maimone, Cheng, and Matthies, 2007), Spirit y Opportunity, demostraron que los vehículos de exploración planetaria garantizan un amplio retorno científico. Spirit y Opportunity inicialmente tenían una vida útil de 90 días cuando amartizaron en 2004, la cual fue ampliamente superada: Spirit recorrió la superficie Marciana durante seis años, y Opportunity durante 15. Estos rovers gemelos eran bastante más avanzados que Sojourner, con un procesador y una locomoción diez veces más rápidos (Rankin et al., 2023). También incluían mejores métodos de navegación autónoma con el denominado AutoNav, con tres características principales. Primero, odometría visual para la localización, la cual proporcionaba una convergencia exitosa en el 95 % de las actualizaciones en la posición del rover (Maimone, Cheng, and Matthies, 2007). Segundo, evaluación del terreno, la cual usaba cámaras estéreo para detectar y estimar la posición de los peligros cercanos. Tercero, un selector de caminos, que elegía la mejor trayectoria para alcanzar un objetivo entre una

serie de caminos distribuidos siguiendo una Gaussiana (Biesiadecki and Maimone, 2006). Spirit y Opportunity utilizaron AutoNav para recorrer de forma autónoma 1825 m y 2433 m respectivamente, es decir, el 23.60 % y 5.40 % de toda la distancia que recorrieron en Marte.

Tras el gran éxito de los MERs, el nuevo objetivo fue realizar tareas científicas más ambiciosas en la superficie Marciana. En concreto, la misión Mars Science Laboratory (MSL) de la NASA fue lanzada en 2012 con el rover Curiosity (Anderson et al., 2012). Este rover alcanzó la superficie de Marte en el cráter Gale, donde sigue explorando hoy. Curiosity tiene un AutoNav parecido al de los MERs pero con un procesador seis veces más rápido. Sin embargo, las capacidades de navegación autónoma de Curiosity no destacan demasiado, realizando de forma autónoma solo un 3.77 % de la distancia total que ha recorrido en Marte. A pesar de tener un equipamiento mucho mejor, esta distancia es menor que la de los MERs, lo cual se debe a los estrictos requerimientos científicos de Curiosity.

Analicemos cómo los teleoperadores planifican y comandan los movimientos de Curiosity cada sol Marciano (Manning and Simon, 2014). En primer lugar, hay un equipo que toma las decisiones de alto nivel denominado el Science Operations Working Group (SOWG). Este equipo está compuesto de científicos, que proponen distintas tareas a ser desarrolladas por el rover, y los ingenieros del rover, que estiman qué tareas son posibles teniendo en cuenta el tiempo y la energía que necesitan. Los miembros del SOWG iteran varias veces, analizando todas las opciones hasta que deciden el plan de exploración para el siguiente sol Marciano. Sin embargo, el SOWG debe terminar con suficiente tiempo antes de la medianoche Marciana. Si no, los Rover Planner (RP)s no podrán utilizar el plan diseñado por el SOWG y traducirlo en una serie detallada de acciones de bajo nivel para el rover. Estas acciones incluyen, por un lado, comandos de movimiento para alcanzar puntos específicos del escenario, los cuales son seleccionados por los RPs tras observar imágenes orbitales y locales del rover. Por otro lado, los comandos de movimiento dan paso a las tareas científicas, con comandos específicos para el brazo robótico y para los instrumentos científicos. La secuencia completa de comandos se comprueba varias veces, finalmente, en un simulador. Con todo esto, para cada sol Marciano, los RPs necesitan aproximadamente seis horas para traducir el plan del SOWG a una lista detallada de comandos para el rover. Por último, temprano en la mañana Marciana, una vez que Curiosity está despierto y funcionando de forma nominal, la lista de instrucciones es enviada. Así, Curiosity empieza a seguir los comandos sin comunicarse más con los teleoperadores hasta que finaliza el sol Marciano. Es importante tener en cuenta que el rover no sabe cómo reaccionar si algún comando sale mal. En ese caso, Curiosity deja de seguir las instrucciones hasta que recibe una nueva lista de comandos, es decir, se pierde por completo el tiempo de trabajo disponible para Curiosity durante ese sol Marciano.

Como conclusión, aunque Curiosity tiene un AutoNav bastante avanzado para evitar obstáculos y alcanzar objetivos en el escenario, la distancia real que el rover

es capaz de recorrer de forma autónoma está muy restringida. Por un lado, debido a que necesita teleoperación directa para planificar las tareas, seleccionar los objetivos científicos, y realizar las tareas científicas. Pero también, por otro lado, debido a la limitada capacidad computacional disponible a bordo, que es utilizada en su mayor parte para mantener activos y en forma todos los sistemas robóticos y científicos del rover.

Los inconvenientes en la autonomía de Curiosity se resolvieron más adelante con las impresionantes capacidades de navegación autónoma del rover Perseverance, el cual amartizó en 2021. Perseverance es el primer paso de Mars Sample Return (MSR), el cual es probablemente el programa de exploración planetaria más ambicioso hasta la fecha. MSR está compuesto de varias misiones con el objetivo final de traer de vuelta a la Tierra varias muestras de terreno Marciano, lo cual es muy interesante desde un punto de vista científico, ya que estas muestras podrían ser analizadas en la Tierra en laboratorios con equipamiento muy avanzado, mucho más avanzado del que puede ser enviado a Marte. Perseverance es muy similar a Curiosity, heredando gran parte de su tecnología, pero incluyendo dos procesadores y una Field Programmable Gate Array (FPGA), la cual se utiliza para visión por computador. Con todo esto, en Perseverance los algoritmos de navegación autónoma ya no están limitados por el tiempo de computación, siendo AutoNav capaz de tomar mediciones del escenario y decidir qué hacer lo suficientemente rápido como para que el rover se mueva a su velocidad máxima de 4.2 cm/s (Rankin et al., 2023). Además, Perseverance incluye un algoritmo llamado Autonomous Exploration for Gathering Increased Science (AEGIS), capaz de seleccionar y adquirir imágenes, de forma autónoma, de objetivos interesantes desde un punto de vista científico. Utilizar este algoritmo elimina parcialmente la necesidad de teleoperar las tareas científicas desde la Tierra, mejorando notablemente la autonomía del rover. De hecho, Perseverance ha recorrido 13 717 m de forma completamente autónoma, lo cual es el 77.58 % de toda la distancia que este rover ha transitado en Marte. Esto es, en solo dos años, 12 veces la distancia recorrida de forma autónoma por Curiosity, el cual lleva ya 11 años explorando Marte.

Destacar también la autonomía de los rovers Marcianos de otras agencias espaciales. Primero, el rover Zhurong de la Chinese National Space Agency (CNSA), que ha realizado de forma autónoma aproximadamente el 40 % de los 450 m que ha recorrido en Marte (Ding et al., 2022). Segundo, el rover Rosalind Franklin ExoMars (Vago et al., 2017) de la European Space Agency (ESA). Este rover será lanzado en 2028 a pesar de la desafortunada guerra en Ucrania, la cuál interrumpió la colaboración entre la ESA y la Russian Federal Space Agency (Roskosmos). ExoMars incluirá la capacidad de alcanzar un objetivo comandado de forma completamente autónoma, analizando el terreno y planificando una trayectoria viable para alcanzar el destino (Winter et al., 2017).

Es evidente, por tanto, que las capacidades de navegación autónoma de los rovers han evolucionado hasta alcanzar resultados impresionantes con Perseverance. Sin

embargo, además de la navegación, los rovers generalmente hacen ciencia mediante tareas de manipulación, utilizando sus brazos robóticos para colocar instrumentos científicos en localizaciones interesantes del escenario. Ejecutar estas tareas de MM no es aún autónomo, ni siquiera para Perseverance. Un elevado nivel de autonomía para MM podría abrir la puerta a nuevas tareas interesantes para los rovers planetarios, por ejemplo, ensamblaje, construcción, o colaboración entre robots. Recopilemos, por tanto, el pasado, presente, y futuro de la MM en exploración planetaria, y cómo la MM autónoma puede beneficiar los resultados de este tipo de misión planetaria.

Ciencia e ingeniería planetaria a través de la Manipulación Móvil

El primer rover Marciano, Sojourner, ya tenía a bordo una herramienta científica. Se trataba de un Alpha Proton X-Ray Spectrometer (APXS), que fue utilizado para analizar la composición de algunas rocas y del terreno Marciano (Matijevic et al., 1997). No obstante, los primeros rovers que incluían la capacidad de MM fueron los MERs, Spirit y Opportunity. Los MERs tenían múltiples instrumentos científicos instalados en el efector final de sus brazos robóticos, como el Microscopic Imager (MI), un APXS como el de Sojourner, o la Rock Abrasion Tool (RAT), capaz de limpiar rocas o el suelo Marciano para exponer los materiales de las capas inferiores. Gracias a estas herramientas, los MERs obtuvieron evidencia de que el clima Marciano fue más caliente en el pasado, con una atmósfera densa y un ciclo activo del agua, características que hacen posible la existencia de vida microbiana.

Las hazañas científicas de los MERs dispararon el interés de la comunidad espacial en incluir más ciencia en el siguiente programa de exploración Marciana, MSL. Como resultado, Curiosity fue diseñado con capacidades punteras de MM (Anderson et al., 2012). Este rover está equipado con un manipulador de cinco Grados de Libertad o Degrees of Freedom (DoF)s, llamado Sample Acquisition, Processing, and Handling (SA/SPaH). Este subsistema puede obtener muestras del interior de rocas o del suelo mediante un instrumento llamado Collection and Handling for In situ Martian Rock Analysis (CHIMRA), y procesar dichas muestras con varios laboratorios científicos a bordo. Además, la Dust Removal Tool (DRT) puede limpiar las superficies para facilitar el posterior análisis científico con varios instrumentos de contacto: un APXS y el Mars Hand Lens Imager (MAHLI), el cual obtiene imágenes microscópicas para analizar los minerales y texturas de la superficie de las rocas y el suelo Marciano. Gracias a estos instrumentos, los descubrimientos científicos de Curiosity son abrumadores. Curiosity encontró moléculas orgánicas de carbono en algunas muestras de terreno, detectó metano en la atmósfera Marciana, y confirmó la presencia de agua en el pasado en forma de ríos y lagos.

Finalmente, MSL preparó el camino para MSR, con Perseverance. Este rover es parecido a Curiosity, incluyendo el llamado Sampling and Caching Subsystem (SCS) (Moeller et al., 2021). El SCS es un brazo robótico de cinco DoFs, capaz de recoger, sellar y guardar muestras de suelo, rocas y regolitos Marcianos. Este procedimiento

tiene requisitos de control bastante estrictos, para evitar que se contaminen ya que, en algún momento, estas muestras serán enviadas a la Tierra. Al igual que Curiosity, el SCS de Perseverance incluye una herramienta de abrasión de rocas y otra de limpieza de polvo, las cuales preparan las superficies para su posterior análisis con otros instrumentos. Hasta la fecha, Perseverance ha descubierto la presencia de rocas volcánicas en el cráter Jezero, donde amartizó. Esto fue un descubrimiento inesperado, ya que se creía que el origen de las rocas del cráter Jezero era puramente sedimentario, de un antiguo lago. Además, Perseverance ha recogido ya 23 muestras de suelo y rocas, depositando diez de ellas en la superficie Marciana para que una misión futura las recoja y las traiga de vuelta a la Tierra.

Aunque ya descartada, la idea inicial para solucionar el problema de traer de vuelta las muestras era el Sample Retrieval Lander (SRL). SRL era la segunda parte de MSR, en la cual la ESA proporcionaría el Sample Fetch Rover (SFR), un rover ligero y rápido diseñado para recoger los tubos con muestras de terreno mediante MM, y traerlos de vuelta al lugar de amartizaje, donde serían enviados de vuelta a la Tierra (Merlo, Larranaga, and Falkner, 2013). Ahora, NASA está planificando llevar a cabo la recogida de los tubos mediante helicópteros equipados con manipuladores, o con el propio Perseverance. En cualquier caso, la ESA también ha avanzado la tecnología de MM en exploración planetaria, con el rover Rosalind Franklin ExoMars. ExoMars obtendrá muestras de la sub-superficie Marciana, al estar equipado con un taladro de dos DoFs capaz de alcanzar hasta 2 m de profundidad, para así obtener muestras bien conservadas del lecho de rocas. Estas muestras están libres de los daños producidos por la radicación y oxidación presentes en la superficie Marciana, lo cual favorece la supervivencia de restos químicos que prueben la existencia de vida, en caso de que los haya.

Marte ha sido, sin lugar a dudas, el actor principal en la exploración planetaria. No obstante, las agencias espaciales de todo el mundo están lanzándose de vuelta a explorar la Luna, ya que probablemente, a largo plazo, será usada como una estación para mantener y facilitar la expansión humana en el espacio. Por ejemplo, la CNSA ha lanzado dos rovers Lunares, Yutu (Xiao, 2014) y Yutu 2 (Lai et al., 2020), los cuales alunizaron con éxito en 2013 y 2018 respectivamente. Equipados con brazos robóticos, espectrómetros, radares de penetración del suelo, o incluso un Advanced Small Analyzer for Neutrals (ASAN), estos rovers chinos encontraron evidencia de al menos nueve capas de roca diferentes en la Luna, lo que implica una composición geológica de la superficie lunar más compleja de lo esperado. Además, estos rovers proporcionaron imágenes de alta resolución de una eyección Lunar. Por otro lado, la NASA está planeando lanzar el Volatiles Investigating Polar Exploration Rover (VIPER) a la Luna en 2024 (Colaprete et al., 2019), para estimar la cantidad de hielo presente en los polos Lunares, el cual podría ser utilizado por futuros humanos trabajando en una estación Lunar interplanetaria. También muchas otras agencias espaciales, empresas privadas, o incluso universidades, están planeando y llevando a cabo sus propias misiones con rovers para la Luna. Por ejemplo, el rover Lunar

Pragyan de la Indian Space Research Organisation que alunizó en agosto de 2023 (Goswami, 2020); el rover Yaoki, que realizará la primera exploración comercial de la Luna en 2023 (*YAOKI - Japanese Lunar Rover* 2023), o el triste alunizaje fallido del rover Lunar Rashid de los Emiratos Árabes Unidos en abril de 2023 (Almaeeni et al., 2021).

Tal y como Perseverance ha demostrado, la autonomía aumenta enormemente el alcance de las misiones espaciales y dispara el retorno científico. Es de esperar, por tanto, que realizar las tareas de MM también de forma autónoma facilitará la exploración científica de las superficies extraterrestres, al mismo tiempo que abrirá la puerta a una nueva gran variedad de tareas para los rovers. Por ejemplo, permitirá el mantenimiento y operación autónomos de los landers durante largos períodos de tiempo, utilizando las capacidades de MM para abrir/cerrar válvulas o conectar/desconectar cables (Klamt et al., 2020). También permitirá recoger materiales o componentes para transportarlos y ensamblarlos in situ (Schuster et al., 2016). Llegando aún más lejos, un equipo de manipuladores móviles autónomos podría colaborar para construir estructuras en superficies extraterrestres, utilizando recursos in situ (Govindaraj et al., 2019). Este concepto de equipos multirobot explorando de forma colaborativa es, de hecho, uno de los principales objetivos de los últimos proyectos de investigación en ciencia planetaria, los cuales generalmente incluyen un rover grande con capacidades de MM autónoma para recoger muestras geológicas, desplegar instrumentos científicos como una antena de radio (Schuster et al., 2020), proporcionar energía con baterías portátiles, o incluso transportar los rovers más pequeños (Brinkmann et al., 2018). Destacar el uso de equipos heterogéneos de robots con patas para exploración planetaria, incluyendo manipuladores móviles con patas, como fue propuesto en el Space Resources Challenge de la ESA por el equipo del Swiss Federal Institute of Technology (ETH) Zurich (Arm et al., 2023).

Así, es indudable afirmar que enormes avances tecnológicos y científicos se han logrado gracias a la robótica y a la MM en el espacio, y esta tendencia va a seguir creciendo a medida que las agencias espaciales y otras compañías privadas de todo el mundo empiecen a contribuir con sus propios rovers para explorar la gran variedad de superficies extraterrestres de nuestro Sistema Solar. Sin embargo, la autonomía en el espacio es aún un desafío que requiere un gran esfuerzo de investigación, y todavía más si consideramos la MM como la principal forma de realizar tareas científicas e ingenieriles. Desarrollar una tecnología tan avanzada como una MM autónoma es un reto en la Tierra y, por consiguiente, aún más difícil en Marte o la Luna.

Analicemos brevemente cómo funciona un robot autónomo, utilizando el clásico paradigma robótico de Sentir, Pensar, Actuar (Siegel, 2003). Sentir engloba todo subsistema a cargo de percibir el entorno del robot o medir su estado interno. Pensar es donde se localiza la inteligencia artificial, utilizando la información recopilada para decidir qué hacer y generar un plan. Por último, Actuar se encarga de los subsistemas que interaccionan con el mundo real, comandándolos de acuerdo al plan generado. Estos tres componentes son intrínsecamente requeridos por un robot para ser

autónomo. Ahora, centrándonos en la autonomía de robots espaciales, el método más común es una arquitectura de Guiado, Navegación y Control o Guidance, Navigation and Control (GNC) (Gerdes et al., 2020). En este caso, el Guiado (Pensar) es el subsistema que planifica los movimientos del rover para alcanzar el objetivo y realizar la tarea; Navegación (Sentir) localiza el rover y mapea el escenario con los sensores exteroceptivos a bordo del robot, y Control (Actuar) dirige los actuadores de bajo nivel para seguir adecuadamente los comandos, teniendo en cuenta el estado del robot y las perturbaciones externas.

La complejidad de un sistema GNC es ya alta para un rover autónomo, y aún más para un manipulador móvil autónomo. En primer lugar, el rover tiene que alcanzar la zona de interés científico, con una arquitectura de navegación autónoma para la base móvil. Después, el rover debe realizar la tarea científica mediante un movimiento autónomo de MM. Finalmente, el rover tiene que desplazarse al siguiente objetivo científico, generalmente deshaciendo sus propios pasos, navegando de nuevo por los mismos escenarios. Un proceso de MM autónomo dificulta particularmente el subsistema de Guiado, ya que el planificador tiene que tener en cuenta un alto número de DoFs, lo que implica que existen prácticamente infinitas formas de resolver el problema de planificación. Además, para casos de exploración extraterrestres, el planificador tiene que ser lo suficientemente ligero computacionalmente para ser lanzado en procesadores aptos para el espacio, robusto y predecible para asegurar que ningún evento fatal ocurrirá, y suficientemente eficientes para mejorar lo que la teleoperación puede hacer. Por todo esto, es evidente por qué los rovers planetarios no son capaces aún de realizar las tareas de MM de forma autónoma.

Contribuciones

Un rover de MM completamente autónomo requiere primero un sistema de navegación autónoma eficiente para la base móvil, basado en GNC, para llegar al área de interés científico donde realizar la tarea de MM. Después, es necesario un planificador de movimientos para MM, incluyendo ambos: la base móvil y el brazo robótico. Esto, hasta la fecha, no es posible en sistemas espaciales, principalmente por su elevado coste computacional. Por ello, esta tesis doctoral intenta llevar más allá los límites de la exploración espacial autónoma, respondiendo a cómo planificar de forma eficiente los movimientos de un rover planetario para realizar tareas de MM, desde la aproximación global del rover al área de interés científico hasta la ejecución local de la tarea de MM. Esta es una pregunta ambiciosa, la cual requiere responder primero a una serie de desafíos menores:

- Cuando el rover está navegando para alcanzar el área de interés científico, y teniendo en cuenta que el rover probablemente tendrá que recorrer las mismas zonas en el futuro, ¿cómo podemos incrementar el conocimiento del que dispone el rover sobre las características del escenario, para mejorar los planes de movimiento futuros?

- Para que el rover realice la tarea científica, ¿cómo podemos considerar las colisiones de forma robusta y eficiente, durante la planificación de movimientos de MM, para garantizar la seguridad del sistema durante la operación?
- Dadas las restricciones energéticas de los vehículos de exploración planetaria, ¿se podrían aplicar algoritmos basados en control óptimo para planificar los movimientos de MM y optimizar el consumo energético, con un coste computacional admisible?
- Teniendo en cuenta la tendencia de explorar el espacio utilizando equipos de robots, ¿se podrían utilizar algoritmos de control óptimo para MM con el objetivo de incrementar la seguridad de las misiones multirobot en el espacio?

Estas cuatro preguntas se exploran a lo largo de esta tesis doctoral, logrando así una serie de contribuciones en el ámbito de la robótica espacial. Dichas contribuciones son las siguientes:

1. **Un procedimiento de actualización dinámica del mapa de costes, para mejorar la navegación autónoma en misiones de ida y vuelta**.

   La navegación autónoma en rovers planetarios normalmente se divide en dos capas: la capa global y la capa local. Generalmente independientes, la capa global se utiliza para planificar los movimientos a medio y largo plazo, utilizando para ello imágenes orbitales. La capa local, por otro lado, se utiliza para replanificar en caso de que se encuentren nuevos obstáculos con los sensores a bordo del rover. Sin embargo, si el rover tiene que deshacer sus propios pasos para alcanzar el siguiente objetivo, recorriendo de nuevo las mismas zonas, se encontrará otra vez los mismos obstáculos locales, a no ser que la capa global sea actualizada incluyendo la información local.

   Por ello, la primera contribución de esta tesis doctoral es un procedimiento dinámico de actualización del mapa de costes global, el cual incrementa la eficiencia de las trayectorias del rover al moverse en escenarios similares, tras realimentar la información local recogida por el sistema durante la misión. Se utilizan dos fuentes principales de información: primero, la posición de los nuevos obstáculos detectados, que no se consideraron inicialmente en el mapa global; segundo, las características de los distintos tipos de terreno en el escenario, que son continuamente recalculadas para mejorar la estimación inicial. Más detalles en el Capítulo 3.

2. **Un planificador de movimientos robusto y eficiente para MM en rovers de exploración planetaria cuyo espacio de trabajo esté limitado**.

   La presencia de obstáculos en el escenario dificulta la planificación de movimientos seguros para un manipulador móvil. A su vez, las herramientas científicas e ingenieriles a bordo del rover limitan el espacio de trabajo del manipulador, lo que también entorpece al planificador de movimientos. Evitar las

auto-colisiones (colisiones del manipulador con el propio robot) es crítico pero requiere un alto esfuerzo computacional, lo cual ha sido normalmente resuelto mediante una simplificación de la solución, realizando movimientos de MM desacoplados (primero mover la base móvil, después mover el manipulador). Ejecutar el movimiento de forma acoplada es más eficiente en términos de energía y tiempo, pero también más costoso computacionalmente y más arriesgado. Un movimiento acoplado requeriría, por tanto, un método menos exigente computacionalmente para evitar las colisiones, por ejemplo, calculando offline el espacio de trabajo del manipulador para luego tener en cuenta las auto-colisiones de forma directa.

Así, la segunda contribución de esta tesis doctoral es un planificador de movimientos acoplados para manipuladores móviles con el espacio de trabajo restringido, el cual garantiza la seguridad del robot durante la misión al tener en cuenta el espacio de trabajo del manipulador antes de planificar los movimientos. Además, la eficiencia se aumenta notablemente ya que el planificador genera movimientos coordinados para la base del rover y el brazo, evitando tiempos muertos al mover ambos sistemas al mismo tiempo. Más detalles en el Capítulo 4.

3. **Un planificador de movimientos óptimo y computacionalmente ligero para sistemas móviles redundantes, aplicado a MM en exploración planetaria**.

   Los sistemas redundantes como los manipuladores móviles, es decir, plataformas que tienen más DoFs que los necesarios para desplazarse en su espacio Cartesiano, tienen una movilidad considerablemente compleja. Aunque son más versátiles, la gran variedad de posibilidades en sus movimientos implica grandes esfuerzos computacionales para planificarlos. La optimización es uno de los métodos más comunes para resolver el problema de planificación de movimientos para sistemas redundantes, no obstante, su eficiencia computacional se ve notablemente afectada al incluir las dinámicas y las restricciones. Este problema se puede mitigar parcialmente con el llamado "warm start": inicializar la optimización con una solución provisional, rápidamente calculada, la cual indica al optimizador por dónde empezar a buscar la solución óptima.

   Se presenta, así, la tercera contribución de esta tesis doctoral: un planificador de movimientos con varias etapas de inicialización. Este planificador es capaz de encontrar movimientos eficientes en términos energéticos y temporales, teniendo en cuenta la cinemática y dinámica del robot redundante, las restricciones del sistema, y las no linealidades. Además, la secuencia de inicialización reduce el tiempo de convergencia de la optimización, haciendo que el planificador sea apto para vehículos de exploración planetaria. Más detalles en el Capítulo 5.

4. **Un controlador predictivo basado en modelos, o Model Predictive Control (MPC), para realizar MM con un rover grande que apoya a otro más pequeño mientras este accede a un túnel Lunar de lava mediante rappelling**.

   Los túneles de lava Lunares son de gran interés científico para estudiar los orígenes de nuestro Sistema Solar, y también para ser utilizados como refugio en futuras misiones espaciales para incrementar el alcance humano en el espacio. Ya que es costoso y peligroso explorar estos túneles con astronautas, utilizar un equipo de robots sería una alternativa más barata y segura. Para ello, un robot pesado podría actuar como un anclaje móvil, utilizando un brazo robótico para apoyar a un segundo robot, el cual trata de descender hacia el túnel de lava haciendo rappelling, para después explorarlo. Esta misión es muy desafiante para los sistemas implicados, ya que deben soportar grandes esfuerzos durante el rappelling. Para incrementar la robustez de la misión, sería clave controlar esos esfuerzos, en especial los que afectan al brazo robótico del rover de anclaje. Un control reactivo de tan alta frecuencia se puede lograr mediante técnicas modernas de control óptimo, como MPC.

   Aprovechando la tercera contribución de esta tesis doctoral (el planificador óptimo), un controlador óptimo fue desarrollado e integrado en un algoritmo de compensación de esfuerzos. Este algoritmo es capaz de compensar los esfuerzos del brazo robótico durante el rappelling mediante movimientos de MM. Este controlador de tensiones es, así, la cuarta contribución de esta tesis doctoral. Más detalles en el Capítulo 6.

Contexto y motivación

Las misiones de exploración planetaria en curso, y las futuras, están demandando capacidades de navegación autónoma mejoradas, para seguir incrementando el retorno científico, como se ha comentado anteriormente. Por esto, la ESA y la European Commission (EC) están financiando proyectos para potenciar el nivel de autonomía de los rovers planetarios. Esto incluye investigación en planificación de trayectorias y movimientos, MM, y colaboración entre rovers. El Space Robotics Laboratory from the University of Málaga (UMA-SRL), Department of Systems Engineering and Automation (ISA), ha participado de forma activa en varios de estos proyectos, gracias a su amplia experiencia en el ámbito de la navegación autónoma para robots móviles. Este es el contexto donde la investigación de esta tesis doctoral se realizó. En concreto, el autor de esta tesis trabajó en tres de estos proyectos, los cuales impactaron significativamente los resultados de su investigación: primero, una colaboración entre la ESA y el UMA-SRL con tres actividades de investigación consecutivas; segundo, el proyecto H2020 financiado por la EC llamado Autonomous Decision making for very long traverses (ADE); tercero, el último proyecto H2020 en robótica espacial de la EC, llamado Cooperative Robots for Extreme Environments (CoRob-X). Presentamos estos proyectos a continuación.

El UMA-SRL lleva colaborando desde 2016 con el Planetary Robotics Laboratory, Robotics and Automation Section, European Space Agency (ESA-PRL), el cual se encuentra en el European Space Research and Technology Center (ESTEC), mediante tres actividades consecutivas de investigación y desarrollo. La primera actividad, con número de contrato 4000118072/16/NL/LvH/gp, se llamó Autonomous Routing on Extreme Surfaces (ARES), y el objetivo fue desarrollar un planificador de caminos eficiente para rovers reconfigurables, considerando distintos modos de locomoción que pueden ser beneficiosos para recorrer distintos tipos de terreno (Pérez-Del-Pulgar et al., 2017). Las pruebas finales llevados a cabo en el ESA-PRL con el rover Exomars Testing Rover (ExoTeR) fueron muy prometedores, por lo que se continuó investigando hasta publicar varios artículos en revistas internacionales de alto impacto. Primero, el Fast Marching Method (FMM), un método para planificación de caminos que genera trayectorias óptimas y suaves, fue ampliado para mapas multicapa y replanificación dinámica (Sánchez-Ibáñez et al., 2019b). Segundo, una arquitectura GNC eficiente para navegación autónoma fue propuesta en (Gerdes et al., 2020). Finalmente, se presentó en (Geromichalos et al., 2020) el uso de Simultaneous Localization and Mapping (SLAM) y emparejamiento de mapas para localización relativa y absoluta en rovers planetarios. Gracias al planificador desarrollado en ARES, la segunda actividad, llamada Path and Motion planning in Planetary Exploration (PM-PE), se centró en investigar la planificación de movimientos para un rover equipado con un brazo robótico, el cual tiene que recoger objetos (Sánchez-Ibáñez et al., 2019a). Los tests finales de PM-PE permitieron a ExoTeR recoger tubos de muestra con un plan de movimientos eficiente y robusto. Sin embargo, fue necesario realizar un movimiento de barrido con el efector final para tener en cuenta los errores de localización en la posición del rover y de la muestra. Para llevar a cabo una misión completa similar a SFR, el rover debía ser capaz de detectar e identificar la muestra antes de recogerla, lo cual fue el principal inconveniente del proyecto PM-PE. Así, la tercera actividad ESA-UMA, llamada Path and Motion planning for a Sample Fetching Rover (PM-SFR), se centró en integrar un sistema inteligente de visión por computador en la arquitectura de navegación autónoma del rover (Castilla-Arquillo et al., 2022), para detectar los tubos durante la trayectoria y ejecutar la misión de recogida de muestras de forma completamente autónoma (Mantoani et al., 2022). Los resultados de esta tercera actividad fueron impresionantes, con ExoTeR navegando totalmente autónomo para alcanzar, detectar y recoger un tubo de muestra.

Fue en el contexto de estas actividades de colaboración entre la ESA y el UMA-SRL que el autor de esta tesis tuvo el privilegio de realizar dos estancias de investigación en ESTEC, Noordwijk, Países Bajos, bajo la supervisión del Dr. Martin Azkarate, con una duración total de seis meses. Durante estas estancias, el autor investigó y desarrolló parte de los algoritmos que constituyen las contribuciones de la tesis, y también realizó algunos de los tests de laboratorio y de campo que se

muestran en el Capítulo 6. En concreto, los algoritmos y experimentos de la Contribución 1 fueron completamente realizados en ESTEC durante la primera estancia de investigación a finales de 2019; el algoritmo propuesto en la Contribución 2 se basa en el planificador de movimientos desarrollado para la segunda actividad de colaboración, PM-PE, y las pruebas de laboratorio de la Contribución 3 fueron completamente llevados a cabo en el ESA-PRL durante la segunda estancia.

La experiencia adquirida y los algoritmos desarrollados durante la colaboración ESA-UMA benefició al UMA-SRL para participar en el proyecto europeo *Autonomous DEcision making in very long traverses (ADE)* 2021. Este fue un proyecto H2020 financiado por la EC, perteneciente al Strategic Research Cluster (SRC) en Space Robotics Technologies. El objetivo de ADE era integrar gran parte de las tecnologías desarrolladas en antiguos proyectos de H2020 para crear una arquitectura de navegación completamente autónoma, permitiendo al rover navegar largas distancias por sí mismo, entre 1 y 3 km. Coordinado por GMV Aerospace and Defence S.A., el consorcio estaba formado por 14 entidades, incluyendo el Robotics Innovation Center (RIC) del German Research Center for Artificial Intelligence (DFKI), el cual estaba a cargo del rover SherpaTT y del simulador.

UMA-SRL era responsable del componente de MM, el cual generaba un plan de movimientos para el sistema rover-manipulador con el objetivo de alcanzar un punto en el escenario y realizar una tarea científica con el brazo robótico (Sánchez-Ibáñez et al., 2020). En este caso, era especialmente difícil generar movimientos seguros ya que el espacio de trabajo del manipulador estaba muy limitado por la presencia de varios instrumentos de localización, detección de obstáculos y generación de Digital Elevation Map (DEM)s. Consecuentemente, en el contexto de ADE, el autor de esta tesis investigó y desarrolló un planificador de movimientos robusto, el cual prioriza la seguridad del manipulador utilizando métodos para evitar las auto-colisiones. Este planificador es la segunda contribución de esta tesis. Con esto, las simulaciones del componente de MM para ADE, así como los experimentos de campo que se realizaron en 2021 en Bremen, Alemania, dieron pie a los resultados que validan la Contribución 2.

En tercer lugar, el UMA-SRL participó en el proyecto *CoRob-X - Cooperative Robots for Extreme Environments* 2023, el cual comenzó en Marzo de 2021, siendo el proyecto H2020 más desafiante de todos. Al ser el último de los proyectos H2020 en robótica espacial, las tecnologías desarrolladas en todos los proyectos anteriores fueron integradas en varias Robotic Explorer Units (REU)s, que tenían que colaborar para explorar superficies planetarias, con el foco en los túneles de lava de la Luna. CoRob-X fue coordinado por DFKI, y el UMA-SRL estaba a cargo de la navegación autónoma de tres de las REUs: los rovers SherpaTT y CoyoteIII del DFKI, y el rover LUVMI-X de Space Application Services (SAS). Además, en CoRob-X se planificó una tarea de MM muy arriesgada: CoyoteIII accediendo mediante rappelling a un túnel de lava, con SherpaTT comportándose como el anclaje móvil que soporta el peso de CoyoteIII mientras desciende.

Esta operación de rappelling requería un planificador y controlador de movimientos rápido y eficiente, para que el brazo robótico de SherpaTT se colocara en las posiciones óptimas que permitieran a CoyoteIII descender de forma segura. Esto significaba minimizar los esfuerzos que el brazo robótico soportaba, intentando redistribuirlos hacia las articulaciones más fuertes del manipulador, además de asegurar que el brazo tenía suficiente movilidad para reaccionar. El desarrollo de un algoritmo tan complejo motivó la cuarta contribución y la estancia del autor de esta tesis en el DFKI RIC en Bremen, Alemania, durante todo el año 2022. Esta estancia de investigación, por otro lado, permitió que se realizara un acuerdo de cotutela para los estudios de doctorado, con la supervisión conjunta del Dr. Carlos Jesús Pérez del Pulgar del UMA-SRL y del Dr. Frank Kirchner de la Universität Bremen. Finalmente, en 2023 se llevaron a cabo los experimentos de campo de CoRob-X en Lanzarote, Islas Canarias, España, donde se realizaron varios rappelling con éxito a un túnel de lava, análogo a los existentes en la superficie Lunar.

Publicaciones

Las contribuciones de esta tesis doctoral fueron publicadas en conferencias y revistas relevantes en el ámbito de la robótica espacial. En concreto, cuatro publicaciones principales avalan dichas contribuciones, que son las siguientes:

- **Improving Autonomous Rover Guidance in Round-Trip Missions Using a Dynamic Cost Map**. Gonzalo J. Paz-Delgado, Martin Azkarate, J. Ricardo Sánchez-Ibáñez, Carlos J. Pérez-del-Pulgar, Levin Gerdes, y Alfonso García-Cerezo. *International Conference on Intelligent Robots and Systems (IROS)* Las Vegas, United States of America (USA), 25-29 10 2020 (Paz-Delgado et al., 2020).

  La Contribución 1 de esta investigación doctoral se resumen en este artículo de conferencia, relacionado con un procedimiento para mejorar los movimientos planificados de forma dinámica en exploración planetaria. Esto se realiza a través de una actualización del mapa de costes del planificador de trayectorias, de modo que el rover sea más consciente de la información que ha recogido sobre el escenario durante el recorrido, y la pueda usar para generar mejores caminos sucesivamente durante la misión. El artículo también muestra una campaña de tests en simulación y experimentos de campo con el rover Heavy Duty Planetary Rover (HDPR) del ESA-PRL en Decos, un terreno análogo a la superficie Marciana localizado cerca de ESTEC en Noordwijk, Países Bajos.

- **Combined path and motion planning for workspace restricted mobile manipulators in planetary exploration**. Paz-Delgado, J. Ricardo Sánchez-Ibáñez, Raúl Domínguez, Carlos J. Pérez-del-Pulgar, Frank Kirchner, y Alfonso García-Cerezo. *IEEE Access* (Paz-Delgado et al., 2023b).

Este artículo de revista está directamente relacionado con el proyecto ADE y la Contribución 2. El artículo presenta un planificador de movimientos y trayectorias acoplados para MM en exploración planetaria, considerando el caso de que el brazo robótico tenga el espacio de trabajo muy restringido. Para asegurar la seguridad durante la misión, el espacio de trabajo del manipulador se calcula offline y se utiliza durante la etapa de planificación de movimientos, de modo que se eviten las auto-colisiones. Además, la eficiencia del plan de movimientos se incrementa, al coordinar los movimientos del rover y del manipulador. El artículo concluye con una serie de experimentos en simulación y de campo con el rover SherpaTT del DFKI RIC, llevados a cabo durante el proyecto ADE.

- **Multi-stage warm started optimal motion planning for over-actuated mobile platforms**. Gonzalo J. Paz-Delgado, Carlos J. Pérez-del-Pulgar, Martin Azkarate, Frank Kirchner, y Alfonso García-Cerezo. *Springer Journal on Intelligent Service Robotics (JIST)* (Paz-Delgado et al., 2023a).

  La investigación y resultados de la Contribución 3, el planificador de movimientos óptimo para plataformas móviles redundantes, se presentan en este artículo de revista. En concreto, se propone un procedimiento genérico para modelar sistemas móviles redundantes, como manipuladores móviles, y una novedosa secuencia de inicialización para planificadores de movimiento basados en control óptimo. Finalmente, concluye mostrando los resultados de una serie de tests de laboratorio con ExoTeR en el ESA-PRL, los cuales validaron el algoritmo presentado. Este artículo fue portada de la revista *Springer JIST*, Volumen 16, Número 3, Julio 2023[1].

- **Cooperative robotic exploration of a planetary skylight surface and lava tube**. Raúl Domínguez, Carlos J. Pérez-del-Pulgar, Iulia Dragomir, Valérie Ciarletti, Gonzalo J. Paz-Delgado, Anne-Claire Berthet, Fabio Polisano, Leon Cedric Danter, Jonathan Babel, y Frank Kirchner. En preparación.

  En este artículo de revista se resumen los resultados de los experimentos llevados a cabo durante 2023 en las pruebas de CoRob-X en Lanzarote, incluyendo el rappelling con un equipo de robots, el cual está directamente relacionado con la Contribución 4 de esta tesis. Por ello, se detalla la arquitectura de rappelling autónomo, incluyendo el MPC para apoyar el rappelling mediante MM, el equipo multirobot, y los experimentos de rappelling en sí.

Por otro lado, durante sus estudios de doctorado el autor colaboró en múltiples actividades de investigación y proyectos relacionados con la robótica espacial, con varias publicaciones más. Las principales se enumeran a continuación:

[1] `https://link.springer.com/journal/11370/volumes-and-issues/16-3`

- **Hardware-Accelerated Mars Sample Localization Via Deep Transfer Learning from Photorealistic Simulations**. Raúl Castilla-Arquillo, Carlos J. Pérez-del-Pulgar, Gonzalo J. Paz-Delgado, y Levin Gerdes. *IEEE Robotics and Automation Letters*, pp 99:1-8, 2022 (Castilla-Arquillo et al., 2022).

  Este artículo de revista propone un método de inteligencia artificial para, de forma autónoma, detectar y estimar la posición de tubos con muestras de terreno Marciano como los de MSR, utilizando una Red Neuronal Profunda (Deep Neural Network) y aprendizaje transferido (Transfer Learning). El conjunto de datos utilizados se obtuvo mediante un simulador fotorealista de Marte en Unreal Engine 4. Este algoritmo de detección de muestras fue integrado en la arquitectura GNC de ExoTeR, y utilizado durante los tests de laboratorio en la Contribución 3.

- **Choosing the Best Locomotion Mode in Reconfigurable Rovers**. Carlos J. Pérez-del-Pulgar, Pablo Romeo-Manrique, Gonzalo J. Paz-Delgado, J. Ricardo Sánchez-Ibáñez, y Martin Azkarate. *Electronics*, 8(7):818, 7 2019 (Pérez-Del-Pulgar et al., 2019).

  Algunos rovers como ExoTeR tienen diferentes modos de locomoción gracias a la reconfigurabilidad de sus patas y ruedas. En este artículo se analizan, así, dos modos de locomoción, mediante simulaciones con Vortex Studio. Estos modos de locomoción son Normal Driving, el modo estándar de conducción con ruedas, y Wheel-walking, donde las patas son usadas para realizar pequeños pasos en una secuencia. Dependiendo de las características del terreno, el mejor modo de locomoción a utilizar se puede estimar con los resultados de este artículo. Además, se presenta un nuevo método para estimar el derrape mientras se usa Wheel-walking.

- **Experimental analysis of slip ratio using the wheel walking locomotion mode in reconfigurable rovers**. Salvador Domínguez-Durante, Carlos J. Pérez-del-Pulgar, Gonzalo J. Paz Delgado, y Martin Azkarate. *30th Mediterranean Conference on Control and Automation (MED)*, 2022 (Dominguez-Durante et al., 2022).

  Continuando con la investigación previa sobre modos de locomoción en rovers reconfigurables, este artículo de conferencia presenta la validación experimental del método de estimación del derrape antes mencionado, el cual permite calcular el derrape mientras se utiliza Wheel-walking. Con este objetivo, se propone una metodología para inducir un cierto derrape a un robot, la cual consiste en un terreno llano y un mecanismo de sujeción móvil.

- **Coupled Path and Motion Planning for a Rover-Manipulator System**. J. Ricardo Sánchez-Ibáñez, Gonzalo J. Paz-Delgado, Pablo Romeo-Manrique, Carlos J. Pérez-del-Pulgar, y Martin Azkarate. *15th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2019 (Sánchez-Ibáñez et al., 2019a).

En este artículo de conferencia los autores presentan el algoritmo en el que se basó el planificador de movimientos de la Contribución 2. Este primer planificador de movimientos genera un túnel alrededor de la trayectoria de la base móvil, en el cual se utiliza FMM en 3D para trazar la trayectoria del efector final del manipulador. Después, el modelo cinemático inverso del brazo robótico proporciona las configuraciones del brazo para seguir dicha trayectoria. El planificador fue finalmente validado mediante simulaciones con ExoTeR en Vortex Studio.

- **Samples detection and retrieval for a Sample Fetching Rover**. Laura M. Mantoani, Raúl Castilla-Arquillo, Gonzalo J. Paz-Delgado, Carlos J. Pérez-del-Pulgar, y Martin Azkarate. *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2022 (Mantoani et al., 2022).

  La arquitectura de recogida autónoma de muestras que se integró en ExoTeR se presenta en este artículo de conferencia, incluyendo los algoritmos de Guiado (planificación y seguimiento de caminos) y el detector de muestras. Se presentan a su vez dos experimentos, primero, en el terreno experimental de búsqueda y rescate de la UMA, y, segundo, en el ESA-PRL en ESTEC. Esta arquitectura de navegación autónoma se compara, más tarde en esta tesis, con los algoritmos propuestos en la Contribución 3.

- **CoRob-X: A cooperative robot team for the exploration of Lunar skylights**. Alexander Dettmann, Thomas Vögele, Jorge Ocón, Iulia Dragomir, Shashank Govindaraj, Matteo De Benedetti, Valerie Ciarletti, Rafik Hassen-Khodja, Thierry Germa, Raphael Viards, Gonzalo J. Paz-Delgado, y Laura M. Mantoani. *16th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2022 (Dettmann et al., 2022).

  Este artículo de conferencia proporciona una visión general del proyecto CoRob-X, el cual está relacionado con la Contribución 4. El artículo introduce los distintos algoritmos utilizados en las REUs y las distintas fases de la misión de exploración del túnel de lava, la cual fue más adelante realizada mediante los experimentos de campo en Lanzarote.

- **Virtual reality lab for rover navigation using Mars datasets**. Raúl Castilla-Arquillo, Gonzalo J. Paz-Delgado, Matteo Madi, y Carlos J. Pérez-del-Pulgar. *17th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2023

  En este artículo de conferencia se presenta un laboratorio de realidad virtual de alta fidelidad para simular rovers planetarios. Desarrollado por el UMA-SRL, el laboratorio de realidad virtual utiliza información orbital de Marte y de los rovers para generar un entorno Marciano con la ayuda de algoritmos de inteligencia artificial. Por un lado, métodos de superresolución permiten

incrementar la calidad de las imágenes orbitales para así generar mejores texturas en el escenario. Por otro lado, se utilizan redes neuronales para crear elementos del terreno en 3D, como rocas, a partir de las imágenes locales del rover.

- **CoRob-X: Demonstration of a cooperative robot team in extensive field tests**. Thomas Vögele, Jorge Ocón, Thierry Germa, Shashank Govindaraj, Carlos J. Pérez-del-Pulgar, Fredrik Bakkevig Haugli, Raúl Domínguez, Leon Cedir Danter, Jonathan Babel, Iulia Dragomir, Anne-Claire Berthet, Fabio Polisano, Gonzalo J. Paz-Delgado, Laura M. Mantoani, Eric Törn, y Valerie Ciarletti. *17th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2023

  Este artículo de conferencia muestra los objetivos y resultados de los experimentos de campo de CoRob-X en Lanzarote, Islas Canarias, España. El artículo resume los resultados de la exploración de un túnel de lava análogo a la Luna, analizando las distintas fases para llevarla cabo, incluyendo el rappelling en el cual se utilizaron los algoritmos de la Contribución 4 de esta tesis doctoral.

Estructura de la memoria

Esta tesis doctoral se divide en siete capítulos, primero, introduciendo el tema y analizando el estado del arte, segundo, describiendo las cuatro contribuciones principales, tercero, presentando los experimentos realizados y los resultados, y cuarto, extrayendo las conclusiones del trabajo realizado y describiendo algunas líneas de trabajo futuro. En concreto, los capítulos que conforman la tesis son los siguientes:

1. **Introducción**. El primer capítulo da contexto a la tesis doctoral, explicando donde encaja la investigación realizada, por qué es necesario mejorar la tecnología en este ámbito, y qué problemas soluciona la tesis. Además, este capítulo pone al lector en contexto sobre en qué circunstancias se originó este trabajo, y resume las contribuciones y publicaciones llevadas a cabo.

2. **Estado del arte en manipulación móvil autónoma**. Este capítulo es un análisis completo de la situación actual de la tecnología en relación con los algoritmos y arquitecturas necesarios para realizar MM de forma autónoma en superficies extraterrestres, principalmente hablando de percepción, planificación de caminos, planificación de movimientos, y control de movimientos.

3. **Actualización dinámica del mapa de costes para mejorar la planificación global de caminos**. La teoría detrás de la Contribución 1 se presenta en este capítulo, describiendo cómo funciona la metodología propuesta para actualizar de forma dinámica el mapa de costes del planificador de trayectorias una vez que el rover está recorriendo el escenario. Primero, el planificador utilizado se introduce para destacar sus peculiaridades; segundo, se exponen

los procedimientos para actualizar los costes, tanto para los nuevos obstáculos detectados como para la estimación de características del terreno.

4. **Planificación robusta de movimientos para manipulación móvil acoplada**. La Contribución 2 se explica al completo en este capítulo, es decir, el planificador robusto y eficiente de movimientos para MM en exploración planetaria, enfocado en robots móviles en los que el espacio de trabajo del manipulador está muy restringido. Se trata de un planificador combinado de trayectorias y movimientos, por tanto, se detallan primero las mejoras del planificador de trayectorias para tener en cuenta la tarea de MM, para después describir el planificador de movimientos en sí y el controlador acoplado necesario para coordinar los movimientos de la base móvil y el brazo robótico.

5. **Planificación óptima de movimientos para robots móviles redundantes**. El cuarto capítulo presenta la Contribución 3, el planificador de movimientos óptimo y computacionalmente ligero para robots redundantes. Para ello, se introduce primero un método genérico para modelar robots móviles redundantes compuestos de varias cadenas cinemáticas. Después, se analiza el planificador óptimo con varias etapas de inicialización, con una descripción detallada de cada una de esas etapas.

6. **Resultados**. Este capítulo incluye todas las pruebas numéricas, de simulación, de laboratorio y de campo llevadas a cabo durante estos estudios doctorales, aquellas que validan las soluciones propuestas y demuestran sus ventajas y desventajas desde los puntos de vista cualitativo y cuantitativo. En concreto, se incluyen cuatro campañas experimentales, una para cada contribución. Primero, los experimentos de ida y vuelta con la actualización dinámica del mapa de costes, realizados con HDPR de la ESA. Segundo, los experimentos de MM completamente segura con SherpaTT en ADE. Tercero, los experimentos de recogida óptima de muestras con ExoTeR en el ESA-PRL. Cuarto, el rappelling coordinado para acceder túneles de lava Lunares, con SherpaTT y CoyoteIII en CoRob-X. Este capítulo también analiza los controladores utilizados para seguir los planes de movimiento óptimo con las distintas plataformas, es decir, el algoritmo de replanificación basado en error y el MPC. Este último constituye la Contribución 4.

7. **Conclusiones y trabajo futuro**. La tesis finaliza con este capítulo, el cual resume todas las soluciones propuestas, analiza los resultados, y propone algunas líneas de trabajo futuro, es decir, otros aspectos relacionados con la MM autónoma que necesitan más investigación, problemas que aún no han sido resueltos, o soluciones que pueden ser mejoradas.

Conclusiones de la tesis doctoral

Tal y como se ha explicado a lo largo de esta tesis doctoral, la autonomía en la exploración espacial es una gran ventaja en términos de eficiencia energética y temporal, teniendo en cuenta que se eliminan casi por completo los retrasos que genera la teleoperación directa desde la Tierra. Ya que una gran cantidad de las tareas científicas que un rover realiza incluyen MM, realizar los movimientos de MM de forma autónoma implica incrementar el número y la complejidad de las tareas que el rover puede llevar a cabo a lo largo de su misión. Esta tesis doctoral, por tanto, explora diferentes formas de alcanzar la realización autónoma de tareas de MM en robots espaciales. Para ello, se responde a cómo planificar los movimientos de un rover planetario de forma eficiente, para llevar a cabo tareas de MM.

Para dar respuesta a esta pregunta, la tecnología terrestre se ha analizado, extraído y particularizado al caso de la exploración planetaria, en varios pasos que constituyen varias contribuciones. Primero, incrementando el conocimiento que el rover posee respecto al escenario, mejorando el planificador global de trayectorias cuando el rover trata de alcanzar la zona de interés científico. Segundo, incluyendo la evasión de obstáculos de una forma eficiente a la par que robusta en el planificador de movimientos, para así incrementar la seguridad cuando se realizan tareas científicas de MM. Tercero, optimizando la energía que se utiliza en los movimientos sin necesidad de un gran esfuerzo computacional, mediante un planificador óptimo que maximiza la eficiencia de los movimientos de MM. Cuarto, controlando de forma óptima los movimientos en operaciones de MM multirobot, mejorando la seguridad de los sistemas involucrados. A continuación, se resumen y extraen algunas conclusiones respecto a estas cuatro contribuciones.

Contribución 1: Actualización dinámica del mapa de costes para mejorar la planificación global de caminos.

Antes de alcanzar un lugar de interés científico y realizar una tarea de MM, un rover planetario generalmente recorre largas distancias. Por ello, alcanzar un nivel de autonomía y eficiencia en MM primero requiere conseguir una navegación global autónoma y eficiente. Teniendo en cuenta que estos recorridos generalmente se realizan en áreas con grandes similitudes, o incluso en las mismas áreas mediante trayectorias de ida y vuelta, la primera pregunta de esta tesis doctoral es: ¿cómo incrementar el conocimiento del rover sobre las características del escenario para mejorar los futuros planes de movimiento?

La Contribución 1 es, por tanto, un procedimiento de actualización dinámica del mapa de costes para mejorar continuamente la calidad de planificación de trayectorias globales durante el recorrido. Integrada dentro del planificador de trayectorias Dynamic Multilayered path planner (DyMu), esta metodología de actualización mediante realimentación mejora el comportamiento del rover en dos sentidos principalmente. Por un lado, los obstáculos detectados localmente se incluyen dinámicamente en el mapa de costes. Así, el rover no volverá a toparse con ellos, al ser evitados directamente en la trayectoria global. Por otro lado, la estimación de las propiedades del terreno se va afinando continuamente durante el recorrido, modificando el coste de las distintas áreas del mapa de coste. Así, el planificador tenderá a generar las trayectorias sobre aquellas áreas del escenario que son más fáciles de recorrer.

Esta metodología de actualización dinámica del mapa de costes fue validada mediante una serie de experimentos tanto en simulación como en el campo, con el rover HDPR del ESA-PRL. La conclusión de estos experimentos es que la realimentación dinámica al mapa de costes mejora el planificador global de trayectorias, ya que el rover es más consciente de las características del escenario. Es especialmente útil para misiones de ida y vuelta, es decir, aquellas en las que el rover va a un lugar de interés científico, realiza una tarea de MM, y vuelve al punto inicial. Durante el camino de retorno el rover recorre las mismas áreas, y toda la información recogida en el camino de ida se utiliza para mejorar la calidad de las trayectorias de vuelta. En cualquier caso, esta metodología también mejora el planificador cuando se recorren terrenos desconocidos, ya que la información recogida in situ puede extrapolarse a zonas similares del escenario, utilizando para ello un mapa de segmentación global.

Contribución 2: Planificador de movimientos robusto y eficiente para MM.

Una vez que el rover ha alcanzado la zona de interés científico, tiene que realizar de forma autónoma un movimiento combinado de MM para llevar a cabo la tarea científica objetivo. Sin embargo, planificar ese movimiento es computacionalmente costoso debido a la dificultad de tener en cuenta las colisiones, si ambos sistemas (base móvil y brazo robótico) se mueven al mismo tiempo. Esta es la segunda pregunta de esta tesis doctoral: ¿cómo tener en cuenta las colisiones de forma robusta y eficiente durante la planificación de movimientos para MM?

La Contribución 2 es, así, un planificador robusto y eficiente de movimientos para MM en exploración planetaria. Este algoritmo planifica primero un camino para la base móvil que evita los obstáculos del escenario hasta alcanzar el objetivo. Después, planifica los movimientos del manipulador teniendo en cuenta la trayectoria de la base móvil. El plan de movimientos generado siempre deja el objetivo dentro del espacio de trabajo del manipulador, ya que el planificador incluye una metodología específica de planificación y control en el último tramo de la trayectoria de la base móvil. Además, el método propuesto es bastante ligero computacionalmente incluso en el caso de que el espacio de trabajo del manipulador esté muy

limitado, ya que las auto-colisiones se tienen en cuenta utilizando un volumen de alcance del manipulador calculado offline.

Este planificador robusto de movimientos fue validado mediante una segunda campaña de experimentos en simulación y de campo, con el rover SherpaTT del DFKI RIC. Estos experimentos demostraron que el planificador propuesto es eficiente gracias a los movimientos coordinados de la base móvil y del manipulador. Un despliegue progresivo del brazo es, entre todas las configuraciones posibles del planificador, la opción más eficiente, al distribuir los movimientos del brazo a lo largo de toda la trayectoria del rover. Además, el planificador de movimientos garantiza de forma intrínseca que el plan es seguro, evitando completamente los obstáculos y las auto-colisiones. Esta seguridad se alcanza de una manera computacionalmente eficiente, gracias al volumen de alcance del brazo robótico, calculado previamente, offline.

Contribución 3: Planificador óptimo de movimientos, y computacionalmente ligero, para sistemas móviles redundantes.

Aunque el planificador de movimientos para MM de la Contribución 2 resuelve el problema de forma robusta y eficiente, sigue siendo un método subóptimo. Como tal, no puede optimizar la energía utilizada durante la operación, por lo tanto, la eficiencia de los movimientos de MM no se maximiza. Además, aunque coordina la base móvil y el manipulador, el movimiento es generado por separado. Este método no permite, por ello, utilizar los grados de libertad de uno de los sistemas (base móvil o brazo robótico) para compensar las limitaciones del otro. Estos problemas se pueden solucionar directamente con un planificador óptimo de movimientos basado en control óptimo para el robot completo, lo que genera la tercera pregunta del doctorado: ¿pueden los algoritmos de control óptimo ser aplicados a la planificación de movimientos de MM en exploración planetaria, con un coste computacional admisible?

La Contribución 3 es, así, un planificador de movimientos basado en control óptimo para robots móviles redundantes. Llamado Multi-staged Warm started optimization Motion Planner (MWMP), este planificador tiene en cuenta las dinámicas del sistema completo, y sus restricciones, al mismo tiempo que mantiene un coste computacional aceptable. Para ello se incluyen varias etapas de inicialización para acelerar la velocidad de convergencia de la optimización. En concreto, tres etapas de inicialización se incluyen en MWMP: un planificador de caminos (aprovechando las mejoras de la primera contribución de esta tesis doctoral), un planificador de movimientos sin restricciones basado en Sequential Linear Quadratic Regulator (SLQ), y un planificador de movimientos con restricciones también mediante SLQ. Para utilizar este planificador se necesita un modelo en espacio de estados del sistema, por tanto, también se presenta un procedimiento genérico para modelar robots móviles redundantes que incluyen múltiples cadenas cinemáticas, por ejemplo, manipuladores móviles.

Antes de las pruebas experimentales, el método propuesto fue evaluado, comparando MWMP con cualquier otra combinación de las etapas de inicialización. Esta comparativa demostró que la secuencia de inicialización propuesta mejora el comportamiento y reduce el coste computacional, con la convergencia más rápida y el porcentaje de éxito más elevado. Después, la viabilidad de MWMP fue confirmada con una serie de pruebas de laboratorio con ExoTeR en el ESA-PRL. Para ello, un seguidor de movimientos fue desarrollado, con el objetivo de que el rover fuera capaz de realizar los movimientos planificados con MWMP. Este seguidor comprueba si existen errores en el estado actual del robot respecto a lo planificado, y solicita un nuevo plan de movimientos en caso de que las desviaciones sean significativas. Los tests de laboratorio confirmaron que MWMP genera movimientos precisos y que cumplen las limitaciones del sistema, siendo capaz de resolver las restricciones no-holónomas de la base móvil, de filtrar los errores de localización del rover y del objetivo, y de llevar a cabo la tarea de MM de una forma eficiente y optimizada.

Contribución 4: Control óptimo de la tensión en rappelling multirobot.

No todas las aplicaciones de MM en el espacio necesitan que el rover alcance primero un punto de interés científico en el escenario para realizar la tarea de manipulación. En algunos casos se necesita una cierta capacidad de reacción. Un ejemplo claro es el caso de una operación de rappelling multirobot en un túnel de lava. Brevemente, un rover grande que se comporta como un anclaje móvil usa su manipulador para permitir que un segundo rover descienda hacia el túnel de lava haciendo rappelling. Por lo tanto, el manipulador soporta el peso del rover pequeño mientras este desciende. El peso del rover genera unos esfuerzos grandes en la cuerda que conecta ambos sistemas, la cual a su vez transmite esos esfuerzos a las articulaciones del brazo robótico. Estos esfuerzos, en concreto los pares, pueden dañar las articulaciones del brazo si el sistema no reacciona a tiempo. En este contexto surge la cuarta pregunta que esta tesis doctoral trata de responder: ¿puede un control óptimo de MM incrementar la seguridad en misiones multirobot en el espacio?

La contribución 4 es, por tanto, un controlador óptimo para MM. Aplicado a una operación de rappelling multirobot, este controlador puede compensar la tensión que la cuerda aplica al brazo robótico del rover grande (el que actúa como un anclaje móvil), mientras el rover pequeño realiza el rappelling y desciende hacia el túnel de lava. Este controlador óptimo utiliza el planificador óptimo MWMP de la Contribución 3, ya que es suficientemente ligero computacionalmente como para ser integrado en un bucle MPC de alta frecuencia. Para mejorar la seguridad de la operación, este algoritmo de control mide continuamente los esfuerzos (fuerzas y pares) que actúan en el efector final del manipulador, y trata de reducirlos. Para ello, intenta alinear el último eslabón del manipulador y la cuerda, eliminando los pares que genera la tensión de la cuerda en la muñeca del manipulador. Los pares se redistribuyen, así, a las articulaciones del hombro y del codo del brazo robótico,

que son más fuertes, evitando que afecten a las articulaciones de la muñeca, que son más débiles.

La metodología MPC de MM para el control de la tensión fue validada mediante simulaciones y experimentos de campo con los rovers SherpaTT y CoyoteIII del DFKI RIC. Los tests en simulación demostraron que el controlador óptimo reduce notablemente los pares que el manipulador soporta durante el rappelling. Por el contrario, las simulaciones también mostraron que mantener el brazo quieto podría llevar a un fallo total, ya que las articulaciones podrían tener que soportar pares superiores a sus límites. La utilidad del controlador propuesto fue finalmente confirmada en los experimentos de campo, ya que múltiples operaciones de rappelling fueron ejecutadas con éxito, sin exceder nunca los límites de par en ninguna de las articulaciones del brazo robótico de SherpaTT.

"Per Aspera, Ad Astra"