

Optimization of a Mobile Manipulator for Consecutive Pick and Place Tasks

Jesse Beuman

Thesis Final Report
DC 2020.100

Supervisor: dr.ir. A.A.J. Lefeber
Supervisor: prof. dr. H. Nijmeijer
Coach: ir. Léon Bemelmans
Coach: ir. Matthieu Bemelmans

Eindhoven University of Technology
Department of Mechanical engineering
Dynamics and Control
Eindhoven
September 2020

Summary

Industrial manipulators have been a highly researched topic for many years. Manipulator trajectories can be optimized to reduce the duration of task motions which is beneficial for the throughput. By increasing the throughput more products can be handled and eventually the profit is increased. However, these manipulator trajectories are optimized for fixed tasks at fixed locations for a large number of repetitions. This is because automation of a task is a manual process. Optimization of variable processes is thus not viable due to the required human interaction. In the flexible manufacturing field lab at the Brainport Industries Campus (BIC) research is committed to automation of variable processes. The goal is to be able to level the cost of mass produced products and customizable products with batch sizes of 1. A sub-project is dedicated to the development of a mobile manipulator. A mobile manipulator is able to perform tasks at multiple locations which means that it is able to pick and place objects. Moreover, it is able to machine a product at various workstations which would otherwise require multiple manipulators. One of the challenges of a mobile manipulator is the increased freedom of motion. Task motions cannot be uniquely related to manipulator motions which increases complexity but also leaves room for optimization. Based on the kinematic and dynamic model of the mobile manipulator a trajectory generation method is formulated. Trajectories are optimized based on energy consumption and evaluated for feasibility in terms of collisions, task motions and joint limits. Based on a comparison with a benchmark situation a significant reduction in energy consumption is observed. By only defining task positions it is possible for the trajectory generation method to generate a trajectory and thus the mobile manipulator is able to operate in a flexible environment without needing human interaction.

Acknowledgments

I would like to take this opportunity to thank the people who have helped me during this graduation project.

I would like to thank Henk Nijmeijer for the guidance, advice and support during our monthly meetings.

I would like to thank Erjen Lefeber for the flexibility while searching for a graduation project which has led to this thesis. Most importantly, thank you for the countless hours of support, guidance and motivation during our weekly meetings.

I would like to thank Léon Bemelmans and Matthieu Bemelmans for welcoming me at VinciTech and providing me with the possibility to discuss the project in a practical manner.

Last but not least, I would like to thank my parents and my girlfriend for their support, both related and unrelated to the graduation project.

Nomenclature

Abbreviations

m	Manipulator frame
MRP	Mobile Robot Platform
DH	Denavit Hartenberg
W	World fixed frame
CF	Cost Function

Coordinates

q	Generalized coordinate of the kinematic system
r	Position
θ	Rotation
${}^c x_a^b$	Coordinate of a expressed in frame b with respect to position c
v, \dot{r}	Linear velocity
$\omega, \dot{\theta}$	Rotational velocity
ξ	End effector position and orientation as function of the joint coordinates
ψ	Euler angles of end effector orientation
φ	Entries of the known rotation matrix of the end effector
$\dot{q}_{n,i}$	Null space velocity
N_i	Null space variation
$r_{c,i}$	Center positions of the manipulator links
\hat{q}	Coordinate constraints
s	Path coordinate
b_i	Polynomial time parameters
p_i	Polynomial path parameters
J	Jacobian
J_a	Analytical Jacobian
B	Transformation matrix of Euler angle velocities to angular velocity vector
J^+	Moore-Penrose inverse

Denavit Hartenberg parameters

$D_{x,i}$	Translation matrix of link length
-----------	-----------------------------------

$R_{x,i}$	Rotation matrix of link twist
$D_{z,i}$	Transformation matrix of link offset
$R_{z,i}$	Rotation matrix of joint angle
a_i	Link length
α_i	Link twist
d_i	Link offset
H_i^j	Transformation matrix of frame i to frame j
R_i^j	Rotation matrix of frame i to frame j
T_i^j	Transformation matrix of frame i to frame j

Manipulator

$q_{m,i}$	Manipulator coordinate of joint i
$L_{m,i}$	Manipulator link length
m_i	Mass of link i
θ_i	Rotation of joint i
l_a, l_b, l_c	Lengths used to derive the geometric inverse of the manipulator
ϵ_i	Value used to derive the geometric inverse of the manipulator
K_i	Values used to define the inverse solutions
Φ	Angles used to derive the geometric inverse of the manipulator
ϕ	Entries of the rotation matrix used to derive the geometric inverse of the manipulator

Notations

s_θ	$\sin(\theta)$
c_θ	$\cos(\theta)$
k	Vector in z direction, $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$
ρ	Rank of an analytical Jacobian
U	Singular value decomposition matrix
Σ	Singular value decomposition matrix
V	Singular value decomposition matrix
σ	Eigenvalue roots of the singular value decomposition
F	Force
τ	torque
d_m	Radius of circle defining manipulator positions
E	Energy
i_{o_i}	Index of sampled optimization variable
$c_{i,var}$	Variation of optimization variables
o_i	Optimization variables

$\mathbf{C}(i, \mathbf{F})$	Case i with fixed predefined initial conditions
$\mathbf{C}(i, \mathbf{R})$	Case i with random initial conditions
$c_{\ddot{r}}$	Constant related to the acceleration of the mobile robot platform
$c_{\ddot{\theta}}$	Constant related to the angular acceleration of the mobile robot platform
$c_{\dot{\theta}}$	Constant related to the angular velocity of the mobile robot platform
c_g	Constant related to the mass of the mobile robot platform

Sub- and superscripts

ω	angular velocity
\mathbf{F}, i	Final task motion configuration of the end effector
\mathbf{S}, i	Starting task motion configuration of the end effector
v	Linear velocity
w	Wrist of the manipulator
+	Upper limit of a variable
-	Lower limit of a variable
ee	End Effector
m	M anipulator base
mrp	M obile R obot P latform
rest	Rest configuration of manipulator
task	Task
W	W orld fixed frame

Mobile robot platform

L_i	MRP dimensions
$L_{x,i}$	MRP dimensions in x -direction
$L_{y,i}$	MRP dimensions in y -direction
I_i	Inertia matrix

Contents

1	Introduction	11
1.1	Background	11
1.2	Problem definition	12
1.3	Objective	12
1.4	Structure	12
2	Literature review	15
2.1	Summary	18
2.2	Conclusion	18
3	Background information/preliminaries	19
3.1	System description	19
3.2	Models	20
3.2.1	Forward kinematics	20
3.2.2	Inverse kinematics	25
3.2.3	Dynamic model	29
3.3	Summary	30
4	Trajectory generation and optimization	31
4.1	The task motion	31
4.2	Trajectory generation method	33
4.2.1	The task motion	33
4.2.2	The initialization of the end effector	33
4.2.3	Mobile manipulator motion	34
4.3	First optimization: MRP path and stopping configurations	34
4.3.1	The MRP path as function of optimization variables	34
4.3.2	MRP configuration and path evaluation	37
4.3.3	The first optimization algorithm	40
4.4	Second optimization: Trajectory	41
4.4.1	Approach	42
4.4.2	Manipulator path generation	42
4.4.3	From path to trajectory	43
4.4.4	Solution evaluation	44
4.4.5	The second optimization algorithm	46
4.5	Summary	46
5	Optimized paths and trajectories	47
5.1	The benchmark trajectory generation method	47
5.2	Workspace and tasks	48
5.2.1	The considered cases	48
5.3	Result of the first optimization	50
5.3.1	The resulting MRP path	50
5.4	Second optimization: Trajectory comparison	52
5.4.1	Expected solutions	53
5.4.2	The resulting energy consumption	53
5.4.3	Conclusion and next comparison to consider	55
5.5	Achieved cost reduction of the second optimization	55

5.5.1	Expected energy reduction	55
5.5.2	Solution comparison	55
5.5.3	Observations and how to continue	58
5.6	The effect of randomized initial solutions	58
5.6.1	Expected differences	58
5.6.2	Optimization results	59
5.6.3	Comparison of the optimized solutions	61
5.6.4	Concluding remarks	64
5.7	Summary	64
6	Conclusions and recommendations	65
6.1	Conclusions	65
6.2	Recommendations	65
A	Toppling of the MRP during actuation	67
B	Parameters	71

Chapter 1

Introduction

1.1 Background

Automation has been a significant topic of interest in recent years [1] which has only grown due to the recent developments on a world wide scale. In the current situation a large number of companies are having difficulty with providing safe working environments for their employees. As a result, work output is reduced. This is not the case for working environments which are mostly occupied by automated robots. However, with current technologies it is not possible to automate every process.

Automation is already widely applied in manufacturing processes. These processes include assembly, machining and picking and placing. In current automated manufacturing lines manipulators perform fixed tasks at fixed locations for a large number of repetitions. Therefore, manipulators are stationary and programmed manually to perform a cyclic motion [2]. When a different order is processed the manufacturing line is altered. In the case of variable processes, humans are employed [3]. As a result, human error is introduced which is disadvantageous for the process duration. Moreover, certain tasks can be ergonomically unfavorable due to lifting and carrying of heavy products [3].

By automating variable processes it is possible to work towards the factory of the future where every process is automated. Flexible manufacturing allows products to be handled on demand rather than producing and storing them in bulk. Additionally, by using flexible setups it is possible to produce the same products with less equipment. This means storage costs are reduced and factories can be laid out more efficiently. When optimizing the cycle time of a manufacturing process it is required to leave a margin for human error which is disadvantageous for the optimization. Fully automated processes are known exactly and thus a theoretical optimal setup can be defined.

At the Brainport Industries Campus an innovation program is currently active in automating variable manufacturing processes [4]. This innovation program is called "Flexible Manufacturing". The Flexible Manufacturing program focuses on automation of the manufacturing processes with small batch sizes and high customization while minimizing downtime due to changing of the task. This means manipulators will have to be able to perform a large variety of tasks at varying locations automatically and efficiently. Therefore, one of the sub-projects in the flexible manufacturing program is the development of a mobile manipulator. A mobile manipulator would be able to perform varying tasks at varying locations and thus replace humans in the manufacturing process to automate it further. For the mobile manipulator to function properly it has to be able to define and execute trajectories automatically based on the tasks and their positions.

One of the major differences between stationary and mobile robotic systems is the power supply. A stationary robot can be powered by cable and is thus not limited in its use by the energy consumption. A mobile manipulator has to transport its own energy source, commonly in the form of electrical energy in a battery. Therefore, a consideration has to be made in terms of energy consumption. A stationary robot can be programmed to work as fast as the power supply can handle while for a mobile manipulator the battery would be drained rapidly. This can be resolved by increasing the battery capacity. However, increasing the weight of a mobile system increases the energy consumption. The work output of a mobile manipulator is therefore a function of the tasks, battery capacity, system specifications and charging rate.

1.2 Problem definition

A trajectory generation method has to be defined for the mobile manipulator provided by VinciTech such that a task sequence can be performed. A task sequence consists of multiple, connected or unconnected, task motions of the end effector of the manipulator. Each task motion of the end effector is prescribed in six degrees of freedom, corresponding to the position and orientation in a three dimensional environment. The task motions are flexible which means hardcoding a trajectory is not possible. Moreover, manually providing trajectories based on the flexible task is ineffective due to the downtime of a manipulator [4]. To be able to adapt to varying tasks an automated trajectory generation method is required. Multiple requirements hold for these trajectories. Most important of all, the tasks are to be performed somewhere along the generated trajectory. Furthermore, the robot should not collide with the objects and structures, should not experience difficulties due to singular configurations and should not be actuated beyond the limits of the actuators. These requirements are needed such that the trajectory provided for the mobile manipulator is feasible. On top of the requirements of feasibility a performance demand is imposed. This demand is that the energy consumption is minimized. The mobile manipulator requires to be charged resulting in downtime. By minimizing the charging frequency throughput and overall efficiency is increased.

By placing a manipulator on a moving platform, the number of degrees of freedom is increased beyond six. Task motions are defined for the end effector in six degrees of freedom and thus the actuation of the end effector is redundant. This means an infinite amount of begin and end configurations as well as trajectories can be formulated for each task motion. Automation of the trajectory generation is therefore not an obvious process. Furthermore, it is impossible to compute every trajectory. Therefore, the trajectories should be generated and optimized in an efficient manner.

1.3 Objective

The goal of this research is to automate the trajectory generation for a mobile manipulator such that it is able to perform various tasks at multiple locations. The trajectory is optimized for energy consumption since the energy capacity of the mobile manipulator is limited. Furthermore, the trajectory is evaluated for physical constraints, both related to the robot and the environment. The optimization is performed based on simulations of generated motion sequences defined by trajectories. These simulations require the kinematic and dynamic description of the system. To be able to achieve the goal of this research a number of sub-goals is defined.

First of all a **kinematic and dynamic model of the mobile manipulator** are required. The kinematic model is needed to be able to relate robot configurations to Cartesian coordinates. This is needed to define the position of the robot in the environment and thus evaluate collisions. Furthermore, the kinematic model is used to transform required task motions to joint trajectories. To **avoid extensive simulations** of infinitely many solutions a strategy is required to be able to cope with the **redundancy**. A **trajectory generation method** can then be applied to convert a task sequence to a robot trajectory. An **optimization algorithm** should then provide the mobile manipulator with a trajectory which is evaluated for feasibility and optimized for energy consumption.

1.4 Structure

This thesis is constructed as follows:

In Chapter 2 a literature study provides insight in the problem. Previous studies are mentioned and used to provide a substantiated decision making process which contributes to an approach which is likely to result in a working principle.

In Chapter 3 more information about the mobile manipulator is given. This information is used to derive the kinematic and dynamic model used for the simulations during optimization.

In Chapter 4 the trajectory generation method and optimization method are explained in detail. Moreover, the complete approach is given in terms of a redundancy resolution.

In Chapter 5 the results of the optimization are discussed. Multiple considerations and comparisons are made to conclude or rule out behavior.

In Chapter 6 conclusions are drawn and considerations for future work are made based on the gained experiences.

Chapter 2

Literature review

The modeling and optimization of manipulator trajectories has been done extensively in the past. Typically optimization is performed to decrease cycle times for industrial manipulators to increase their efficiency in terms of output [5] [6] [7] [8] [9] [10]. However, over the past few years interest has shifted towards energy optimization due to environmental concerns, energy cost and more strict regulations [11][12]. [13]. In industry, electric drives consume two thirds of the total energy which suggests a significant impact can be made by optimization [14] [15].

Manipulator motions often consist of constrained task motions and point-to-point (PTP) motions to initialize for the task motion or transport goods. Especially during these PTP motions optimization can be effective due to the freedom of motion [11]. Various optimization strategies have been used effectively to reduce the energy consumption of manipulators. In [11] the power consumption of the motors is modeled in detail. While in [16] the power consumption is simplified to the square of the torque delivered by the actuators. For PTP motions it can be seen that analytical formulations are used to define trajectories [17] [11] [18]. By doing so the smoothness of the trajectories can be chosen. Moreover, the complete trajectory is considered during optimization which means global optima can be found.

By placing a manipulator on a mobile platform the freedom of actuation is increased. As a result redundancy is introduced and the optimization problem becomes more complex. Moreover, energy consumption becomes more important due to the limited amount of energy which can be transported. In [19] [20] [21] [22] it can be seen that for redundant manipulators the Moore-Penrose inverse is used to transform end effector motions to joint trajectories. Optimization can be performed by varying the null space. The null space alters the joint trajectories while maintaining the end effector motion which is required to perform the tasks. In [20] the null space is optimized such that the joint limits of a redundant manipulator are avoided while performing a tool tip task. In [19] the null space of a redundant surgical manipulator is used to satisfy constraints, follow a trajectory and avoid collisions. In [22] optimization of the null space is applied to a mobile manipulator. This mobile manipulator is an underwater vehicle equipped with a manipulator. Optimization is applied to actuate the complete system as an expert pilot would to avoid unwanted kinematic behavior but also to keep the underwater vehicle steady and keep the end effector within the field of vision of the pilot.

An alternative to the Moore-Penrose inverse is constraining a number of actuators such that the remaining degrees of freedom match the number of degrees of freedom of the task motion. What this effectively means for a mobile manipulator is that the manipulator is separated from the mobile platform. This method is employed for applications like mobile machining and painting manipulators [23] [24] [25]. Due to the increased inaccuracies when actuating the manipulator and the mobile platform simultaneously, it can be required for the mobile platform to be stationary during task motions as in [23]. By fixing the base position of the manipulator, the joint trajectory is defined by the task motion and inverse kinematics. This requires an analysis of the base position such that the task motion is executable [23] [24] [26] [27] [25]. In [25] the trajectories are evaluated to reduce the torque load, in [23] [26] [25] the performance is evaluated by the manipulability to reduce the effort during task motions. In [28] the mobile platform is optimized for tipping stability during manipulator motions. In [29] the goal is to optimize the mobile manipulator positions to reduce the number of repositionings. In all mentioned cases feasibility of the task motion is considered when defining the position of the mobile platform.

By making use of the Moore-Penrose inverse the joint trajectories can be completely defined by an optimization algorithm as function of the task motions, constraints and optimization variables. This can be beneficial for hyper-redundant manipulators [30, p. 265] for which it can be intuitively hard to manually compose a trajectory. It is expected that optimization of the null space results in counter-intuitive trajectories as can be seen in [22] where optimization is specifically required to be able to operate the system without any unwanted behavior. Moreover, by implementing inverse solutions it is required to actively avoid singular configurations at all times as the Moore-penrose inverse often results in singular configurations [20]. By separating the actuation of the mobile platform and manipulator it is possible to control the individual systems more intuitively and possibly more effectively. This is why separation of the systems is the preferred method to resolve the redundancy.

In [31] it can be seen that path planning of a mobile manipulator is most often achieved by implementing a genetic algorithm (GA), a type of evolutionary algorithm (EA). The feasibility analysis of mobile platform positions results in unconnected regions in the search space. As a result, gradient based methods depend on the initial conditions of a solution [32] [26]. A benefit of a GA is the ability to globally search unconnected regions of solutions. In [26] [25] [27] it can be seen that optimization of mobile manipulator placements is performed using genetic algorithms. In [33] a similar problem is solved where the relative position between a manipulator and object is optimized.

A benefit of the GA is that it can approach the global optimal solution. This is because the EA can be used to evaluate the complete trajectory. Moreover, the black box formulation of the GA allows for the GA to be used in a variety of situations which is necessary to both evaluate a cost function and the feasibility.

Path planning of the mobile manipulator is based on PTP motions as result of the separation of the mobile platform and the manipulator. The mobile platform has to be positioned such that the manipulator is able to perform its task motion. A path can be formulated as function of the mobile platform configurations. The manipulator then has to be initialized for the task motion. A commonly used method for manipulator path generation is the polynomial description as in [34] [35] [36] [37]. A polynomial can be defined as function of constraints based on position or a derivative of any order [38, p. 188-199]. The number of constraints defines the required order of the polynomial. For optimization purposes the order of the polynomial is increased such that the trajectory can be varied. In [39] a comparison is made between polynomials with respect to their degree and performance. It can be seen that higher order polynomials have a better result but significantly increase the computation time.

Closely related to the polynomial formulation is the spline formulation. A spline consists of multiple polynomials which allows more freedom to vary the trajectory [38, p. 188-199]. An example of a spline is shown in Figure 2.1. This method can be used when a polynomial description does not allow the required flexibility when defining a trajectory. This can be seen in [40] where multiple control points are necessary to avoid collisions. In [18] a spline formulation is used because of the large number of constraints.

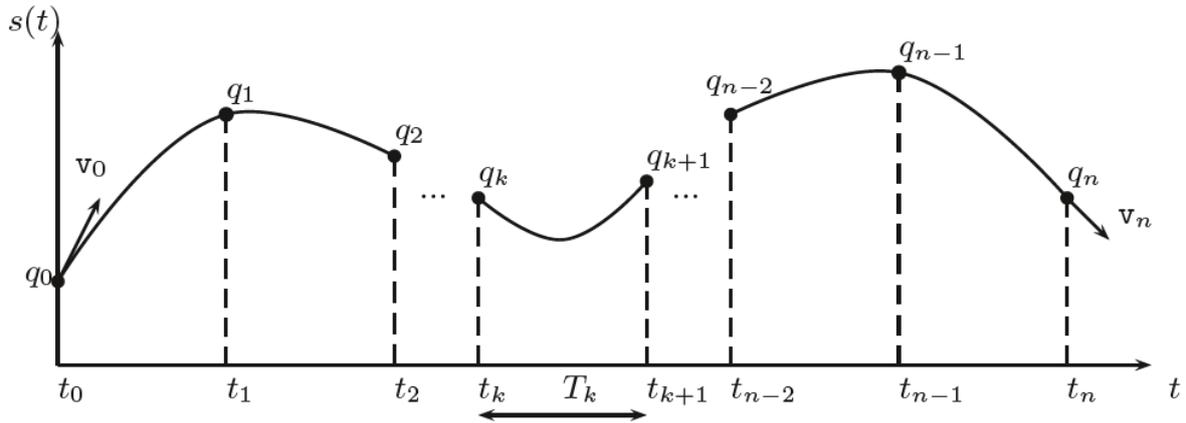
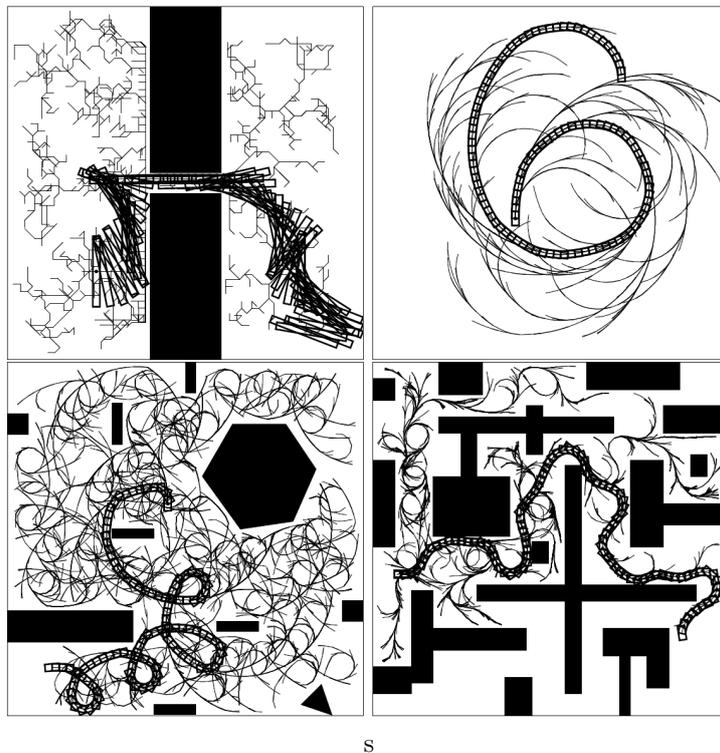


Figure 2.1: Example of a spline consisting of n segments defined by position and velocity constraints [38]

In [41] configurations are optimized and connected by a path generated by a rapidly exploring random tree (RRT). A RRT is a sampling based method which searches the environment randomly [42]. The samples allow paths to be generated in complex environments as in [43] and the examples shown in Figure 2.2. Figure 2.2 shows how potential paths are generated until a complete path connects the initial and goal configuration. For optimization purposes this method is less suitable because random search is not beneficial. Moreover, to be able to optimize solutions the trajectories should be a function of chosen variables. Based on the problem specific environment a polynomial trajectory description is chosen. An analytical trajectory description is chosen to be able to optimize globally. The spline formulation is not chosen because the environment is not significantly cluttered or complex.



s

Figure 2.2: Various examples of RRT variations and applications for a 2D body moving through a 2D environment [42]

2.1 Summary

From previous studies was seen that optimization using the Moore-Penrose inverse kinematics and the null space could result in unwanted behavior of a system. Since the mobile manipulator consists of two distinct systems it is expected that the unwanted behavior is likely to occur. Separation of the manipulator and the moving platform means that the actuation of both can be defined individually. Moreover, separating requires a feasibility analysis in relation to manipulator base positions and task motions. To do so, a genetic algorithm is shown to be a favorable optimization method. An analytical description of the joint trajectories is considered so that a trajectory is optimized globally. A polynomial description is chosen as it seems suitable for the problem specific situation.

2.2 Conclusion

To generate trajectories for the mobile manipulator a genetic algorithm will be implemented. The moving platform and manipulator are separated. As a result, trajectories are generated based on base positions of the manipulator from which the task motions are performed and the constrained task motions. The base positions have to be evaluate for feasibility during task motions. The trajectories are described by polynomials based on point-to-point motions.

Chapter 3

Background information/preliminaries

In this chapter the mobile manipulator is described. Furthermore, the kinematic and dynamic models are described. The kinematic model provides information about the relation between the actuator motions and system motions. The dynamic model relates the motions to forces and torques. Both models are required in order to properly define trajectories.

3.1 System description

The mobile manipulator consists of two distinct systems. The first system is a manipulator and the second is a mobile robot platform referred to as MRP. The manipulator is equipped with an end effector. An end effector is a tool at the end of a manipulator used to perform tasks.

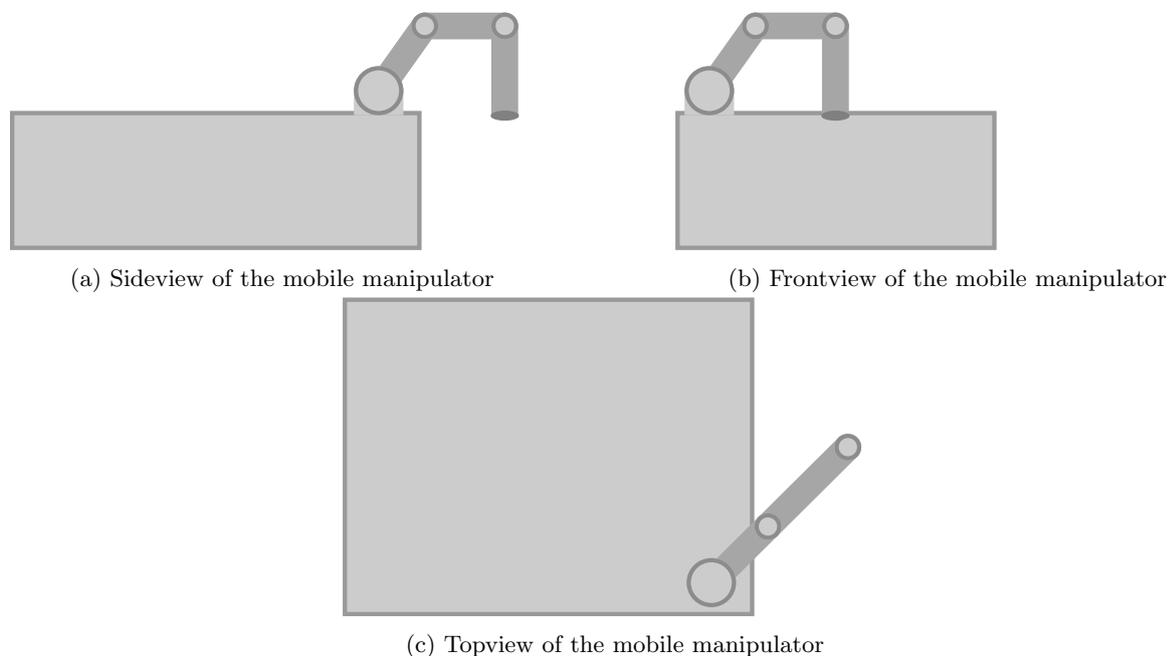


Figure 3.1: Schematics of the VinciTech mobile manipulator

The manipulator consists of three links and six joints granting six degrees of freedom. Each joint is actuated by a stepper motor. The motors are situated at the base of the manipulator and are connected to the joint by two cables running over a series of pulleys. This configuration of the motors allows for a lightweight design of the manipulator because the significant weight of the motors is not part of the

moving load. A drawback of the transmission is the non-linear elongation of the cables, friction and slack.

The mobile robot platform is an omnidirectional vehicle with four wheels, one at each corner of the rectangular body. This means that the MRP is overactuated but less likely to tip over. Each wheel of the MRP is driven by two stepper motors, one for steering and one to drive. As a result it is not possible to instantly move in any direction because the wheels have to be aligned in the correct direction. The manipulator is placed on the MRP such that the end effector is able to reach its task positions.

The current manipulators are controlled based on a kinematic model without active consideration of the dynamic behavior. This is possible because the stepper motors are driven based on a reference position instead of a torque. The position control can be done in open and closed loop depending on the application and reliability of the system. The stepper motors are sufficiently strong so they do not slip, which implies the possibility of an open loop control. However, because of the non-linear behavior of the cables a closed loop modified PI-controller has to be implemented such that the end effector follows its desired trajectory. This control strategy is sufficient for the current motion profiles where acceleration and velocity are limited.

3.2 Models

The mobile manipulator is described by two models to be able to generate and evaluate trajectories. The first model is the kinematic model to describe motions of one coordinate frame in the other. The forward kinematic model describes the end effector trajectory as function of the joint trajectories. The motion of the end effector can be fully constrained to execute tasks. For this reason the inverse kinematics is formulated which describes the joint trajectories as function of the end effector trajectory. The second model is the dynamic model which describes the forces and torques as function of the kinematics. The dynamic model is used to compute the energy consumption as function of actuator torques.

3.2.1 Forward kinematics

The mobile manipulator consists of two systems for which the models are derived differently. The MRP motions and positions can be described relatively straightforward in the Cartesian coordinate system by Euclidean transformations, namely, two translations and one rotation. This is because the MRP can drive in x - and y -direction while rotating around the z -axis. The end effector position and orientation is described by a number of translations and rotations specified by the Denavit Hartenberg parameters. This method allows for a general expression of the end effector configuration with six variables, namely the joint angles. The intermediate steps of the Denavit Hartenberg convention provide the position and orientation of each of the intermediate frames which is useful for the dynamic model.

MRP position model

The coordinates of the MRP describe both the position of the MRP itself and the position and orientation of the manipulator. The coordinates are expressed in the fixed world frame and refer to the center of the MRP at zero height. The position of the center of the MRP is described by the position vector

$$r_{\text{mrp}} = \begin{bmatrix} x_{\text{mrp}}^{\text{W}} & y_{\text{mrp}}^{\text{W}} & z_{\text{mrp}}^{\text{W}} \end{bmatrix}^{\text{T}}, \quad (3.1)$$

where $z_{\text{mrp}}^{\text{W}}$ remains zero at all times. The orientation of the MRP is given by $\theta_{\text{mrp}}^{\text{W}}$ which is the rotation around the z -axis. The coordinates are in relation to the fixed world frame W as indicated by the superscript. The position and orientation of the MRP also refer to the center and orientation of the MRP fixed frame which is used later.

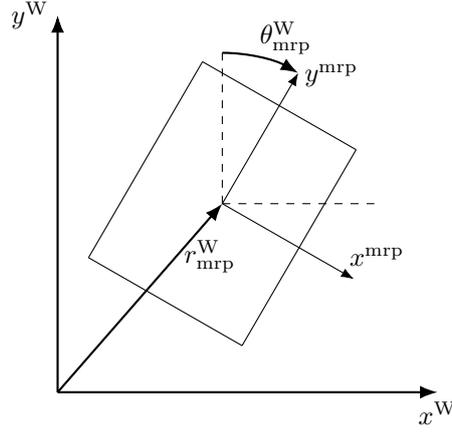


Figure 3.2: Schematic of the MRP position and orientation

The configuration of the MRP can be used to describe the position of the base of the manipulator. The manipulator is not positioned at the center of the MRP and thus both position and orientation define the manipulator base position. The base position of the manipulator is defined by r_m which is also the origin of frame 0 which is used in the kinematic model of the manipulator.

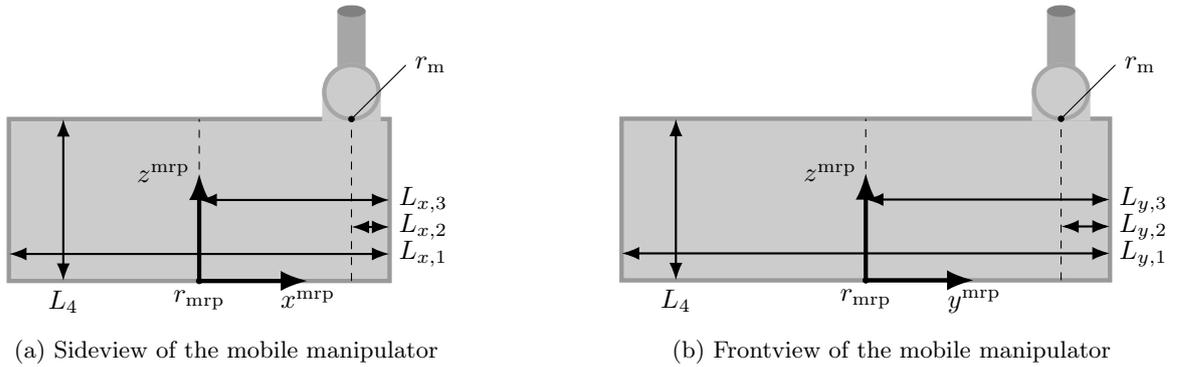


Figure 3.3: Schematics of the mobile manipulator

The manipulator base does not move with respect to the MRP. The constant position of the base of the manipulator on the MRP is defined by the MRP dimensions as

$$r_m^{\text{mrp}} = \begin{bmatrix} L_{x,3} - L_{x,2} & L_{y,3} - L_{y,2} & L_4 \end{bmatrix}^\top. \quad (3.2)$$

The MRP rotates around its z -axis with θ_{mrp} which results in the rotation matrix R_{mrp}^W as in

$$R_{\text{mrp}}^W = \begin{bmatrix} \cos(\theta_{\text{mrp}}) & -\sin(\theta_{\text{mrp}}) & 0 \\ \sin(\theta_{\text{mrp}}) & \cos(\theta_{\text{mrp}}) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.3)$$

Because the manipulator base is fixed to the MRP the rotation matrices R_{mrp}^W and R_m^W are identical and thus R_m^{mrp} is equal to the identity matrix. The frames m and mrp share the same orientation but are not identical as they are translated. The position of the manipulator with respect to the origin of the world fixed frame is defined by

$$\begin{aligned} r_m^W &= r_{\text{mrp}}^W + {}^{\text{mrp}}r_m^W, \\ r_m^W &= r_{\text{mrp}}^W + R_{\text{mrp}}^W r_m^{\text{mrp}}, \end{aligned} \quad (3.4)$$

where ${}^{\text{mrp}}r_m^{\text{W}}$ indicates the position of the manipulator expressed in the world frame relative to the position of the MRP frame.

Denavit Hartenberg

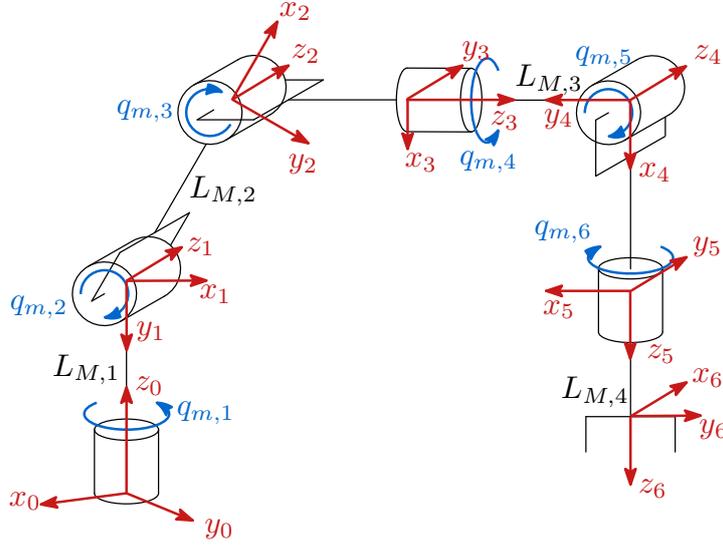


Figure 3.4: Schematic of the manipulator frames corresponding to the Denavit Hartenberg convention

A conventional method for describing the kinematics of a manipulator is the Denavit Hartenberg convention [38, p. 76]. In this notation the position of the end effector is described by a series of transformations corresponding to the joint frames of the manipulator represented by the schematic of Figure 3.4. The transformations result from matrix multiplications describing rotation and transformation of one frame to the next. The actuation of a joint is performed along the z -axis of each intermediate frame and can be either prismatic or revolute. Each transformation is described by four parameters specified in Table 3.1. Depending on the joint type either d_i or θ_i is variable. The manipulator consists of only revolute joints and thus all θ_i are variable. The specified θ_{off} in Table 3.1 are the offsets for the initial configuration. The generalized coordinates of the VinciTech manipulator are defined as

$$q_{m,i} = \theta_i + \theta_{\text{off},i} \quad (3.5)$$

such that the manipulator is in full upright extension for $\theta_i = 0 \forall i \in [1, 8]$.

Link	d_i	a_i	α_i	$\theta_{\text{off},i}$
1	$L_{m,1}$	0	$-\frac{\pi}{2}$	0
2	0	$L_{m,2}$	0	$-\frac{\pi}{2}$
3	0	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$
4	$L_{m,3}$	0	$-\frac{\pi}{2}$	0
5	0	0	$\frac{\pi}{2}$	0
6	$L_{m,4}$	0	0	0

Table 3.1: Denavit Hartenberg parameters of the VinciTech manipulator

The parameters a_i , α_i , d_i and θ_i respectively specify the link length, link twist, link offset and joint angle [38][p. 77]. The four Denavit Hartenberg parameters of each frame transformation are filled into the following equations

$$\begin{aligned}
R_{x,i} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D_{x,i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
R_{z,i} &= \begin{bmatrix} \cos(q_{m,i}) & -\sin(q_{m,i}) & 0 & 0 \\ \sin(q_{m,i}) & \cos(q_{m,i}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D_{z,i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{3.6}$$

The transformation matrix is formulated by multiplications of the individual parameter transformations as in

$$\begin{aligned}
H_i^{i-1} &= R_{z,i} D_{z,i} D_{x,i} R_{x,i}, \\
H_i^{i-1} &= \begin{bmatrix} R_i^{i-1} & r_i^{i-1} \\ \mathbf{0}^{1 \times 3} & 1 \end{bmatrix}.
\end{aligned} \tag{3.7}$$

The transformation of frame i to $i+1$ results from a multiplication. Furthermore, multiplying with the inverse of the transformation matrix results in the transformation for frame $i+1$ to frame i . Therefore, the transformation matrices can be multiplied sequentially to express each frame in any other frame. The matrix T_i^j expresses the orientation and position of frame i in frame j as in

$$\begin{aligned}
T_j^i &= H_{i+1}^i \times H_{i+2}^{i+1} \times \dots \times H_j^{j-1} \\
T_i^j &= H_{j-1}^j \times H_{j-2}^{j-1} \times \dots \times H_i^{i+1} \\
T_i^j &= (H_j^{j-1})^{-1} \times (H_{j-1}^{j-2})^{-1} \times \dots \times (H_{i+1}^i)^{-1}.
\end{aligned} \tag{3.8}$$

The position of the end effector in frame W is defined by the Denavit Hartenberg notation of the position of the sixth frame plus the position of the base of the manipulator.

$$\begin{aligned}
r_{ee}^m &= r_6^0 \\
r_{ee}^W &= r_m^W + {}^m r_{ee}^W. \\
r_{ee}^W &= r_m^W + R_m^W r_{ee}^m.
\end{aligned} \tag{3.9}$$

The position of the end effector defined by the DH-parameters is expressed in the manipulator base frame 0. This frame is identical to frame m in both position and orientation.

Kinematic model

With the Denavit Hartenberg convention the position of the end effector is translated from generalized coordinates of the manipulator to Cartesian coordinates. However, for trajectories velocities and accelerations are required. The geometric Jacobian J translates the velocity of the joints to the velocities in Cartesian coordinates. The frame of the end effector is frame 6 as would result from the Denavit Hartenberg convention. The Jacobian is split in two parts, the linear velocity and angular velocity matrix denoted by v and ω respectively:

$$\begin{aligned}
\dot{r}_6^0 &= v_6^0 = J_v \dot{q}_m \\
\dot{\omega}_6^0 &= J_\omega \dot{q}_m.
\end{aligned} \tag{3.10}$$

The matrices can be combined to find the complete six by six Jacobian of the manipulator. The row size of the Jacobian is defined by the number of joints and the column size is defined by the position and orientation, both in three directions:

$$\xi = J\dot{q}_m \text{ with } \xi = \begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} \text{ and } J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}. \quad (3.11)$$

The revolute actuation can be described by the vectors z_i^j related to each frame. The vector is found by multiplying the rotation matrix of the corresponding frame with $k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$. The coordinates are defined by the DH convention where actuation is performed around the z -axis, hence the usage of vector k such that

$$\begin{aligned} z_{i-1}^0 &= R_{i-1}^0 k, \\ J_\omega &= \begin{bmatrix} z_0 \dots z_{n-1} \end{bmatrix}, \end{aligned} \quad (3.12)$$

The velocity of the end effector can be found by the chain rule for differentiation as in

$$v_n^0 = \sum_{i=1}^n \frac{dr_n^0}{dt} = \sum_{i=1}^n \frac{\partial r_n^0}{\partial q_{m,i}} \frac{dq_{m,i}}{dt} = \sum_{i=1}^n \frac{\partial r_n^0}{\partial q_{m,i}} \dot{q}_{m,i}. \quad (3.13)$$

The linear velocity Jacobian J_v is thus

$$J_v = \frac{\partial r_n^0}{\partial q_m}. \quad (3.14)$$

The geometric Jacobian describes the position and rotation of the end effector by the six joint velocities. This notation is not intuitive to prescribe a required configuration of the end effector and thus the analytical Jacobian is introduced. The analytical Jacobian is based on a minimal representation for the orientation of the end effector frame. The position of the end effector is defined in x - y - and z -direction and the orientation is described by Euler angles. The prescribed end effector configuration is then

$$q_{ee} = \begin{bmatrix} r_{ee} & \psi \end{bmatrix}^\top \quad (3.15)$$

with $r_{ee} = \begin{bmatrix} x_{ee} & y_{ee} & z_{ee} \end{bmatrix}^\top$ and $\psi = \begin{bmatrix} \psi_1 & \psi_2 & \psi_3 \end{bmatrix}$. The Euler angles are around the z - y - and z -axis respectively. The derivative of the end effector configuration is described by the analytical Jacobian and the joint velocities as in

$$\dot{q}_{ee} = J_a \dot{q}_m. \quad (3.16)$$

The angular velocity of the end effector in frame m results from $\dot{\psi}$ multiplied by their corresponding rotation matrices described by the Euler angles similar to (3.12).

$$\omega = \begin{bmatrix} c_{\psi_3} s_{\psi_2} \dot{\psi}_1 - s_{\psi_3} \dot{\psi}_2 \\ s_{\psi_3} s_{\psi_2} \dot{\psi}_1 + c_{\psi_3} \dot{\psi}_2 \\ \dot{\psi}_3 + c_{\psi_2} \dot{\psi}_1 \end{bmatrix} = \begin{bmatrix} c_{\psi_3} s_{\psi_2} & -s_{\psi_3} & 0 \\ s_{\psi_3} s_{\psi_2} & c_{\psi_3} & 0 \\ c_{\psi_2} & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \end{bmatrix} = B(\psi) \dot{\psi} \quad (3.17)$$

The angular velocity described by the derivatives of the Euler angles is equal to the angular velocity described by the joint velocities:

$$\begin{bmatrix} v_n^0 \\ \omega_n^0 \end{bmatrix} = \begin{bmatrix} r_{ee}^m \\ B(\psi) \dot{\psi} \end{bmatrix} = J \dot{q}_m \text{ with } B(\psi) = \begin{bmatrix} c_{\psi_3} s_{\psi_2} & -s_{\psi_3} & 0 \\ s_{\psi_3} s_{\psi_2} & c_{\psi_3} & 0 \\ c_{\psi_2} & 0 & 1 \end{bmatrix}. \quad (3.18)$$

Rewriting for J_a with (3.16) then yields the expression for the analytical Jacobian based on the geometric Jacobian:

$$J_a = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & B^{-1}(\psi) \end{bmatrix} J. \quad (3.19)$$

The matrix $B(\psi)$ should be invertible and thus the determinant of $B(\psi) \neq 0$. The determinant of $B(\psi)$ is zero for $\sin(\psi_2) = 0$. This is the case when the axis of rotation of ψ_1 and ψ_3 are lined up.

3.2.2 Inverse kinematics

The inverse kinematic model describes the joint angles as function of the end effector configuration. This is needed to generate trajectories for defined end effector motions. To be able to compute the inverse kinematics the system has to be non-redundant or an infinite amount of solutions is possible. This means that the number of equations to solve should be equal to the number of actuator variables. For a manipulator with six joints and end effector with six degrees of freedom the number of equations and actuators is equal. The inverse kinematics of the manipulator is solved at position level and at velocity or acceleration level. Each of these approaches has its own application.

Geometric inverse

The position inverse of the end effector is required to evaluate the reachability of end effector positions in relation to the joint limits of the manipulator. By evaluating the inverse solution it can be avoided that the manipulator is ordered to move to an infeasible configuration. The inverse can be solved due to the non-redundancy of the manipulator. However, this does not mean that a single solution can be found. Due to the base rotation and elbow and wrist configuration for each two solutions are possible resulting in a total of eight solutions. First of all to compute the inverse solution the position and orientation of frame 6, the end effector frame, must be provided. In that case the entries of T_6^0 are known as in

$$T_6^0 = \begin{bmatrix} \varphi_{1,1} & \varphi_{1,2} & \varphi_{1,3} & x_6^0 \\ \varphi_{2,1} & \varphi_{2,2} & \varphi_{2,3} & y_6^0 \\ \varphi_{3,1} & \varphi_{3,2} & \varphi_{3,3} & z_6^0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R_6^0 & r_6^0 \\ \mathbf{0}^{1 \times 3} & 1 \end{bmatrix}. \quad (3.20)$$

The matrix T_6^0 is described by six variables. The three Euler angles describe the rotation matrix or orientation of the end effector with respect to the zero frame. The rotation matrix is the three by three matrix consisting of the parameters $\varphi_{i,j}$. The vector r_6^0 describes the position of the end effector with respect to the base of the manipulator which are the remaining three variables.

From the end-effector frame some relations can be derived. For instance, the orientation of the third link multiplied by the link length $L_{m,4}$ provides information about the relative position of the wrist joint in relation to the end-effector position. Since the end-effector position is known we can derive the position of the wrist link in the zero frame as vector x_w . The third link coincides with the z -axis of the end effector frame and thus

$$\begin{aligned} x_w &= x_6^0 - L_{m,4}\varphi_{1,3}, \\ y_w &= y_6^0 - L_{m,4}\varphi_{2,3}, \\ z_w &= z_6^0 - L_{m,4}\varphi_{3,3}. \end{aligned} \quad (3.21)$$

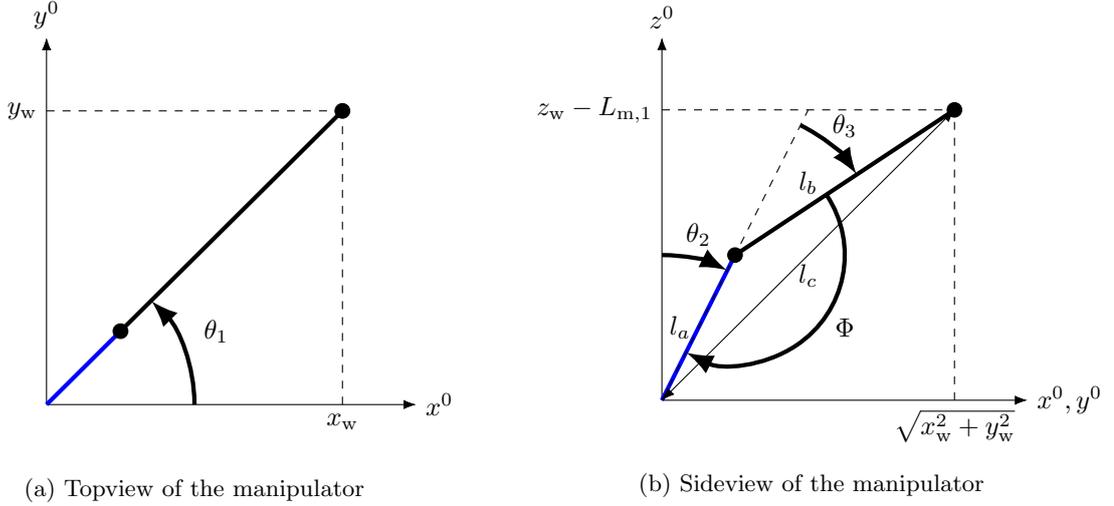


Figure 3.5: Geometric relations for computing θ_1 , θ_2 and θ_3

From the position of r_w it is possible to derive θ_1 to θ_3 from geometric relations as depicted in Figure 3.5. First θ_1 is found by using x_w and y_w . The position x_w^1 , which is the wrist position in frame 1, can be positive and negative for the same angle of θ_1 . The angle θ_1 can therefore be a function of r_w and $-r_w$ which corresponds to a base flip of π [rad]. The angle θ_1 is defined by the value of K_1 as in

$$\theta_1 = \text{Atan2}(K_1 y_w, K_1 x_w), \quad (3.22)$$

where $K_1 \in (-1, 1)$ is used to describe the eight possible solutions to the inverse kinematics for one end effector configuration. The angle θ_3 has two solutions corresponding to the elbow up and elbow down configuration. Therefore, for θ_3 the next variation is applied by means of the values of $K_2 \in (-1, 1)$. In both cases θ_3 is found using the cosine rule with the lengths of the links and the absolute distance to r_w . The cosine rule is formulated using

$$\begin{aligned} l_a &= L_{m,2} \\ l_b &= L_{m,3} \\ l_c &= \sqrt{x_w^2 + y_w^2 + (z_w - L_{m,1})^2}. \end{aligned} \quad (3.23)$$

By using the cosine rule the angle Φ between link 1 and 2 is found. The angles θ_3 and Φ sum to π [rad] and thus

$$\begin{aligned} l_c^2 &= l_a^2 + l_b^2 - 2l_a l_b \cos(\Phi), \\ \epsilon_1 &= \frac{l_c^2 - l_a^2 - l_b^2}{-2l_a l_b} \\ \Phi &= \text{Atan2}(\sqrt{1 - \epsilon_1^2}, \epsilon_1) \\ \theta_3 &= K_2(\pi - \Phi). \end{aligned} \quad (3.24)$$

The angle between the virtual link l_c and the z -axis is the sum of θ_2 and the angle between l_a and l_c . With the found θ_3 the angle θ_2 is

$$\theta_2 = \text{Atan2}(\sqrt{x_w^2 + y_w^2}, z_w - L_{m,1}) - \text{Atan2}(L_{m,3} \sin(\theta_3), L_{m,2} + L_{m,3} \cos(\theta_3)). \quad (3.25)$$

The remaining joint angles are not derived geometrically but result from the analytical description of the orientation. This analytical description is the result of the Denavit Hartenberg convention where the transformation matrices are multiplied. By premultiplying R_n^0 with the inverse of the rotation matrices as function of θ_1 , θ_2 and θ_3 the angles θ_4 , θ_5 and θ_6 are derived.

$$\begin{aligned}
R_1^0 R_2^1 R_3^2 R_4^3 R_5^4 R_6^5 &= R_6^0 \\
R_6^3 &= (R_3^0)^{-1} R_6^0 \\
\begin{bmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_6 s_4 - c_4 c_5 s_6 & c_4 s_5 \\ c_4 s_6 + c_5 c_6 s_4 & c_4 c_6 - c_5 s_4 s_6 & s_4 s_5 \\ -c_6 s_5 & s_5 s_6 & c_5 \end{bmatrix} &= \begin{bmatrix} c_1 c_2 c_3 - c_1 s_2 s_3 & -s_1 & c_1 c_2 s_3 + c_1 c_3 s_2 \\ c_2 c_3 s_1 - s_1 s_2 s_3 & c_1 & c_2 s_1 s_3 + c_3 s_1 s_2 \\ -c_2 s_3 - c_3 s_2 & 0 & c_2 c_3 - s_2 s_3 \end{bmatrix}^{-1} R_n^0 \quad (3.26) \\
\begin{bmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_6 s_4 - c_4 c_5 s_6 & c_4 s_5 \\ c_4 s_6 + c_5 c_6 s_4 & c_4 c_6 - c_5 s_4 s_6 & s_4 s_5 \\ -c_6 s_5 & s_5 s_6 & c_5 \end{bmatrix} &= \begin{bmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} \\ \phi_{2,1} & \phi_{2,2} & \phi_{2,3} \\ \phi_{3,1} & \phi_{3,2} & \phi_{3,3} \end{bmatrix}
\end{aligned}$$

The values of $\phi_{3,3}$, $\phi_{1,3}$ and $\phi_{3,1}$ are rewritten such that the joint angles are found. The angle θ_5 is dependent of the relative position of the wrist and thus

$$\begin{aligned}
\phi_{3,3} &= \cos(\theta_5), \\
\theta_5 &= \mathbf{sign}(y_w) \text{Atan2}(K_3 \sqrt{1 - \phi_{3,3}^2}, \phi_{3,3}). \quad (3.27)
\end{aligned}$$

Once more $K_3 \in (-1, 1)$ is used to describe all possible solutions. The solution of θ_4 is a function of θ_5 as in

$$\begin{aligned}
\phi_{1,3} &= \cos(\theta_4) \sin(\theta_5), \\
\epsilon_2 &= \frac{\phi_{1,3}}{\sin(\theta_5)}, \\
\theta_4 &= -\mathbf{sign}(x_w) \mathbf{sign}(y_w) \text{Atan2}(K_3 K_1 \sqrt{1 - \epsilon_2^2}, \epsilon_2). \quad (3.28)
\end{aligned}$$

The remaining angle θ_6 is a function of θ_2 , θ_5 and the relative position of the manipulator to the end effector configuration resulting in:

$$\begin{aligned}
\phi_{3,1} &= -\sin(\theta_5) \cos(\theta_6), \\
\epsilon_3 &= \frac{\phi_{3,1}}{-\sin(\theta_5)}, \\
\theta_6 &= \begin{cases} -\mathbf{sign}(x_w) \mathbf{sign}(y_w) K_1 K_2 K_3 \text{Atan2}(\sqrt{1 - \epsilon_3^2}, \epsilon_3), & \text{if } x_w^2 \leq L_{m,2}^2 \\ -\mathbf{sign}(\theta_2) \mathbf{sign}(x_w) \mathbf{sign}(y_w) K_1 K_3 \text{Atan2}(\sqrt{1 - \epsilon_3^2}, \epsilon_3), & \text{if } x_w^2 > L_{m,2}^2 \end{cases}. \quad (3.29)
\end{aligned}$$

Each solution of the inverse kinematics is now a function of the values of K_i for $i \in [1, 3]$. To be able to define all solutions eight unique combinations of K_i are required. Therefore, the values of K_i are defined by

$$\begin{aligned}
K_1 &= \begin{bmatrix} -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix} \\
K_2 &= \begin{bmatrix} -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \end{bmatrix} \\
K_3 &= \begin{bmatrix} -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix}. \quad (3.30)
\end{aligned}$$

Now that all possible inverse solutions are known it is possible to evaluate the feasibility of a required end effector configuration in relation to the joint limits. Only if none of the eight solutions is feasible it can be concluded that the end effector configuration cannot be reached.

Inverse Jacobian

The Jacobian describes the velocity of the end effector as function of the joint velocities. However, during tasks motions the joint trajectory has to be defined as function of the end effector trajectory. Therefore, the inverse of the analytical Jacobian is needed. In section 3.2.1 it was mentioned that the trajectory of the end effector can be described by six parameters, namely the position and orientation both in three directions. The manipulator consists of six joints which results in a square analytical Jacobian. The Jacobian as in (3.19) is then invertible as long as the determinant $\mathbf{Det}(J_a) \neq 0$ which results in

$$\dot{q}_m = J_a^{-1}(q)\dot{q}_{ee}, \quad (3.31)$$

where the joint trajectory is a function of the end effector position and orientation in Euler angles.

Moore-Penrose inverse

For a redundant system the position and orientation of the end effector are described by more than six coordinates. The rank of the Jacobian is 6 and the number of joint coordinates is m , with $m > 6$. As a result the Jacobian is non-square since the number of columns exceeds the number of rows. The non-square Jacobian cannot be inverted which relates to the infinitely many possible solutions. Instead the pseudoinverse or Moore-Penrose inverse of the Jacobian can be computed [38, p. 151]. The Moore-Penrose inverse gives one of the solutions to the inverse kinematics and is described as follows

$$\begin{aligned} \mathbf{I} &= (J_a J_a^\top)(J_a J_a^\top)^{-1} \\ \mathbf{I} &= J_a (J_a^\top (J_a J_a^\top)^{-1}) \\ \mathbf{I} &= J_a J^+ \\ J^+ &= J_a^\top (J_a J_a^\top)^{-1}. \end{aligned} \quad (3.32)$$

The inverse Jacobian solution is then implemented similar to the inverse solution of a non-redundant system where the joint velocity is a function of the end effector velocity as in

$$\dot{q}_m = J^+(q_m)\dot{q}_{ee}. \quad (3.33)$$

To be able to define all of the possible joint trajectories the Moore-Penrose inverse has to be extended. For each redundant coordinate a velocity relation between the joints can be defined for which the end effector velocity remains unaltered. This relation results from

$$\dot{q}_{ee} = J_a(q)\dot{q}_{n,i} = 0. \quad (3.34)$$

The vectors $\dot{q}_{n,i}$ for $i \in [1, m - 6]$ span the null space of matrix J_a which can be solved using the singular value decomposition [38, p. 153-154]:

$$\begin{aligned} J_a &= U\Sigma V^\top \\ \Sigma &= \left[\mathbf{I}^{6 \times 6} \cdot \sigma^{6 \times 1} \quad \mathbf{0}^{6 \times (m-6)} \right] \text{ for } \sigma_1 \geq \sigma_2 \cdots \geq \sigma_\rho > 0 \\ \rho &= \mathbf{rank}(J_a). \end{aligned} \quad (3.35)$$

The matrices $U^{6 \times 6}$ and $V^{m \times m}$ result from the eigen vectors of $J_a J_a^\top$ and $J_a^\top J_a$ corresponding to the eigen values σ_i^2 , see [38, p. 443-444]. The last $m - \rho$ columns of V correspond to the independent null space of J_a . Assume non-singular configurations only, which means full row rank of J_a and thus $\rho = 6$. The vectors $\dot{q}_{n,i}$ are a function of V :

$$\dot{q}_{n,i} = V_{i+6}^\top \text{ for } i = [1 \dots (m - 6)]. \quad (3.36)$$

Since J_a is a function of q_m the resulting singular value decomposition is also a function of q_m . This means that the independence of all $\dot{q}_{n,i}$ is defined by q_m as long as the system is in a non-singular configuration.

The joint trajectory of the redundant system is then defined by the Moore-Penrose inverse and the null space of the analytical Jacobian as in

$$\dot{q}_m = J^+(q_m)\dot{q}_{ee} + \sum_{i=1}^{m-6} N_i \dot{q}_{m,i}. \quad (3.37)$$

For each unique combination of the values of N_i a different inverse solution is described. The values of N_i could also be described as function of time.

3.2.3 Dynamic model

Based on the kinematics of the system a dynamic model can be constructed by means of the Newton Euler method. This dynamic model is needed to evaluate any property related to forces or moments. The Newton Euler method treats each body in turn by describing the linear and angular motion. The bodies are connected and thus the coupling forces and torques are present in the equations of each body in a recursive manner.

The forces and moments are not only necessary for the required actuator inputs but also for the reaction forces of the systems with their surroundings. Furthermore, the recursive formulation is effective for a serial manipulator. The Newton Euler method is often used for numerical calculation [38, p. 239]. Therefore, the Newton Euler method is chosen to describe the dynamic equations.

Newton Euler

The Newton Euler method of describing the dynamic model is constructed on a recursive formulation of the motion, forces and moments. The mobile manipulator is a serial system which means that the acceleration of each body is a summation of the accelerations of each body in relation to the previous body in the chain. This results in

$$\ddot{r}_{i+1}^j = \ddot{r}_i^j + R_i^j \ddot{r}_{i+1}^i, \quad (3.38)$$

with R_i^j corresponding to the rotation matrices defined in Section 3.2.1. The angular acceleration of each of the frames in the zero frame results from the forward recursion. The MRP rotation is in the frame W in this notation.

$$\begin{aligned} k &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top \\ \omega_{\text{mrp}}^{\text{W}} &= k \dot{\theta}_{\text{mrp}} \\ \dot{\omega}_{\text{mrp}}^{\text{W}} &= k \ddot{\theta}_{\text{mrp}} \\ \omega_i^{\text{W}} &= \omega_{i-1}^{\text{W}} + R_i^{\text{W}} k \dot{q}_{m,i} \\ \dot{\omega}_i^{\text{W}} &= \dot{\omega}_{i-1}^{\text{W}} + R_i^{\text{W}} k \ddot{q}_{m,i} + S(\omega) R_i^{\text{W}} k \dot{q}_{m,i} \end{aligned} \quad (3.39)$$

where $\omega_0 = \omega_{\text{mrp}}$ and $\dot{\omega}_0 = \dot{\omega}_{\text{mrp}}$. The matrix multiplication with $S(\omega)$ defines the derivative of the corresponding rotation matrix [38, p 124-128]. The rotation matrices are formulated as $R_i^{\text{W}} = R_0^{\text{W}} R_i^0$ where R_i^0 is described by the DH parameters. The matrix R_0^{W} is equal to R_m^{W} .

The Cartesian accelerations of the frames expressed in the world fixed frame W are as follows

$$\begin{aligned} \ddot{r}_m &= \ddot{r}_{\text{mrp}} + \dot{\omega}_{\text{mrp}} \times {}^{\text{mrp}}r_m^{\text{W}} + \omega_{\text{mrp}} \times (\omega_{\text{mrp}} \times {}^{\text{mrp}}r_m^{\text{W}}), \\ \ddot{r}_i &= \ddot{r}_{i-1} + \dot{\omega}_i \times {}^{i-1}r_i + \omega_i \times (\omega_i \times {}^{i-1}r_i), \\ \ddot{r}_{c,i} &= \ddot{r}_{i-1} + \dot{\omega}_i \times {}^{i-1}r_{c,i} + \omega_i \times (\omega_i \times {}^{i-1}r_{c,i}), \end{aligned} \quad (3.40)$$

where each coordinate is expressed in the world fixed frame W and the position of frame 0 is the position of the manipulator base and thus $r_0 = r_m$ and $\dot{r}_0 = \dot{r}_m$. The position $r_{c,i}$ defines the position of the

center of mass of the corresponding link.

The forces and moments between each of the bodies can now be derived by a backwards recursive formulation. Starting at the final link the forces and moments are derived based on the acceleration of the previous link. Since it is the final link the force F_{i+1} is non-existing. Continuing down the chain eventually the forces of the final body with the environment is derived. The variables are expressed in the frame of the subscript, hence the rotation matrices to transform the variables from frame $i + 1$ to frame i :

$$\begin{aligned} F_i &= m_i \ddot{r}_{c,i} - R_{i+1}^i F_{i+1} - m_i g_i \\ \tau_i &= R_{i+1}^i \tau_{i+1} - F_i \times r_{c,i}^i + (R_{i+1}^i F_{i+1}) \times {}^{i+1}r_{c,i}^i + I_i \dot{\omega}_i + \omega_i \times (I_i \omega_i). \end{aligned} \quad (3.41)$$

The matrices I_i represent the inertia matrices of each body. The vectors g_i correspond to the gravitational acceleration in frame i . The z -components of τ_i represent the applied moments of the manipulator actuators. The forces and torques working on the MRP are used to describe the inputs of the MRP actuation. For the representation as described above the model is assumed to be ideal and thus without any friction or other losses.

3.3 Summary

In this chapter a detailed description of the mobile manipulator is given. This description is needed when making decisions about the trajectory generation and optimization methods which are discussed in the next chapter. The forward and inverse kinematics can be used to formulate a solution for the joint trajectories of the prescribed task motions. The dynamic description can be used to evaluate a cost function related to energy consumption as is required. All of the system parameters of the VinciTech mobile manipulator are described in Appendix B.

Chapter 4

Trajectory generation and optimization

In this chapter the trajectory generation and optimization methods for the mobile manipulator are described. For each task a motion is predefined which the end effector has to follow. The first step of the trajectory generation is to define manipulator and MRP configurations for which the end effector is initialized for the task motions. The trajectory is then generated by describing the coordinates as function of time from one configuration to the next. The configurations and trajectories are a function of free variables which can be optimized.

Two optimization algorithms are used. The first algorithm optimizes the MRP configurations for each task and provides an initial collision free MRP path for the second optimization. The second algorithm considers the configurations resulting from the first algorithm and generates time dependent trajectories for both the MRP and the manipulator. The first algorithm is used to reduce the computational effort while finding a feasible path by optimizing for a kinematic cost function. The path of the first algorithm is used as initial solution to generate the time dependent MRP trajectory. The second optimization considers a dynamic cost function which is computationally expensive. The second optimization is performed on segments defined by the MRP motion of one task to the next and multiple manipulator motions related to the initialization and execution of the task motion. By optimizing the segments separately the number of optimization variables is reduced.

4.1 The task motion

The trajectory of the mobile manipulator is generated as function of the constrained task motion. A task motion describes the configuration of the end effector in 6 degrees of freedom. The task motions are a function of the provided task positions, the environment and the workspace of the MRP. The trajectory generator has to describe the mobile manipulator joint coordinates as function of time such that along the trajectory each of the task motions is performed by the end effector. To actuate the system the joint coordinates are required as function of time.

Each task motion describes a straight line motion of the end effector for which the orientation remains constant. The end effector is initialized in front of a task position $r_{\text{task},i}$ at a fixed distance d_{task} . From the initial configuration the end effector moves forward along the z -axis of the end effector until $r_{\text{task},i}$ is reached. From $r_{\text{task},i}$ the motion is reversed until the end effector returns at the initial configuration.

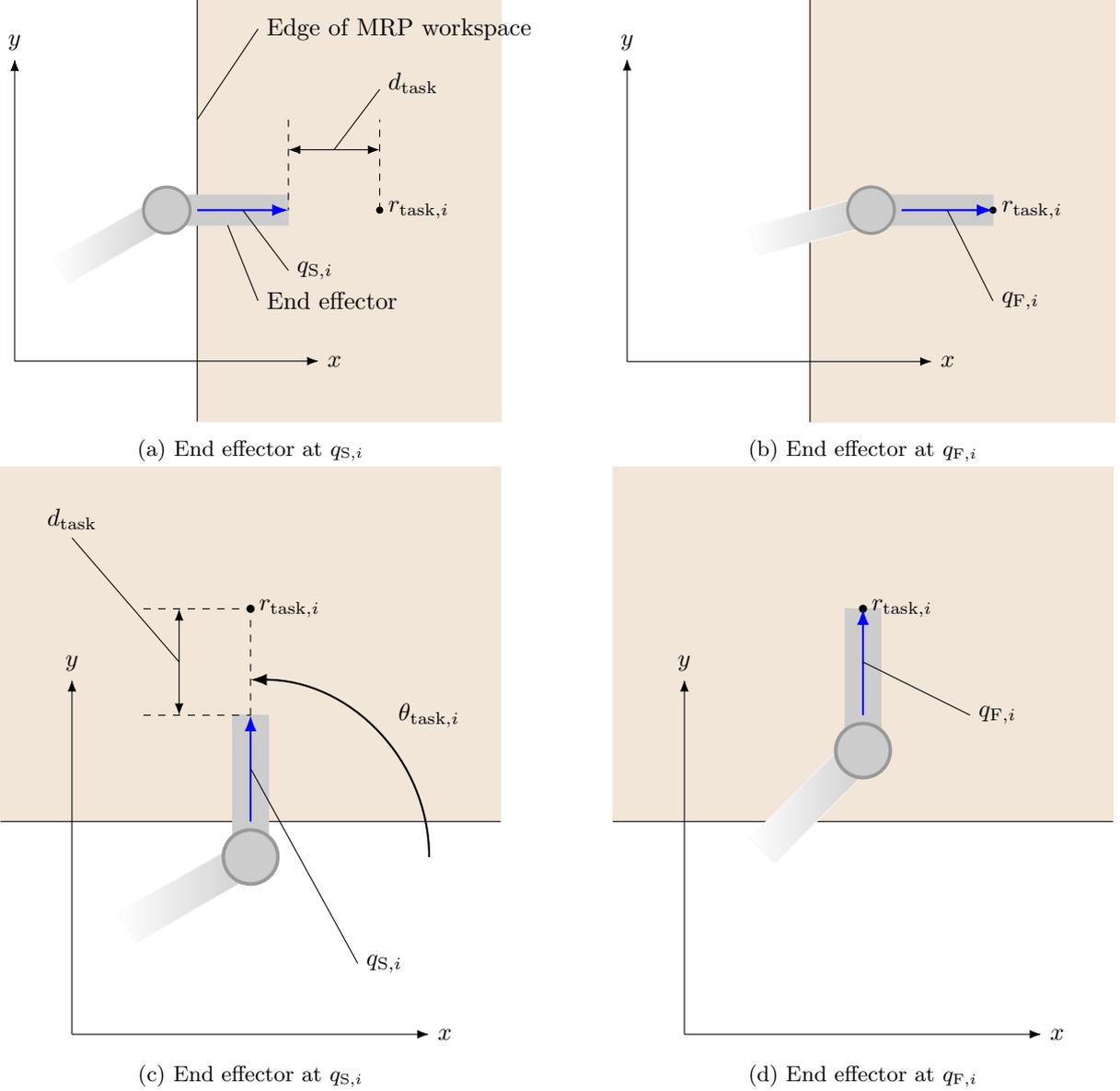


Figure 4.1: Topview of the task motions of the end effector as function of the task position and workspace

Each task motion is described by a straight line between two end effector configurations. The first configuration is the end effector in the initial configuration as in Figure 4.1a. In this configuration the tip of the end effector is at distance d_{task} from the task position. In the second configuration the tip of the end effector is at the task position as in Figure 4.1b. The start configuration $q_{S,i}$ is described by $r_{\text{task},i}$, approaching distance d_{task} and the orientation in Euler angles $\psi_{S,i}$ such that:

$$q_{S,i} = \begin{bmatrix} r_{S,i} \\ \psi_{S,i} \end{bmatrix} \quad \text{where } r_{S,i} = r_{\text{task},i} - R_z(\theta_{\text{task},i}) \begin{bmatrix} d_{\text{task}} \\ 0 \\ 0 \end{bmatrix} \quad \text{and } \psi_{S,i} = \begin{bmatrix} 0 \\ \frac{1}{2}\pi \\ \theta_{\text{task},i} \end{bmatrix}. \quad (4.1)$$

The final configuration is only a function of the task position. The configuration of the end effector remains constant throughout the complete task motion and thus:

$$q_{F,i} = \begin{bmatrix} r_{F,i} \\ \psi_{F,i} \end{bmatrix} = \begin{bmatrix} r_{\text{task},i} \\ \psi_{S,i} \end{bmatrix}. \quad (4.2)$$

The angle θ_{task} is both the orientation of the end effector and direction of motion of the end effector as in Figure 4.1. The end effector moves from $q_{S,i}$ to $q_{F,i}$ and then back to $q_{S,i}$.

4.2 Trajectory generation method

In order to perform a task motion the end effector trajectory has to be translated to actuator trajectories. Furthermore, to be able to start the task motion the end effector has to be initialized to the configurations $q_{S,i}$ as defined in section 4.1. Three options are considered to be able to generate the actuator trajectories. The first option is the Moore-Penrose inverse of the non-square Jacobian in combination with a null space trajectory used to optimize the resulting actuator trajectory. The second option is defining three of the redundant coordinates such that the remaining actuator coordinates can be defined by the inverse of the square Jacobian. The defined coordinates should comply with the physical limitations of the remaining coordinates in relation with the end effector trajectory. The third option is to define the begin and end state of each of the actuator coordinates. The trajectory is then generated by forward kinematics.

4.2.1 The task motion

The task motion of the end effector is constrained in six degrees of freedom. Therefore, the actuator trajectories have to be defined by either of the inverse methods. Because of the non-linear kinematic equations it is difficult to predict the actuator trajectories as function of the Moore-Penrose inverse. Therefore, it is also not straight forward to control the actuator trajectories with the null space. The null space trajectories have to be defined by a function which can be altered by optimization variables. To be able to counteract the non-linear kinematics, the null space trajectories have to consist of a large number of optimization variables. If this is not the case the actuators might exhibit unwanted behavior, for instance accelerations of the MRP. Because the MRP is significantly heavier than the manipulator these accelerations contribute greatly towards the energy consumption. By fixing the MRP position these accelerations are removed and the manipulator motion is fully constrained. The inverse of the square Jacobian can then be used to compute the actuator trajectories. The initial configuration of both the MRP and manipulator define the resulting actuator trajectories. Because of the more intuitive trajectory generation and the possibility to fix the MRP the second option is chosen.

4.2.2 The initialization of the end effector

The inverse kinematics during the task motion should not result in singularities or exceeding of the physical joint limits. Therefore, the mobile manipulator has to be initialized by a configuration for which is guaranteed that the resulting actuator trajectories during the task motion are feasible. During initialization of the end effector for the task motion the manipulator might have to pass through singularities. At a singularity the Jacobian is not full rank which means that the inverse Jacobian cannot be used. Therefore, to initialize the mobile manipulator for the task motion the trajectories are generated with the forward kinematics.

The initial configuration of the mobile manipulator at the task configuration is not yet specified. Moreover, the mobile manipulator is described by nine degrees of freedom while the end effector configuration is described by only six degrees of freedom. Therefore, the mobile manipulator exhibits three degrees of freedom redundancy and a unique solution cannot be computed. Instead, an infinite range of mobile manipulator configurations correspond to the same end effector configuration. To define the possible mobile manipulator configurations two methods are considered.

The first method to find the mobile manipulator configurations is to use an optimization algorithm. The cost function of the algorithm is defined by the error between the required and generated end effector configuration. The optimization problem is not complex and solutions are likely to be found. However, finding various solutions systematically is not possible. During task motions the MRP configuration is fixed which means the remaining degrees of freedom of the manipulator are constrained. It is needed to be able to vary the MRP configuration such that the complete task motion is guaranteed to be feasible.

The second method to find the mobile manipulator configurations is to define three of the redundant mobile manipulator coordinates such that the remaining coordinates can be computed by the non-redundant inverse kinematics, similarly to the constrained task motion. By defining the coordinates of the MRP the inverse solution of section 3.2.2 can be implemented. To be able to use the manipulator inverse kinematics the configuration defined for the MRP must comply with the physical limitations of the manipulator and the required end effector configuration. Because it is possible to parameterize and vary the MRP configuration and thus the infinite inverse configurations of the mobile manipulator, this method is chosen.

4.2.3 Mobile manipulator motion

The motion of the mobile manipulator is described by forward and inverse kinematics. The forward kinematics is used to be able to pass through singularities and the inverse kinematics is used to be able to follow a trajectory with the end effector. The configurations between which the forward kinematic trajectory is constructed is defined by MRP configurations. The manipulator configuration follows from the MRP configuration and the inverse solution of the end effector at the initial position of the task motion. The manipulator trajectory then results from the inverse kinematics after which the mobile manipulator continues to the next task.

4.3 First optimization: MRP path and stopping configurations

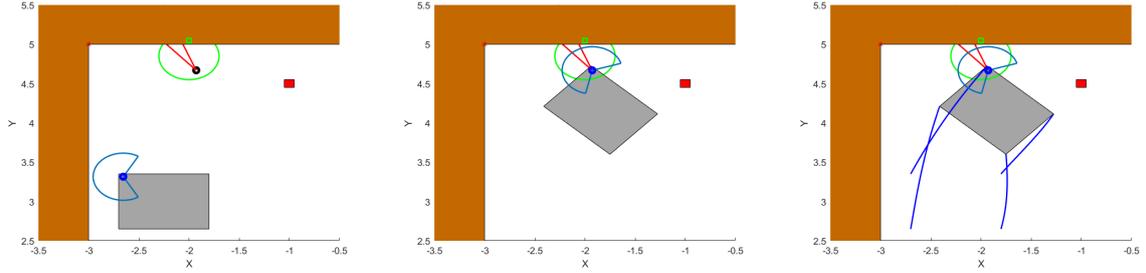
The goal when defining a path for the MRP is that the manipulator is able to execute the task motions while minimizing a cost function. Placing the MRP within a certain range of a task is not sufficient to guarantee that the task motion is executable. Instead the manipulator motion has to be evaluated for singularities, the physical limitations of the joints and collisions. Configurations of the manipulator can be evaluated analytically but to guarantee that the complete task motion is feasible simulations have to be performed. Therefore, a type of evolutionary algorithm (EA) is used, namely the genetic algorithm (GA). The GA allows for a blackbox numerical optimization in which the simulations can take place.

The GA produces optimization variables which are used to perform simulations to compute a cost. Based on the cost of multiple solutions a new iteration of multiple solutions is formulated. To be able to optimize the MRP path by means of a GA the configurations and paths of the MRP are defined as function of optimization variables. A cost function is used to evaluate the solutions. To be able to search for feasible solutions a feasibility check of generated solutions is performed. During this check a penalty is given to the cost if a generated solution results in collisions or infeasible joint trajectories. The algorithm is then able to identify feasible and infeasible solutions. The complete process of path generation and evaluation is explained in detail in this section.

4.3.1 The MRP path as function of optimization variables

The kinematic model of the MRP describes its position in x - and y -direction and the rotation around the z -axis. The coordinates are expressed in frame W . The path is first described as function of s , the variable $s \in [0, 1]$ representing the fraction of the path which is traveled. For each task a path is generated. The complete path is found by connecting the individual paths. The MRP stops at each configuration and thus only continuity of the position is required. During the second optimization the function $s(t)$ is defined to translate the paths to trajectories.

The MRP configuration as function of s is found by first defining the configuration at $s = 0$ and $s = 1$. The configuration is given by the position in x and y direction and the rotation around the z -axis $q_{\text{mrp}} = \begin{bmatrix} x_{\text{mrp}} & y_{\text{mrp}} & \theta_{\text{mrp}} \end{bmatrix}$. The begin and end configurations are then connected by a function for which the MRP does not collide with the environment. The configuration $q_{\text{mrp}}(s)|_{s=1}$ is formulated by describing possible manipulator positions r_m^W as function of the task position. The position r_{mrp}^W is then a function of θ_{mrp} , r_m^W and r_m^{mrp} . The configuration $q_{\text{mrp}}(s)|_{s=0}$ is either the initial configuration or $q_{\text{mrp}}(s)|_{s=1}$ of the previous segment.



(a) The initial configuration of the MRP (b) The MRP at the task configuration (c) A generated path between the configurations

Figure 4.2: An example of a generated path for the MRP

In Figure 4.2 the MRP is shown by the grey rectangle. The manipulator position on the MRP r_m^{MRP} is shown by the blue circle. The partial circle in blue around the manipulator position indicates the range of the first joint of the manipulator. The black circle is the generated position by the algorithm r_m^{W} from which the task can be executed. For this position the manipulator motion is evaluated for the task motion which results in a range of the first joint which has to be executable. This range is indicated by the red partial circle. As can be seen, the configuration of the MRP is chosen such that the range of the first joint during the task motion fits within the joint range. No collisions occur and thus a path can be generated. The path of the wheels is shown in Figure 4.2c.

The generated manipulator base position r_m^{W} is a function of the task position $r_{\text{task}}^{\text{W}}$, an offset and a distance defined by the system parameters and the angle θ_m around the z -axis. With the end effector at the task position as in Figure 4.1b and 4.1d the position of the wrist can be derived. From the wrist joint a maximum distance d_m to r_m^{W} is defined as function of the link lengths and the height of the second joint of the manipulator:

$$d_m = \sqrt{(L_{m,2} + L_{m,3})^2 - (z_{\text{task}}^{\text{W}} - L_{m,1} - r_m^{\text{W}})^2}. \quad (4.3)$$

The manipulator may not collide with the environment and thus a minimum and maximum value for θ_m is defined by θ_m^- and θ_m^+ respectively. The values θ_m^- and θ_m^+ define the intersections of the circle with radius d_m centered at the wrist position and the edge of the MRP workspace as in Figure 4.3.

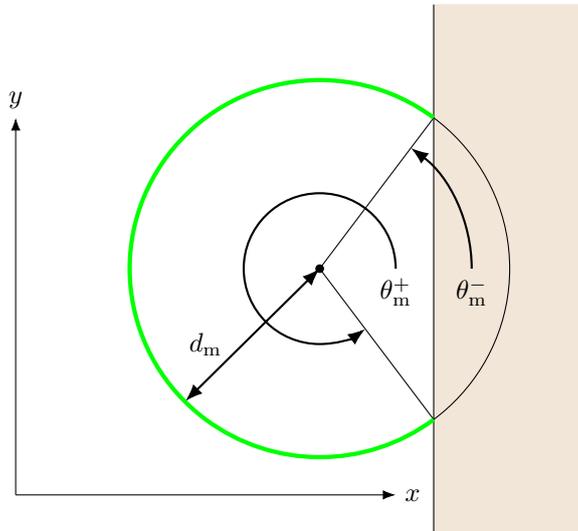


Figure 4.3: Visualization of the partial circle in which the manipulator base can be positioned

The minimum and maximum angle in combination with the distance d_m describe the green partial circle of figures 4.2 and 4.3. The position r_m^{W} is described to be in the partial green circle by the optimization variables o_1 and o_2 as in

$$\begin{aligned}\theta_m &= \theta_m^- + o_1(\theta_m^+ - \theta_m^-) \text{ for } 0 \leq o_1 \leq 1, \\ r_m^W &= r_{\text{task}}^W - \begin{bmatrix} \cos(\theta_{\text{task},i}) \\ \sin(\theta_{\text{task},i}) \end{bmatrix} L_{m,1} + \begin{bmatrix} \cos(\theta_m) \\ \sin(\theta_m) \end{bmatrix} o_2 d_m \text{ for } 0 \leq o_2 \leq 1,\end{aligned}\quad (4.4)$$

where the positions are considered in the x - and y -direction with the fixed z -position as function of the MRP dimensions.

The possible configurations of the MRP are derived using the relative positions of the manipulator and the limits of joint 1. For each manipulator position the joint paths are derived using the inverse kinematics. In order for the manipulator to be able to execute the task from that position the minimum and maximum value of all joint paths have to lie within their respective minimum and maximum value. As before, the superscript $+$ indicates the upper limit of a parameter and $-$ the lower limit. For the orientation of the MRP θ_{mrp} should hold that

$$\begin{aligned}\theta_{\text{mrp}} + \theta_1^- &\leq \min(\theta_1(s)) \leq \theta_{\text{mrp}} + \theta_1^+ \forall \{s \mid 0 \leq s \leq 1\} \\ \theta_{\text{mrp}} + \theta_1^- &\leq \max(\theta_1(s)) \leq \theta_{\text{mrp}} + \theta_1^+ \forall \{s \mid 0 \leq s \leq 1\},\end{aligned}\quad (4.5)$$

As a result the orientation of the MRP is described by relating the limits of joint 1 to the minimum and maximum values of the inverse solution. The limits θ_1^+ and θ_1^- are shown in Figure 4.2 by the blue partial circle centered around r_m . The limits of the orientation of the MRP result from

$$\begin{aligned}\theta_{\text{mrp}}^- &= \max(\theta_1(s)) - \theta_1^+ \text{ with } \{s \mid 0 \leq s \leq 1\}, \\ \theta_{\text{mrp}}^+ &= \min(\theta_1(s)) - \theta_1^- \text{ with } \{s \mid 0 \leq s \leq 1\}.\end{aligned}\quad (4.6)$$

The orientation is defined by varying the value of the third optimization variable o_3 between 0 and 1 as in

$$\theta_{\text{mrp}} = \theta_{\text{mrp}}^- + o_3(\theta_{\text{mrp}}^+ - \theta_{\text{mrp}}^-) \text{ for } 0 \leq o_3 \leq 1. \quad (4.7)$$

The MRP and θ_1 rotate around the same axis, although the centers of rotation do not coincide. Therefore, the position r_{mrp}^W is a function of θ_{mrp} , r_m^{mrp} and r_m^W . The MRP center position is found by

$$r_{\text{mrp}}^W = r_m^W - R_m^W r_m^{\text{mrp}}. \quad (4.8)$$

The MRP configurations at the tasks are now derived as function of the optimization variables and are defined by $q_{\text{mrp},i}$. The next step is to generate a path between each of the configurations. Each path is generated by a polynomial with one free variable. When more tasks are defined then the sequence defined above is repeated and more intermediate configurations are generated. The initial configuration of each path is then the final configuration of the previous path. The polynomials are defined as function of the variable s . An n th order polynomial function of the coordinate q is defined by

$$q(s) = p_1 + p_2 s + p_3 s^2 + \dots + p_n s^{n-1} = \sum_{i=1}^n p_i s^{i-1}. \quad (4.9)$$

The derivative of $q(s)$ results from the chain rule of differentiation.

$$\dot{q}(s) = (p_2 + 2p_3 s + \dots + p_n(n-1)s^{n-2}) \dot{s} = \sum_{i=2}^n p_i(i-1)s^{i-2} \dot{s} \quad (4.10)$$

The polynomial parameters p_i are a function of the begin and end constraints of each of the MRP coordinates. Furthermore, to allow optimization the polynomial has to be a function of one of the

optimization variables. Therefore, a fourth order polynomial is used. The constraints on the coordinate paths are defined at $s(t_0) = 0$ and $s(t_f) = 1$. The begin and end state are defined by $q(s(t_0))$ and $q(s(t_f))$. The velocity is zero at both $s(t_0) = 0$ and $s(t_f) = 1$.

$$\hat{q} = \begin{bmatrix} q(s(t_0)) \\ \dot{q}(s(t_0)) \\ q(s(t_f)) \\ \dot{q}(s(t_f)) \end{bmatrix} = \begin{bmatrix} p_1 + p_2 s_{t_0} + p_3 s_{t_0}^2 + p_4 s_{t_0}^3 + p_5 s_{t_0}^4 \\ (p_2 + 2p_3 s_{t_0} + 3p_4 s_{t_0}^2 + 4p_5 s_{t_0}^3) \dot{s}_{t_0} \\ p_1 + p_2 s_{t_f} + p_3 s_{t_f}^2 + p_4 s_{t_f}^3 + p_5 s_{t_f}^4 \\ (p_2 + 2p_3 s_{t_f} + 3p_4 s_{t_f}^2 + 4p_5 s_{t_f}^3) \dot{s}_{t_f} \end{bmatrix} \quad (4.11)$$

where p_5 takes the value of the corresponding optimization variable related to the respective coordinate.

The goal is to express the values of $p = [p_1 \ p_2 \ p_3 \ p_4]^\top$ as function of the constraints and the selected optimization variable. Since the number of equations and the number of unknown values is equal (4.11) can be rewritten such that

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} = \begin{bmatrix} q(s(t_0)) \\ 0 \\ -3q(s(t_0)) + 3q(s(t_f)) - p_5 \\ 2q(s(t_0)) - 2q(s(t_f)) - 4p_5 \end{bmatrix} \quad (4.12)$$

The path of the MRP is a function of three optimization variables, one for each coordinate of q_{mrp} , o_4 , o_5 and o_6 . The configurations and paths of the MRP are thus defined by six values, the optimization variables. When multiple tasks are defined a multiple of six optimization variables is needed.

4.3.2 MRP configuration and path evaluation

The complete path of the MRP is optimized for a kinematic cost function given by

$$\mathbf{CF} = \sum_{j=1}^4 \sum_{i=2}^{n_{\text{samples}}} \sqrt{(x_{\text{wheel}}(i) - x_{\text{wheel}}(i-1))^2 + (y_{\text{wheel}}(i) - y_{\text{wheel}}(i-1))^2}, \quad (4.13)$$

which is the sum of the path lengths of each MRP wheel with sufficiently large n_{samples} . The four wheels are located at the corners of the MRP. In the kinematic model the MRP is defined by the position in x - and y -direction and the rotation around the z -axis. These three coordinates are not chosen as cost function parameters because they require a weighting factor to relate the distance in meters to the rotation in radians. Furthermore, when traveling from A to B the traveled distance can in general be related to energy consumption [44].

In the optimization algorithm the generated configurations and paths have to be evaluated. This evaluation has to be done in order to guarantee feasibility but also to compute the cost of the solutions in order to arrive at a better solution. Infeasibility can occur when the manipulator is unable to perform the task motion from its generated position or the generated path results in a collision. Furthermore, by evaluating the generated solutions in an efficient order unnecessary computations are avoided. The order of evaluation is based on the feasibility as result of certain optimization variables. First the manipulator position is considered, then the MRP configuration and finally the MRP path. When a generated solution is infeasible a penalty is given and the evaluation of this solution is not continued. This prevents further computation of infeasible solutions.

Infeasibility of the task motion

The first feasibility check of a generated solution is related to r_m in relation to the task motion. The position and initial configuration of the manipulator define the joint path during the task motion. The

joint path is evaluated for feasibility along the task motion. The configurations of the end effector at the start and finish of the task motion, q_S and q_F respectively, are used to define two inverse solutions, θ_S and θ_F . To avoid having to simulate the complete joint path for each of the eight initial configurations, $\theta_S(j)$ and $\theta_F(j)$ with $\{j \in \mathbb{R} : 1 \geq j \geq 8\}$ are evaluated beforehand.

The eight manipulator configurations corresponding to both q_S and q_F are first evaluated on exceeding the joint limits. If none of the eight solutions lie within the joint limits then the solution is infeasible and further evaluation is not required. For feasible configurations is checked if there exist pairs of θ_S and θ_F for which the manipulator does not have to pass through a singularity to get from θ_S to θ_F . This check is necessary because during the task motion the inverse kinematics are required. To be able to compute the inverse of the analytic Jacobian at all times, the manipulator is not allowed to pass through singularities. When a feasible pair is found the final check is to simulate the inverse kinematics to verify that the joint limits are not exceeded along the path or the solution is infeasible.

To determine if the manipulator has to pass through a singularity when moving from θ_S to θ_F the resulting angles of joints 1, 3 and 5 are evaluated. For the evaluation of θ_1 the angle from r_m^W to r_{task}^W is needed to detect a base flip.

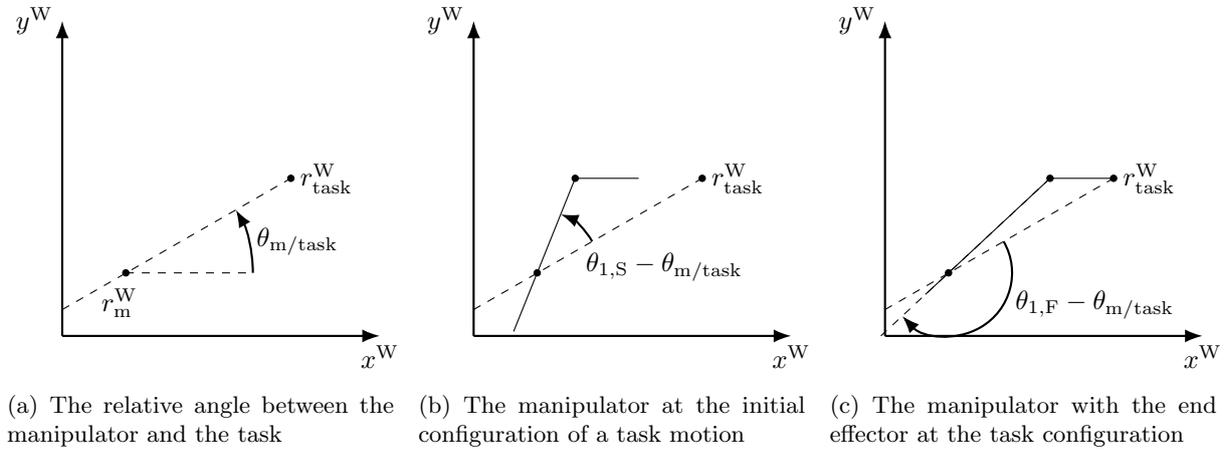


Figure 4.4: Example of base flip

With the positions of the task and the manipulator base the angle $\theta_{m/task}$ is derived as follows

$$\theta_{m/task} = \text{atan2}(y_{task}^W - y_m^W, x_{task}^W - x_m^W). \quad (4.14)$$

The angles θ_1 of both inverse solutions are on the same side of a singular configuration for pairs of $\{i \in \mathbb{R} : 1 \geq j \geq 8\}$ and $\{j \in \mathbb{R} : 1 \geq j \geq 8\}$ if the following relation holds:

$$\text{sign}(\theta_{1,S}(i) - \theta_{m/task}) = \text{sign}(\theta_{1,F}(j) - \theta_{m/task}). \quad (4.15)$$

When a base flip occurs the signs are not equal which corresponds to the situation visualized in figures 4.4b and 4.4c. For the angles of joints 3 and 5 the relation to frame W is irrelevant and only the sign is considered. The solutions are on the same side of a singular configuration for a pair of $\{i \in \mathbb{R} : 1 \geq j \geq 8\}$ and $\{j \in \mathbb{R} : 1 \geq j \geq 8\}$ if the following relation holds:

$$\begin{aligned} \text{sign}(\theta_{3,S}(i)) &= \text{sign}(\theta_{3,F}(i)) \\ \text{sign}(\theta_{5,S}(i)) &= \text{sign}(\theta_{5,F}(i)). \end{aligned} \quad (4.16)$$

The signs of the three joints are compared for all possible combinations of the values of i and j . The pairs for which the requirements are met are saved and the joint paths are then generated with the inverse Jacobian. The complete joint path is then evaluated on collisions and exceedance of the joint limits.

Visualization of manipulator position feasibility

The feasibility of the manipulator position r_m is defined by the joint path and if the manipulator collides with the environment. The joint path should not exceed the limits or cross a singularity. In Figure 4.4 the feasibility of the task motions is visualized by the base positions of the manipulator relative to the task and environment. From the green areas the manipulator is able to perform the task motions, from the red area the manipulator either crosses a singularity, collides with the environment or is required to actuate beyond the limits of the joints.

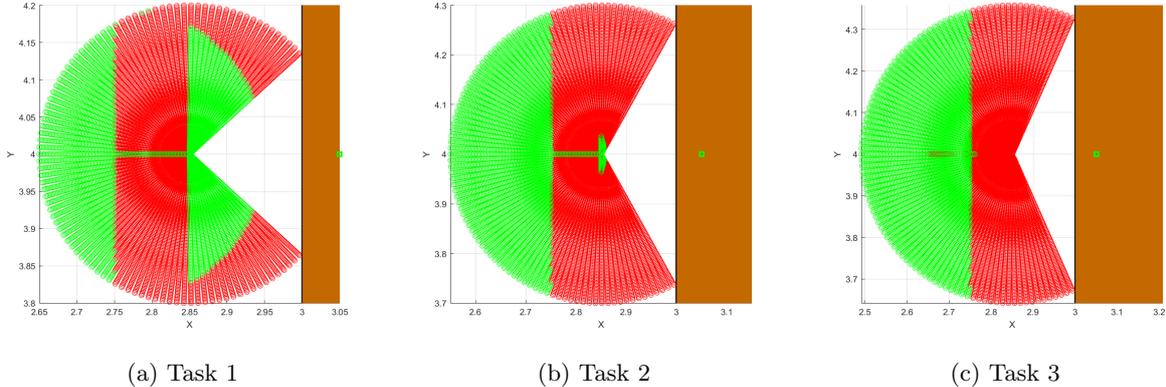


Figure 4.5: Feasibility visualization of the manipulator positions of the example situations, green areas define feasible manipulator positions and red areas define infeasible manipulator positions

The task locations, shown by the green squares, are 0.05 [m] outside of the edge of the workspace at a height of 0.80 [m], 0.70 [m] and 0.60 [m] respective to task 1, 2 and 3. What can be seen from Figure 4.5 is that the feasibility check is not unnecessary and that the regions of feasibility vary as function of the task position.

MRP infeasibility

The second feasibility check is related to collision detection. First the generated MRP configurations are considered at each of the tasks. The MRP can be orientated around the manipulator position as function of the optimization variables, where the allowed orientation of the MRP is defined by the joint limits of θ_1 and its path resulting from the first feasibility check at that r_m^W . For certain MRP configurations it might be possible that the MRP overlaps with an obstacle. To check if the MRP overlaps with an obstacle the body is sampled into a point cloud. A collision is detected if any of the points lie within the dimensions of an obstacle. A collision deems the generated solution infeasible. For the third and final check the collision detection is performed along the complete path of the MRP for both the MRP and the manipulator. This concludes the complete feasibility evaluation of a generated solution for the MRP path planning.

Penalty of an infeasible solution

The feasibility checks described above are used to evaluate the generated solutions in the optimization algorithm. This algorithm is based on a black box formulation where the generated solutions are related to a cost. By relating the solutions to their costs the next solution is generated. The feasibility checks are not related to a physical parameter or cost; the solution is either physically executable or not. However, in order for the optimization algorithm to generate a better solution the costs have to vary as function of the performance of the solution. Therefore, for each of the feasibility checks a different penalty is considered in order to make a distinction between the solutions.

Three penalties are given with decreasing magnitude. This is done to make a distinction between the three feasibility checks. The first, and thus highest, penalty on the cost is for the manipulator base position. When the manipulator is unable to perform the task from the generated position then none of the optimization variables are part of a valid solution. This is the worst possibility for a solution. The

second penalty is given for the configuration of the MRP. For a feasible manipulator base position multiple MRP configurations are possible, characterized by o_3 . If the generated MRP configuration collides with an object then the path can never be feasible. The variables o_1 and o_2 are valid which is a step closer to a feasible path and thus a lower penalty is given. The third and final penalty is given when for a feasible MRP configuration an infeasible path is generated. This penalty is the lowest such that the variables o_1 to o_3 are more likely to be selected in the next iteration. The penalties are given per task. This means that the cost increases for multiple infeasible positions, configurations or paths. Moreover, it is possible to separate the evaluation of the separate paths. By doing so the valid parts of a solution are not discarded.

The penalties are significantly larger than the cost resulting from the cost function. Therefore, a feasible solution is always selected over a penalized solution. When the infeasibility is detected earlier in the sequence of the solution evaluation, unnecessary computations can be avoided. The penalties result in a scoring related to the quality of the solution. Therefore, the optimization algorithm is able to generate solutions as function of previous solutions.

4.3.3 The first optimization algorithm

The first optimization algorithm serves two purposes. Namely, finding a feasible MRP path and improving the path by evaluating a cost function. A feasible solution is sought by evaluating the cost as function of the penalties. When a penalty is given the kinematic cost function is not considered because this would always push the solutions in a certain direction. When a feasible solution is found the cost is defined by the cost function. This result in:

$$\begin{aligned}
 d_{\text{mrp}} &= \sum_{i=2}^{n_{\text{samples}}} \sqrt{(r_{\text{mrp}}(i) - r_{\text{mrp}}(i-1))^2} \\
 d_{\text{wheels}} &= \sum_{j=1}^4 \sum_{i=2}^{n_{\text{samples}}} \sqrt{(r_{\text{wheel},j}(i) - r_{\text{wheel},j}(i-1))^2} \\
 \mathbf{CF} &= \begin{cases} d_{\text{mrp}} + d_{\text{wheels}}, & \text{if } \mathbf{penalty} = 0 \\ \mathbf{penalty} & \text{if } \mathbf{penalty} \neq 0 \end{cases}
 \end{aligned} \tag{4.17}$$

Because of the penalties the cost is not defined by a right/wrong measure and thus the algorithm can iteratively reduce the cost.

The evolutionary algorithm which is used to find the near optimal MRP path is closest related to a genetic algorithm. A genetic algorithm generates the next iteration of solutions based on the previous solutions and their costs. For each solution one or more parent solutions are selected from the previous generation. The selected parent solutions are a number of the solutions resulting in the lowest cost. A new generation is formulated by:

1. A number of the unaltered best solutions of the previous iteration
2. A small random variation $c_{i,\text{var}}$ on a number of the previous best solutions
3. Combinations of the best solutions
4. Mutations of the best solutions with a chance of randomizing individual optimization variables

The optimization variables are defined by a minimum and maximum value o_i^- and o_i^+ respectively. This range of values is sampled by n_{o_i} of samples. The value of the optimization variable is then defined by an index selected by the optimization algorithm i_{o_i} such that:

$$o_i = o_i^- + \frac{i_{o_i} - 1}{n_{o_i} - 1} (o_i^+ - o_i^-) \text{ with } \{i_{o_i} \in \mathbb{R} : 1 \geq i_{o_i} \geq n_{o_i}\}. \tag{4.18}$$

The optimization algorithm stopping criterion is defined by the penalties and the number of consecutive iterations for which the cost is unaltered. The algorithm stops when the cost is unaltered for

40 iterations provided that the cost is not a function of a penalty, meaning the solution is feasible. In Section 4.3.2 it was discussed that it is possible that a feasible solution does not exist. In that case the algorithm has to stop searching after n_{\max} iterations. Without this criterion the algorithm never stops searching.

The small random variation $c_{i,\text{var}}$ is decreased as function of the number of iterations for which the cost is unaltered, n_{same} , and the stopping criterion, n_{stop} . The i th optimization index of iteration k is then:

$$\begin{aligned} i_{o_i}(k) &= \mathbf{Min} \left(\mathbf{Max} \left(i_{o_i}(k-1) + i_{o_i,\text{var}}, 1 \right), n_{o_i} \right), \\ i_{o_i,\text{var}} &= \left\{ i_{o_i,\text{var}} \in \mathbb{R} : -\lceil c_{i,\text{var}} \left(1 - \frac{n_{\text{same}} - 1}{n_{\text{stop}}} \right) \rceil \geq i_{o_i,\text{var}} \geq \lceil c_{i,\text{var}} \left(1 - \frac{n_{\text{same}} - 1}{n_{\text{stop}}} \right) \rceil \right\}, \end{aligned} \quad (4.19)$$

where $i_{o_i}(k-1)$ is a solution of the previous iteration.

To increase the efficiency of the search algorithm multiple problem specific solutions are generated. The goal of the problem specific solutions is to evaluate the individual segments of the MRP path of multiple solutions. It is unlikely that a single solution generates the lowest cost for each of the individual segments. Therefore, the optimization variables, of multiple solutions, which result in the lowest cost of each of the individual segments are combined. The initial MRP configuration of each segment is the final configuration of the previous segment and thus the resulting solution is not a perfect combination of the individual segments. However, since the range of MRP configurations is confined it is likely that with the same path variables a collision free path is found with different initial configurations.

Multiple problem specific solutions are used in order to combine potential best solutions of multiple segments. First of all a solution of the next generation is defined by combining the best solutions of all segments. Both the configurations and paths are combined in one solution. This solution is logically chosen to combine all best segments in one best total solution.

The second problem specific solution combines the best solutions, but only for the optimization variables defining the configuration of the MRP at the task position. The remaining optimization variables, those defining the MRP path, are randomized. In this case feasible configurations are combined and simultaneously feasible paths are randomly sought. In other solutions in which a solution is chosen with a feasible configuration but unfeasible path, the optimization variables defining the path are also considered for the new solution. In that case new paths are not considered.

Other problem specific solutions are generated where combinations are made between configurations and paths based on the total cost or costs of individual segments. The solutions are altered by randomly varying the optimization variables by a fraction of its total range. This is meant to search around the current best solution to iteratively reduce the cost. In Appendix B the parameters of the optimization algorithm are given.

4.4 Second optimization: Trajectory

The MRP and manipulator configurations are specified by the optimized solution of the first algorithm. When a solution is provided by the first algorithm then for each of the configurations it is guaranteed that the manipulator is able to perform the fully constrained task motion. This is done by evaluating the initial joint configuration and the joint path as function of the inverse kinematics. Furthermore, an initial MRP path is provided which connects the feasible MRP configurations at each task. The provided MRP path is free of collisions and optimized for shortest distance traveled by each wheel.

In the second optimization the complete path of both the MRP and the manipulator is generated as function of the configurations defined by the first optimization. The paths are transformed into trajectories by specifying a time dependent function $s(t)$. The time dependent trajectories allow the computation of forces and torques which are necessary to evaluate the dynamic cost function related to energy consumption.

4.4.1 Approach

In the second optimization algorithm trajectories are generated as function of optimization parameters. The trajectories consist of two components, namely a path and the time dependent function $s(t)$. The configurations are not a function of the optimization variables, instead they result from the first optimization algorithm.

The complete trajectory of the mobile manipulator is divided into five segments. The first segment corresponds to the MRP motion and is thus defined by the separation of the manipulator and the MRP. The second segment is the initialization of the end effector to configuration $q_{S,i}$. The third and fourth segment correspond to the task motion where the end effector moves from $q_{S,i}$ to $q_{F,i}$ and then back to $q_{S,i}$. Finally, the fifth segment is defined by returning to manipulator to θ_{rest} .

The segments are not subject to the same constraints. Therefore, not all segments are defined by the same amount of optimization variables. Each segment serves a different purpose and thus the joint trajectories are either formulated using the forward or inverse kinematics. The forward kinematics is used to initialize the manipulator in the configuration $\theta_{S,i}$ and the inverse kinematics is then used for the constrained task motion.

The trajectory generation is formulated by moving from one configuration to the next. The next configuration is either defined for the MRP or for the manipulator. At each configuration the system is at standstill. This allows the optimization to be separated for each segment. As a result, the number of optimization variables per segment optimization is drastically reduced. This is beneficial for the computational efficiency of the complete optimization as the total number of solutions scales exponentially per variable.

4.4.2 Manipulator path generation

The segments of the manipulator trajectory are defined by the requirements of the end effector motion. The end effector is either completely free in how to move from one configuration to the next or a contour-following motion is specified, the task motion. The joint is defined by either forward or inverse kinematics.

The first motion of the manipulator corresponds to the initialization of the end effector for the task motion. The two configurations between which the manipulator moves are the rest configuration during MRP motions θ_{rest} and the inverse solutions $\theta_{S,i}$ as defined earlier. The configuration θ_{rest} is defined in Appendix A. The manipulator either moves from θ_{rest} to $\theta_{S,i}$ during initialization or vice versa when returning to the rest configuration.

In section 4.3.1 the MRP path is defined by begin conditions and end conditions. The manipulator path is generated in the same way such that each coordinate is a function of a fourth order polynomial. Each coordinate is a function of four constraints and one optimization variable. Therefore, for the manipulator six optimization variables are available to vary the path during optimization.

$$q_m(s) = p_1 + p_2s + p_3s^2 + p_4s^3 + p_5s^4 \text{ with } \{s | 0 \leq s \leq 1\} \quad (4.20)$$

The remaining two segments of the manipulator motion are defined by the task motions of the end effector. The first task segment moves the end effector from $q_{S,i}$ to $q_{F,i}$ and the second task segment returns the end effector to $q_{S,i}$. The task motion is fully constrained and the manipulator is non-redundant. This means that the Jacobian is square. During the optimization of the MRP was guaranteed that the manipulator does not attain singular configurations during the task motions as long as the manipulator is initialized by $\theta_{S,i}$. Therefore, the Jacobian is invertible along the complete task motion and the joint paths can be derived by the inverse Jacobian:

$$q_m(s) = \int_0^s J_a^{-1} \dot{q}_{ee}(s, \dot{s}) ds \text{ with } \{s | 0 \leq s \leq 1\}. \quad (4.21)$$

4.4.3 From path to trajectory

The time independent paths of separate coordinates are generated as functions of constraints and optimization variables. Each coordinate of a segment is a function of the same function $s(t)$. Therefore, the functions of the paths are fixed in relation to each other. The function $s(t)$ transforms the paths to time dependent trajectories. A function for $s(t)$ is needed for each of the segments.

For each of the segments the function $s(t)$ should be 0 at the begin time of the segment and 1 at the end time of the segment. Furthermore, $\dot{s}(t) \geq 0 \forall t \in [t_0, t_f]$ such that the same portion of a segment is not traveled more than once. A polynomial is used to express s as function of time. This allows for an analytical description of the function parameters to guarantee that $\dot{s}(t) \geq 0 \forall t \in [t_0, t_f]$. The function of $s(t)$ and its constraints are given by:

$$\begin{aligned} s(t) &= b_1 + b_2 t + b_3 t^2 \\ s(t_0) &= b_1 + b_2 t_0 + b_3 t_0^2 = 0 \\ s(t_f) &= b_1 + b_2 t_f + b_3 t_f^2 = 1 \\ \dot{s}(t) &= b_2 + 2b_3 t \geq 0. \end{aligned} \tag{4.22}$$

The parameter b_3 is defined by an optimization variable. The parameters b_1 and b_2 are a function of b_3 and the constraints. The begin time $t_0 = 0$ which results in

$$\begin{aligned} b_1 &= 0 \\ b_2 &= t_f^{-1}(1 - b_3 t_f^2). \end{aligned} \tag{4.23}$$

Rewriting \dot{s} results in the following relation:

$$\begin{aligned} t_f^{-1}(1 - b_3 t_f^2) + 2b_3 t &\geq 0 \\ b_3(2t_f t - t_f^2) &\geq -1. \end{aligned} \tag{4.24}$$

Finally the range of b_3 is defined by the duration of the segment:

$$-t_f^{-2} \leq b_3 \leq t_f^{-2}. \tag{4.25}$$

For each segment it is possible to transform the path into a trajectory as long as the duration is known. The duration of each segment is predefined and thus not part of the optimization problem.

The duration of a segment is defined by a path length and a maximum acceleration magnitude. A third order polynomial trajectory is used to define the value of t_f , which corresponds to a trajectory description without optimization variables. First of all the position, velocity and acceleration are defined by

$$\begin{aligned} q(t) &= p_1 + p_2 t + p_3 t^2 + p_4 t^3, \\ \dot{q}(t) &= p_2 + 2p_3 t + 3p_4 t^2, \\ \ddot{q}(t) &= 2p_3 + 6p_4 t. \end{aligned} \tag{4.26}$$

By defining constraints on the initial and final state as positions and zero velocities the relation $p_3 = -\frac{3}{2}p_4$ is found. This results in

$$\begin{aligned} q(t) &= -\frac{3}{2}p_4 t^2 + p_4 t^3, \\ \dot{q}(t) &= -3t + 3p_4 t^2, \\ \ddot{q}(t) &= -3p_4 + 6p_4 t. \end{aligned} \tag{4.27}$$

The acceleration thus varies linearly from $-3p_4$ to $3p_4$ as function of t such that the velocity is zero at t_f . This means that

$$\ddot{q}(t) = -\ddot{q}^+ + \frac{2}{t_f}\ddot{q}^+t. \quad (4.28)$$

By defining a maximum magnitude for the acceleration, which is thus $3p_4$, the duration can be expressed as function of the traveled distance and the maximum acceleration:

$$\begin{aligned} q(t_f) &= -\frac{1}{2}\ddot{q}^+t_f^2 + \frac{1}{3t_f}\ddot{q}^+t_f^3 = \frac{1}{6}\ddot{q}^+t_f^2, \\ t_f &= \sqrt{\frac{6q(t_f)}{\ddot{q}^+}}. \end{aligned} \quad (4.29)$$

The distance $q(t_f)$ is defined by the distance between begin and end state of the MRP

$$q(t_f) = \sqrt{(r_{\text{mrp}}(t_0) - r_{\text{mrp}}(t_f))^2}, \quad (4.30)$$

or for manipulator segments the maximum joint rotation scaled by the manipulator dimensions

$$q(t_f) = \frac{L_{m,1} + L_{m,2} + L_{m,3} + L_{m,4}}{\pi} \mathbf{max} \left(\sqrt{(q_m(t_0) - q_m(t_f))^2} \right). \quad (4.31)$$

The resulting duration does not guarantee that the acceleration of the trajectory does not exceed \ddot{q}^+ . However, the duration does result in feasible trajectories by defining finite limits to the duration and acceleration. In Appendix A an evaluation of the system parameters is performed which provides the limits of operation such that the MRP does not topple. With the chosen value of \ddot{q}^+ these limits are not exceeded. Time optimization is not within the scope of this project which means pushing the system to operate at its limits is not the case.

4.4.4 Solution evaluation

In the second optimization algorithm the MRP and manipulator positions are generated as function of time to be able to evaluate a dynamic cost function. This dynamic cost function is defined by the energy consumption E . Furthermore, parameters are considered which relate to the feasibility, as was done in the first optimization algorithm where penalties are given as function of evaluation of the feasibility of a solution.

The energy consumption of the mobile manipulator is a function of the power of the actuators over time. The power can be divided into power resulting from forces and torques, P_F and P_τ respectively. The total energy consumption is then:

$$E = \int_{t_0}^{t_f} P_\tau + P_F dt. \quad (4.32)$$

To compute the power the relevant forces and torques are needed. The linear and rotational power consumption of the MRP and the manipulator are derived differently. The required power to actuate the MRP is a function of both the rotation and translation. The power consumption of the manipulator is only related to the rotation of each joint and their corresponding torques.

MRP

During the time optimization of the MRP trajectory the path is again considered as function of the optimization variables. The found path of the first algorithm is not a function of the same cost function and is not necessarily the best solution for the dynamic cost function. Moreover, the path is transformed to a trajectory. The forces required to move the MRP are necessary to compute the energy consumption. The forces are a function of the trajectory and are derived as follows:

$$F_{\text{mrp}}^{\text{W}} = m_{\text{mrp}}\ddot{r}_{\text{mrp}} + F_{\text{m}}^{\text{W}} + \sum_{i=1}^4 F_{\text{wheel},i}^{\text{W}}. \quad (4.33)$$

The force related to the acceleration of the MRP is derived using Newton's second law of motion. Similarly the reaction force F_{m}^{W} is derived which results from the acceleration of the manipulator in the world frame. Finally the wheels exert a force on the MRP as result of friction. The friction forces act in the opposite direction of the wheel velocity as in

$$\begin{aligned} \dot{r}_{\text{wheel},i}^{\text{W}} &= \dot{r}_{\text{mrp}} - S(\dot{\theta}_{\text{mrp}})R_{\text{mrp}}^{\text{W}}r_{\text{wheel},i}^{\text{mrp}}, \\ F_{\text{wheel},i}^{\text{W}} &= -\dot{r}_{\text{wheel},i}^{\text{W}}k_{\text{fric}}. \end{aligned} \quad (4.34)$$

The matrix multiplication $S(\dot{\theta}_{\text{mrp}})R_{\text{mrp}}^{\text{W}}$ defines the derivative of the rotation matrix [38, .p 124-128]. The value of k_{fric} is chosen to be zero since the value is unknown. Furthermore, by choosing the value to be zero the solutions are generalized and not specific for this system.

The power to translate the MRP linearly results from the dot product of the force described in (4.33) and the velocity. The absolute value is used since the system is unable to regenerate energy during deceleration and thus

$$P_{F_{\text{mrp}}} = |F_{\text{mrp}}^{\text{W}} \cdot \dot{r}_{\text{mrp}}|. \quad (4.35)$$

In terms of rotation the same relations can be found. The torque required to rotate the MRP around its center is given by

$$\tau_{\text{mrp}}^{\text{W}} = R_{\text{mrp}}^{\text{W}}I_{\text{mrp}}R_{\text{W}}^{\text{mrp}} \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_{\text{mrp}} \end{bmatrix} + \tau_{\text{m}}^{\text{W}} + r_{\text{m}}^{\text{mrp}} \times F_{\text{mrp}}^{\text{W}} + \sum_{i=1}^4 r_{\text{wheel},i}^{\text{W}} \times F_{\text{wheel},i}^{\text{W}}, \quad (4.36)$$

where the terms related to angular acceleration, reaction moment of the manipulator and reaction moments of the wheels are present. The power as function of the torque results from the multiplication with the angular velocity, also taken as absolute value:

$$P_{\tau_{\text{mrp}}} = |\tau_{\text{mrp}} \cdot \dot{\theta}_{\text{mrp}}|. \quad (4.37)$$

The energy consumption of both the translation and rotation for a trajectory is found by integrating the sum of the powers over time:

$$E_{\text{mrp}} = \int_{t_0}^{t_f} P_{\tau_{\text{mrp}}} + P_{F_{\text{mrp}}} dt. \quad (4.38)$$

The cost is a function of the energy consumption or the penalty:

$$\mathbf{CF} = \begin{cases} E_{\text{mrp}} & \text{if } \mathbf{penalty} = 0, \\ \mathbf{penalty} & \text{if } \mathbf{penalty} \neq 0. \end{cases} \quad (4.39)$$

Manipulator

The optimization of the manipulator joint trajectories is split into two situations. During the inverse kinematic trajectory generation the path is fixed. In the first algorithm collisions and joint limits were evaluated and guaranteed to meet the requirements. Therefore, feasibility checks do not have to be performed. For the forward kinematic solutions the path is variable and thus collision and the physical joint limits have to be evaluated. Similarly, penalties are given such that the optimization algorithm is

able to distinguish solutions. When a joint path exceeds its limit a penalty is given relative to the sum of the exceedance

$$\begin{aligned} \text{penalty}_i &= \begin{cases} \int_{t_0}^{t_f} \left(\frac{\pi}{18} - (q_{m,i}^+ - q_{m,i}(t)) \right)^2 & \text{if } q_{m,i}^+ < q_{m,i}(t), \\ \int_{t_0}^{t_f} \left(\frac{\pi}{18} - (-q_{m,i}^- + q_{m,i}(t)) \right)^2 & \text{if } q_{m,i}^- > q_{m,i}(t), \end{cases} \\ \text{penalty} &= 1000 \sum_{i=1}^6 \text{penalty}_i. \end{aligned} \quad (4.40)$$

Therefore, it is easier for the algorithm to identify a better solution as there is a larger variety in the costs.

In section 3.2.3 the torques τ_i are derived for each of the actuators of the manipulator. Each actuator delivers its torque at one of the joints. The power consumption is derived by the multiplication of each actuator torque and its velocity:

$$P_{\tau_m} = \sum_{i=1}^6 |\tau_i \dot{\theta}_i|. \quad (4.41)$$

$$E_m = \int_{t_0}^{t_f} P_{\tau_m} dt. \quad (4.42)$$

The cost is a function of the energy consumption or the penalty:

$$\mathbf{CF} = \begin{cases} E_m & \text{if } \text{penalty} = 0, \\ \text{penalty} & \text{if } \text{penalty} \neq 0. \end{cases} \quad (4.43)$$

4.4.5 The second optimization algorithm

The second optimization algorithm optimizes the paths for a dynamic cost function. The motion of the complete system is divided into segments and thus only the trajectory between two fixed configurations is optimized at once. The number of optimization variables is therefore greatly reduced which is beneficial for the efficiency of the optimization. It is not required to combine multiple segments of multiple solutions in order to accelerate the search algorithm, as is done in the first algorithm. The evolutionary algorithm therefore consists of only the standard operators mentioned in section 4.3.3. In Appendix B the parameters of the optimization algorithm are given.

4.5 Summary

This Chapter provides the approach of the trajectory generation and optimization. A separation is made between the mobile robot platform and the manipulator in terms of actuation and optimization. This result in two optimizations. The first optimization is used to optimize the MRP path such that a consideration is made for the complete task sequence. The second optimization is performed for each individual segment which results from the separate actuation of the two systems with energy consumption as the cost. Genetic algorithms are implemented because of the penalties which are used to define infeasible solutions. With the provided trajectory generation and optimization method a comparison can be made between the benchmark trajectory and optimized trajectories.

Chapter 5

Optimized paths and trajectories

In chapters 3 and 4 the mobile manipulator is described and a new trajectory generation method is derived. The new trajectory generation method provides a solution to the trajectory generation problem by utilizing an optimization algorithm. The effectiveness of the optimization algorithm and the optimization strategy is tested by evaluating the results of simulations. First of all, the current trajectory generation method is used to generate a benchmark solution. This trajectory generation method follows a fixed pattern with fixed parameters to generate the trajectory. The resulting energy consumption is compared to the energy consumption of an optimized trajectory. Based on the results of the comparison between the benchmark and optimized trajectory a second comparison is made. The second comparison utilizes varying initial conditions for the optimization in order to demonstrate the effectiveness of the optimization algorithm.

5.1 The benchmark trajectory generation method

The benchmark trajectory is based on the trajectory generation method implemented by VinciTech. Trajectories are generated by connecting waypoints. These waypoints are located at fixed positions in relation to the task positions. The path generation between waypoints is formulated as in Section 4.3.1 by (4.12) with $p_5 = 0$. The function of $s(t)$ is linear from begin time to end time, $s(t) = t_0 + \frac{t}{t_f - t_0}$. Therefore, the complete trajectory is a function of only the fixed parameters of the mobile manipulator, the tasks and the environment. The assumption is made that the fixed parameters always result in a feasible trajectory.

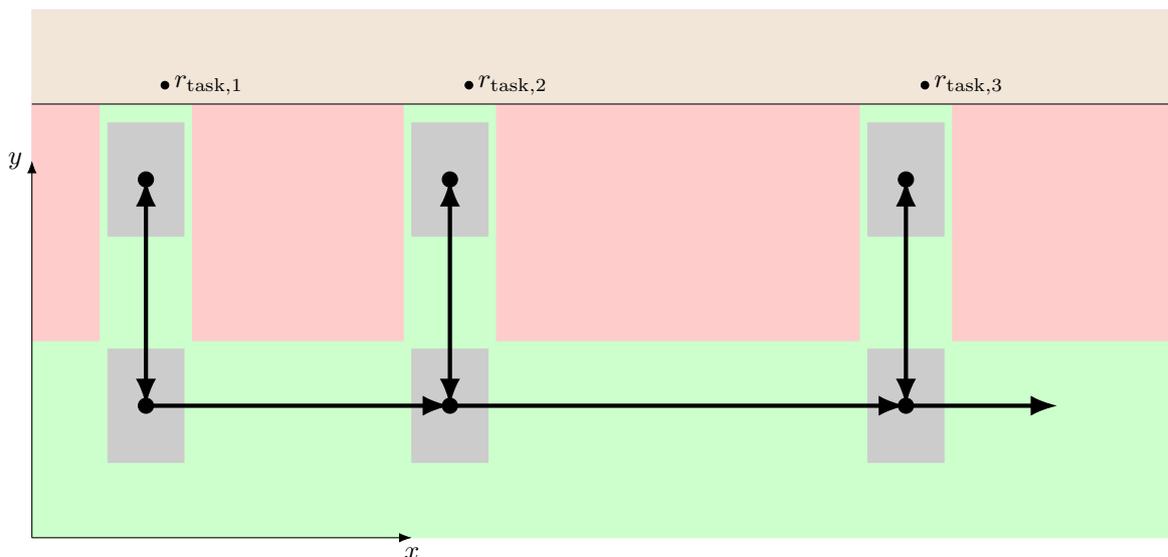


Figure 5.1: MRP workspace for fixed trajectory generation

The workspace of the MRP is defined by areas which are guaranteed to be free of obstacles and

areas in which it is possible for obstacles to be present. Since the trajectory generator is based on fixed parameters only the area free of obstacles is used. In Figure 5.1 the area free of obstacles is shown by the green surface and the areas in which obstacles can be present are shown by the red surfaces. At each task location the MRP is able to approach the task in a straight line. To move from one task to the next the MRP has to return to a safe distance from the edge of the workspace. The waypoints of the MRP are shown by the black dots. The MRP position in front of a task aligns the manipulator with the task in x -direction. The remaining waypoints are generated by moving to a fixed distance from the edge of the workspace in y -direction such that the MRP can move to the next task in the safe area. The waypoints of the manipulator result from the task motion of the end effector and the inverse kinematics.

5.2 Workspace and tasks

The first comparison to make is between the fixed trajectory and the optimized solution generated by the optimization algorithm. The goal of this study is to determine if the optimization algorithm is able to generate a better solution than the trajectory generation method which is used at VinciTech. Therefore, this first comparison concludes if an improvement is accomplished. The comparison is formulated by simulating the fixed trajectory and the optimized trajectory. Four workspace configurations are defined in which the mobile manipulator has to perform a number of tasks.

5.2.1 The considered cases

To verify the trajectory generation and optimization methods four situations are considered. Two of the configurations have tasks on a single side of the workspace and two have tasks on varying sides of the workspace. Both configurations are considered with and without obstacles present. These configurations are considered such that the effect of tasks in varying directions can be seen. Furthermore, the presence of obstacles can illustrate how the new method compares to the fixed method when avoiding obstacles. The configurations result in six cases as follows:

1. Three tasks on one side of the workspace, benchmark
2. Three tasks on one side of the workspace, no obstacles, optimized
3. Three tasks on one side of the workspace, obstacles, optimized
4. Four tasks on two side of the workspace, benchmark
5. Four tasks on two side of the workspace, no obstacles, optimized
6. Four tasks on two side of the workspace, obstacles, optimized

The first three cases are similar to the situation depicted in Figure 5.1. Three tasks are considered such that the trajectories before and after the middle task are affected by obstacles. Cases 4 to 6 consist of four tasks such that two tasks are placed on two sides of the workspace. The trajectory of the MRP moving from one side of the workspace to the other is in that case also influenced by the tasks before and after. Adding more tasks does not demonstrate different behavior or functionality of the applied methods. The trajectory of the benchmark is not affected by the obstacles which results in two benchmark cases.

Parameters

The tasks of each of the cases are defined for a square workspace of which two sides are used. The edges of the workspace are at -3 [m] and 3 [m] in x -direction and -1 [m] and 5 [m] in y -direction. At each edge of the workspace a wall is placed with height 0.5 [m]. Outside and above this wall the tasks are at the coordinates given in tables 5.1 and 5.2.

Task	x [m]	y [m]	z [m]	θ_{task} [rad]
1	-2	5.05	.7	$\frac{1}{2}\pi$
2	0	5.15	.7	$\frac{1}{2}\pi$
3	2	5.05	.79	$\frac{1}{2}\pi$

Table 5.1: Positions of the tasks considered by case 1, 2 and 3

Task	x [m]	y [m]	z [m]	θ_{task} [rad]
1	0	5.05	.7	$\frac{1}{2}\pi$
2	2	5.15	.7	$\frac{1}{2}\pi$
3	3.05	3	.79	0
4	3.05	1	.74	0

Table 5.2: Positions of the tasks considered by case 4, 5 and 6

The initial configuration of the MRP is described by its center and orientation. For cases 1, 2 and 3 the initial configuration of the MRP is $q_{\text{mrp}} = \begin{bmatrix} x_{\text{mrp}}^W & y_{\text{mrp}}^W & \theta_{\text{mrp}} \end{bmatrix} = \begin{bmatrix} -2.25 & 3 & \frac{1}{2}\pi \end{bmatrix}$ and for cases 4, 5 and 6 $q_{\text{mrp}} = \begin{bmatrix} -0.5 & 3 & \frac{1}{2}\pi \end{bmatrix}$. All obstacles are boxes of 0.1 [m] by 0.1 [m] and 0.2 [m] high. The size of the obstacles is not related to the functionality of the applied methods. In Table 5.3 the center positions of the obstacles are given.

-	Case 3		Case 6	
obstacle	x [m]	y [m]	x [m]	y [m]
1	-1	4.5	1	4.5
2	1	4.5	2.5	2

Table 5.3: Center positions of the obstacles

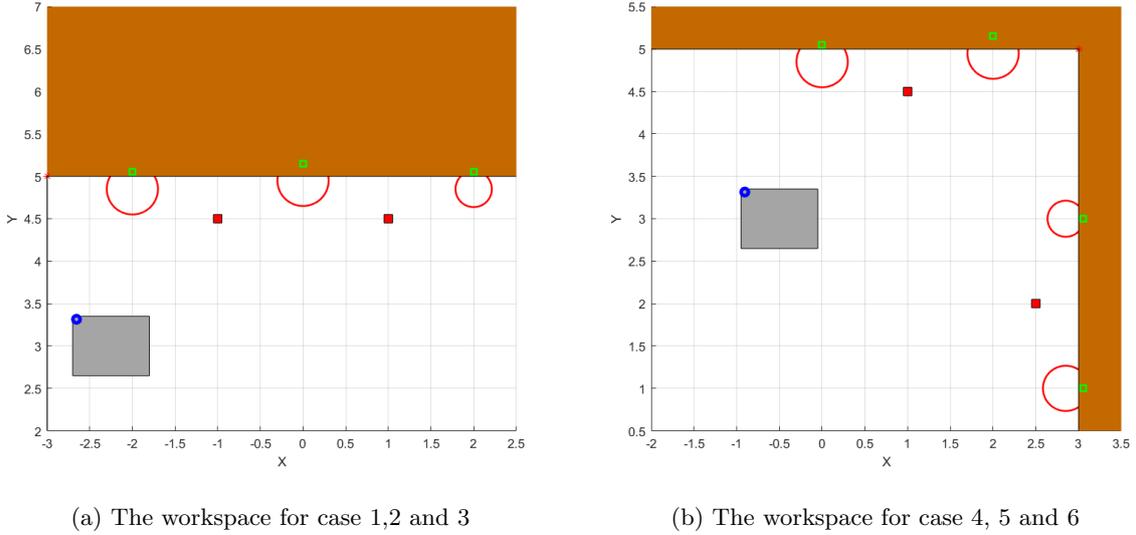


Figure 5.2: The workspaces shown with obstacles

In Figure 5.2 the green squares indicate the task positions as in tables 5.1 and 5.2. The red circles correspond to the circle defined in (4.4) which is related to the reach of the manipulator. The MRP is indicated by the grey rectangle with the blue circle indicating the manipulator position on the MRP. The red squares indicate the obstacles which are only present for cases 3 and 6.

The workspace, initial MRP configuration and task positions can now be used to generate the trajectories, both for the benchmark and optimization cases.

5.3 Result of the first optimization

Two optimizations are performed to result in a trajectory for the MRP and manipulator. The first algorithm is used to reduce computational effort while finding a path for the MRP. The second algorithm optimizes segments of the trajectory of both the MRP and manipulator for a dynamic cost function.

In the first algorithm the MRP path and the stopping configurations are optimized for a kinematic cost function. The optimization considers the complete path of the MRP which is not done during the segmented second optimization. Furthermore, the first optimization guarantees feasibility of both the MRP and manipulator path during task motion. Since the optimization is performed for the kinematic cost function (4.13), only the kinematics are required to be computed. This allows for less time consuming computations which is beneficial while evaluating the complete MRP path.

From the optimized solutions is expected that a motion in a straight line is always preferable. However, in two of the cases obstacles are present which does not allow a straight line. For these cases the MRP will have to make a turn around the obstacles. To reduce the cost of the optimization it might be beneficial to rotate the manipulator towards the tasks to reduce the total movement of the MRP.

5.3.1 The resulting MRP path

The expectation of the optimized result cannot be translated to the cost intuitively. However, the cost can be evaluated to examine the functionality of the algorithm. The optimized solutions can then be simulated and plotted to compare them with the expected results.

The cost of the first algorithm is a function of the path lengths of the wheels and the penalties resulting from the feasibility checks. In Figure 5.3 the cost of the cases is plotted for the iterations which are not influenced by the penalties. It can be seen in figures 5.3b and 5.3d that the initial solution is infeasible resulting in a penalty indicated by the red plot. The initially infeasible solutions are caused by the obstacles which do not allow a straight line motion between the configurations.

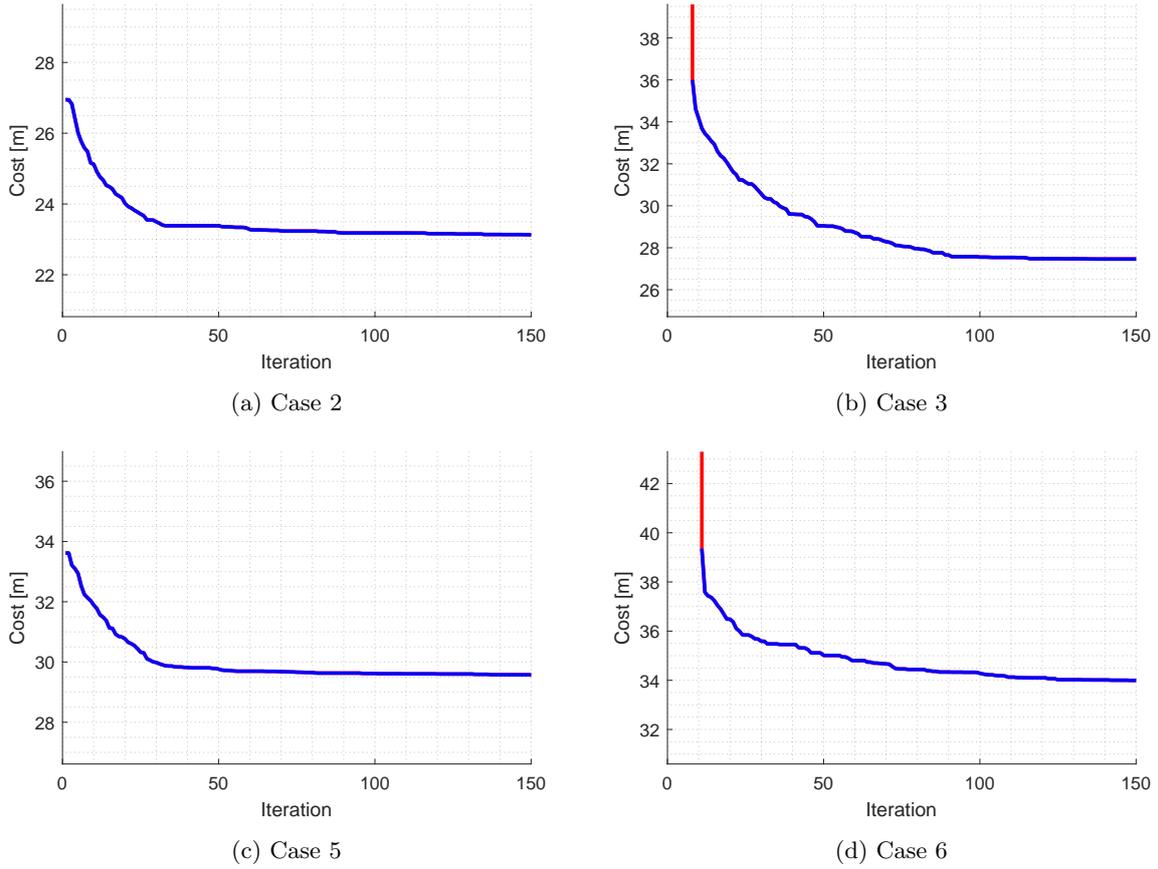


Figure 5.3: Cost of the first algorithm plotted against the iterations

In Figure 5.4 the optimized MRP paths are shown. For the cases without obstacles, a straight line motion can be seen of the center of the MRP as was expected. While driving, the MRP rotates around its center such that the manipulator is ahead of the MRP centerpoint. As a result, the MRP is required to move a shorter distance to place the manipulator at its task position. For the cases with obstacles a similar trend can be seen. The motion is reduced by moving in such a way that the manipulator moves minimally and the obstacles are narrowly avoided.

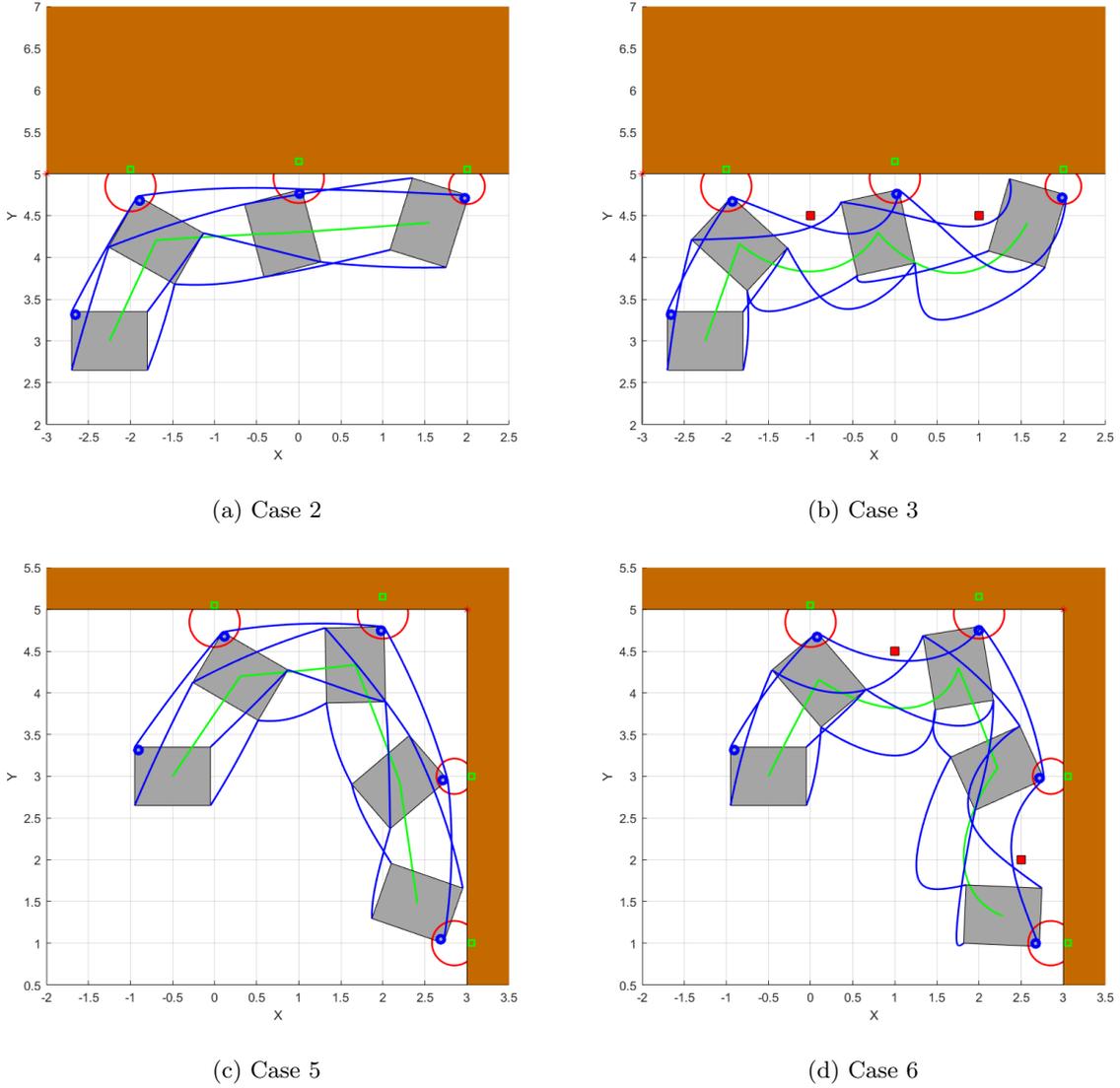


Figure 5.4: Comparison of MRP paths for all optimized cases

In Figure 5.4 the same visualization as in Figure 5.2 is used except the MRP is plotted at each of the stopping configurations. Five paths are plotted, one for each wheel at the corners of the MRP in blue and one for the center position of the MRP in green.

The solutions of the first algorithm do not consider the energy consumption. This means that the computational time is greatly reduced. This is used to efficiently find an optimized solution which considers feasibility of the required motions, collisions and an alternative cost function which can be related to energy consumption. Furthermore, the complete MRP path is considered such that the paths between tasks are affected by each other. This is not possible in the second optimization.

5.4 Second optimization: Trajectory comparison

The first algorithm is used to generate a path for the MRP which takes into account feasibility and an alternative kinematic cost function. With the solution of the first algorithm the trajectories of the manipulator can be defined as result of a dynamic cost function considering the energy consumption. Furthermore, the MRP path is again optimized to result in a trajectory optimized for energy consumption.

5.4.1 Expected solutions

From the four optimized cases is expected that a workspace without obstacles greatly impacts the energy consumption as result of the path length. Without any obstacles present, the MRP can drive freely within the workspace and a straight line between task positions of the MRP is possible. Therefore, the energy consumption of the obstacle free optimized cases considered by the optimization algorithm is expected to be lowest. Adding the obstacles to the workspace requires the MRP to travel a greater distance within the same time period. The increase in required acceleration and velocity is expected to increase the energy consumption by a notable amount. However, since the MRP is only required to start and stop moving once, compared to three times for the benchmark simulation, a reduction of the energy consumption is still expected.

Two reductions of the energy consumption are expected as result of the two algorithms. The first reduction of the energy is expected to come from the MRP path optimization of the first algorithm. The path generated by the first algorithm is used to initialize the second algorithm where it is converted to a trajectory. The second algorithm specifically considers the energy consumption as cost rather than the path length. Therefore, further reduction of the cost is expected. The manipulator trajectory is a function of the MRP configuration which makes it less straightforward to compare the benchmark trajectories with the optimized trajectories.

5.4.2 The resulting energy consumption

The benchmark trajectories for cases 2 and 3 and cases 5 and 6 are identical. Therefore, for each benchmark trajectory two comparisons can be made, namely between the optimized trajectories with and without obstacles. Based on the mentioned expectations the energy consumption of the benchmark trajectory should be largest followed by case 3 and 6 and finally case 2 and 5 with the lowest energy consumption. This result can be seen in figures 5.5 and 5.6.

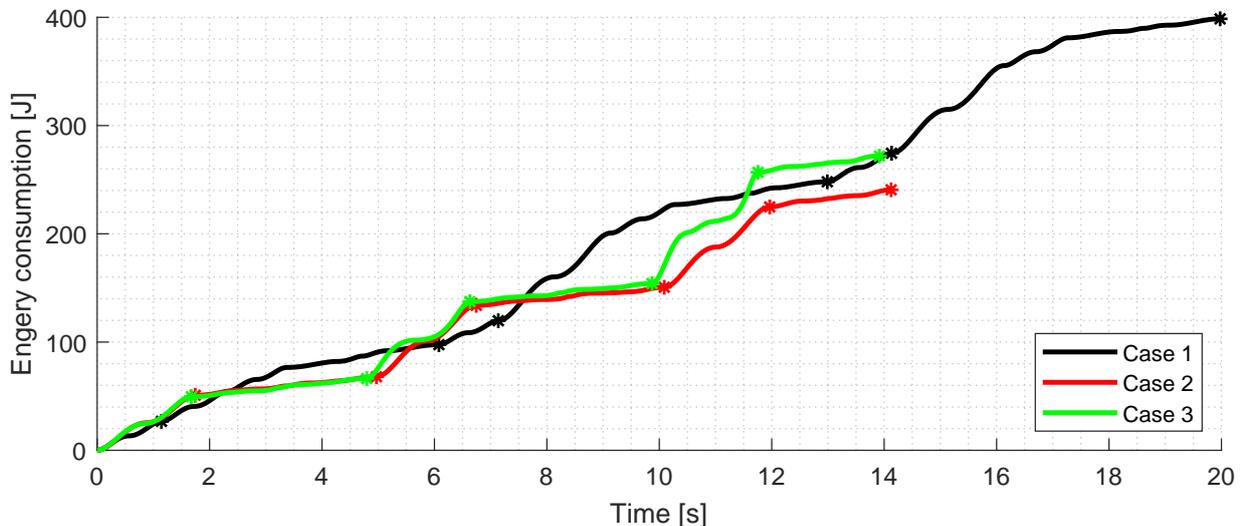


Figure 5.5: Energy consumption over time of cases 1, 2 and 3

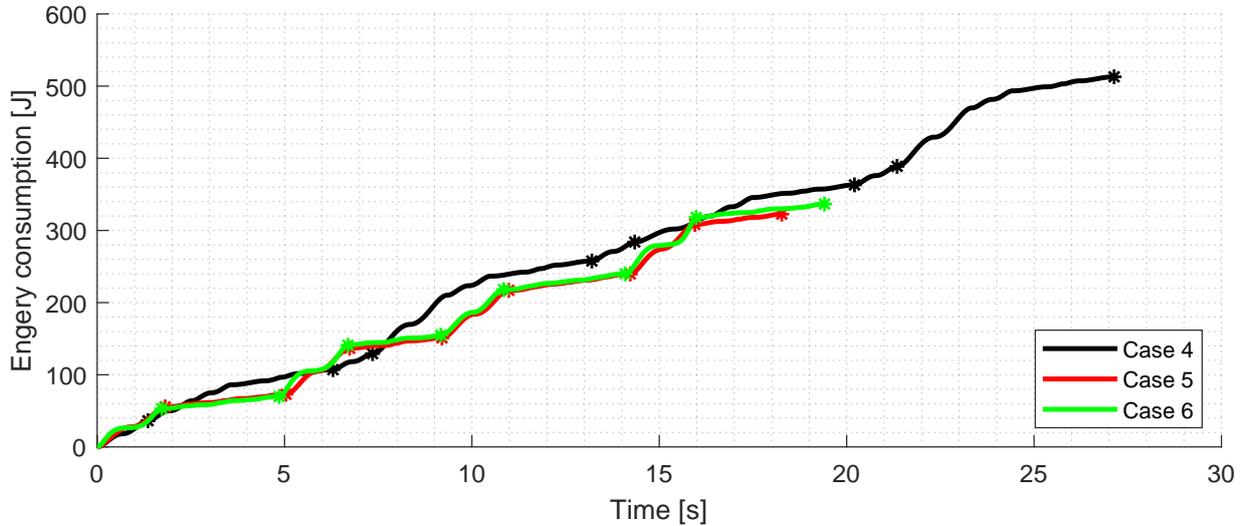


Figure 5.6: Energy consumption over time of cases 4, 5 and 6

In Table 5.4 the energy consumption of each case is given. By comparing case 1 till 3 and 4 till 6 it becomes clear that the expectation was correct. The fixed trajectories exhibit the highest energy consumption and the cases without obstacles the lowest energy consumption.

Case	Total [J]	Manipulator [J]	MRP [J]	MRP path length [m]
1	398.59	59.59	339.01	33.51
2	240.69	49.61	191.07	18.56
3	271.83	48.54	223.29	21.79
4	512.85	79.00	433.85	45.17
5	322.57	68.65	253.92	24.15
6	336.67	71.12	265.55	27.84

Table 5.4: Energy consumption per case

When comparing the energy consumption of each of the cases with the same task configuration it can be seen that both with and without obstacles a significant improvement is achieved. This is the case for both the manipulator and the MRP. With this data alone it is not possible to define how the energy reduction is achieved.

It is not straightforward to compare the results for the manipulator energy consumption. This is because the trajectories are influenced by the configuration of the MRP. For the MRP the path length is considered to be of influence on the energy consumption. A relation can be seen between the path lengths and the energy consumption for both workspace configurations. This can be seen by comparing the MRP path length with the MRP energy consumption where similar ratios are found for all trajectories. The ratios, defined by the MRP energy consumption divided by the MRP path length, can be seen in Table 5.5.

Case	1	2	3	4	5	6
Ratio	10.117	10.294	10.246	9.604	10.514	9.539

Table 5.5: Ratios between the energy consumption and path length of the MRP trajectories

The ratios in Table 5.5 refer to the energy consumption of the MRP divided by the path length of the wheels of the MRP. The ratios of all trajectories are roughly the same but vary too much to link the path length directly to the energy consumption. Moreover, all MRP paths are in the same order of magnitude and thus they do not form a reliable basis to generalize the relation between path length and energy consumption. Furthermore, the weight distribution of the system is not uniform.

5.4.3 Conclusion and next comparison to consider

From the results it can be seen that a significant energy reduction is achieved by the optimization algorithm in comparison with the benchmark. As expected the trajectories without obstacles allow the lowest energy consumption. However, from these results it is not possible to conclude what causes the found reduction. It is hypothesized that the biggest reduction is caused by the reduction of the path length of the MRP which is substantiated by the ratio between the path length and energy consumption.

A second comparison has to be made in order to evaluate how the optimization algorithms combined achieve the found energy reduction. In this comparison the initial and optimized solution of the second algorithm are considered. This could clarify if the cost is reduced by the second algorithm or if the energy consumption is at or close to a minimum by optimizing the path length.

5.5 Achieved cost reduction of the second optimization

It was already found that the optimized trajectories require less energy to be executed than the benchmark trajectories. However, it is unclear if the reduction of the path length or the second optimization algorithm causes the improvement. Moreover, from the previous results, it cannot be concluded if the optimization algorithm is functional for all segments. Therefore, a comparison is made between the initial and optimized solution of the second optimization to define the achieved energy reduction by the second optimization.

In order to examine the functionality of the optimization algorithm, the initial solution of the algorithm is compared to the optimized solution. For each segment the best solution per iteration can be evaluated in order to ascertain a reduction. Cases 2, 3, 5 and 6 from previous section are again considered.

5.5.1 Expected energy reduction

It is expected that the second algorithm is able to reduce the initial cost by a notable amount. However, since the considered path is shorter than the fixed trajectory, it is probable that a portion of the energy reduction is already achieved by the path reduction. This implies moving the MRP between configurations in a single motion instead of three individual motions. It is now possible to compare the manipulator segments because the begin and start configurations are identical. Some of the manipulator segments are defined by more optimization variables than the MRP segments. This could affect the functionality of the algorithm.

5.5.2 Solution comparison

It was mentioned that the goal is to compare the initial solution with the optimized solution to evaluate what is achieved by the optimization algorithm. However, in some cases the initial solution results in collisions. The collisions can occur for the segments in which the manipulator moves from rest configuration to the initial configuration of the task motion and back. The remainder of the segments was shown to be free of collisions and thus the initial solution is always feasible. Because some of the initial solutions are not feasible the choice is made to compare the optimized solution with the first solution which is feasible. These solutions are found by random search.

By using the first feasible solution of each segment figures 5.7 and 5.8 are obtained. In these plots the energy consumed over time is shown. What can be seen is that the energy reduction of most segments is insignificant.

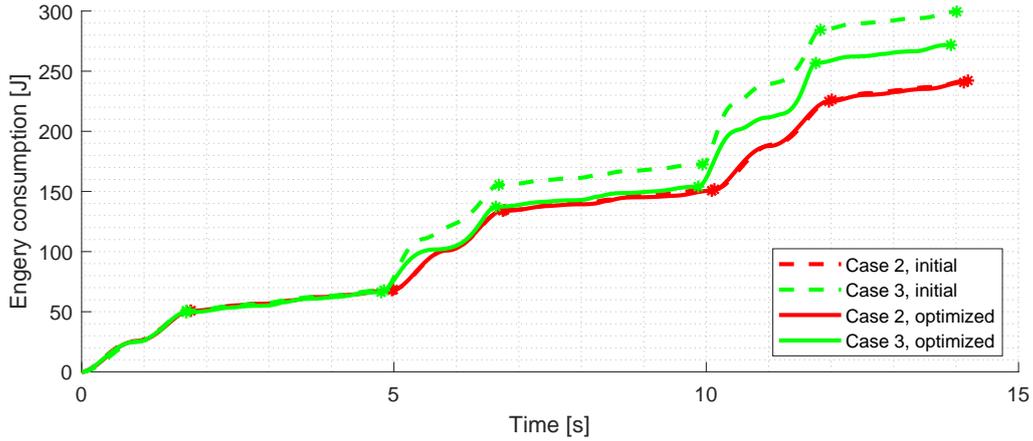


Figure 5.7: Energy consumption over time comparison between initial and optimized solutions of cases 2 and 3

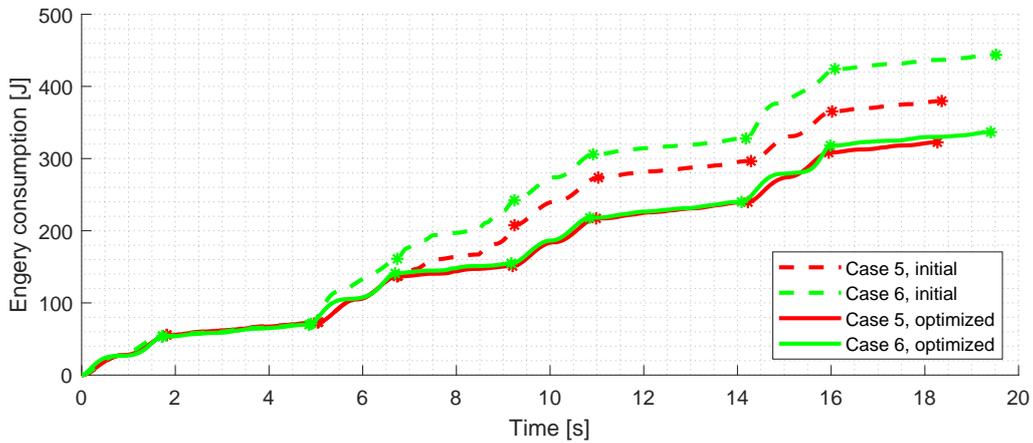


Figure 5.8: Energy consumption over time comparison between initial and optimized solutions of cases 5 and 6

In Table 5.6 the achieved reduction of the initial and optimized solutions are shown in comparison with the benchmark situation. It can be seen that for case 2 the second algorithm barely reduces the cost. For the remaining cases a significant reduction can already be seen at the initial solution but the second algorithm is still able to reduce the cost even further.

Case	Benchmark	Initial		Optimized	
	E [J]	E [J]	Reduction	E [J]	Reduction
2	398.59	242.20	39.24%	240.69	39.62%
3	398.59	299.44	24.88%	271.83	31.80%
5	512.85	379.73	25.96%	322.57	37.10%
6	512.85	443.71	13.48%	336.67	34.35%

Table 5.6: Energy reduction of the initial and optimized solutions

From plots 5.7 and 5.8 it is unclear if the optimization algorithm works and which segments achieve the biggest relative energy reduction. In the figures 5.9, 5.10 and 5.11 the accomplished reduction per

segment is plotted as function of the number of iterations. The algorithm is specified to halt the search after 40 identical solutions. Therefore, the number of iterations varies per segment. The reduction is visualized as fraction of the first feasible solution found by the algorithm. Furthermore, plots of which the initial solution is infeasible are given a red color. Plots corresponding to the task motion are given a green color. MRP and task motion segments are initially feasible as result of the first algorithm.

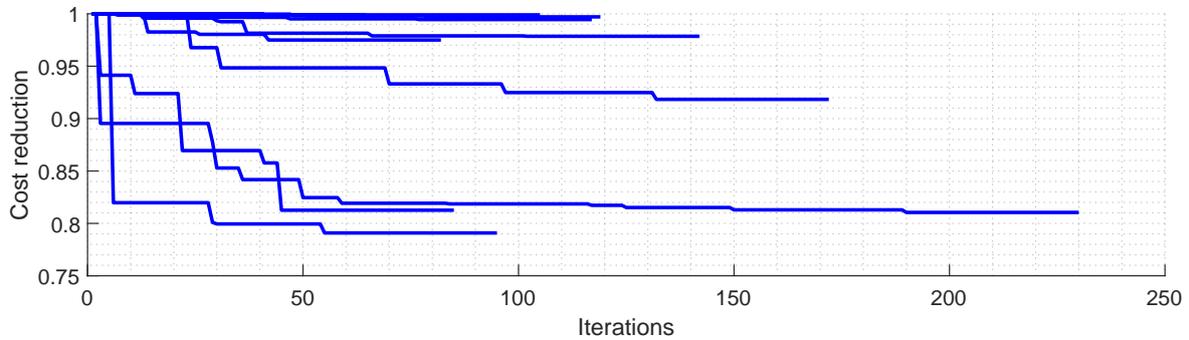


Figure 5.9: The cost reduction as function of the iterations for the MRP

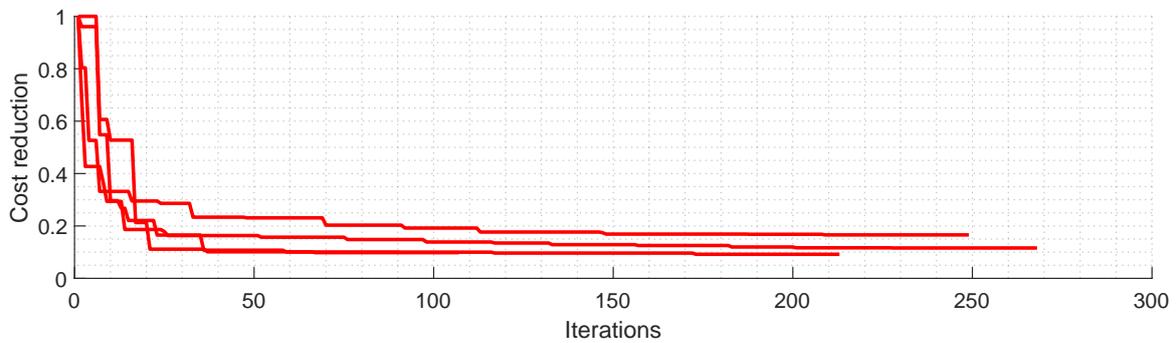


Figure 5.10: The cost reduction as function of the iterations for the manipulator: initially infeasible

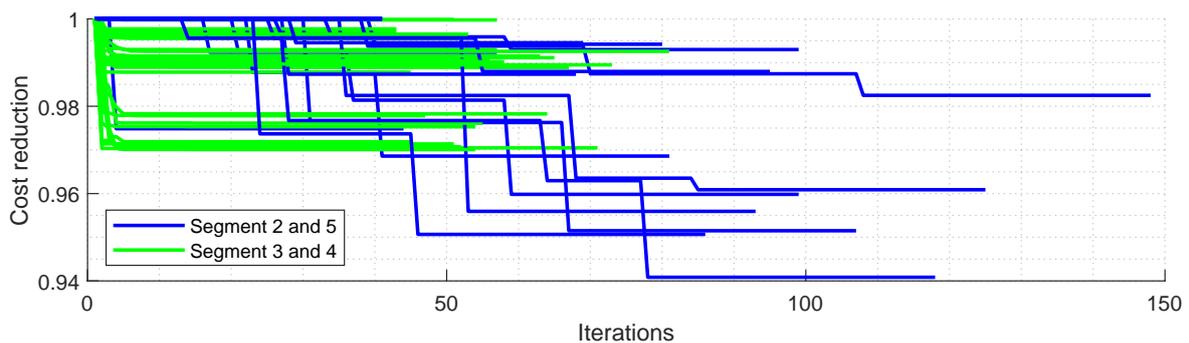


Figure 5.11: The cost reduction as function of the iterations for the manipulator: initially feasible

It can be seen from Figure 5.9 that all initial solutions are feasible for the MRP trajectories, as guaranteed by the first algorithm. Furthermore, a large variety in the energy reduction is achieved. From Figure 5.10 it can be seen that none of the randomly initialized solutions is close to the minimum since the minimum cost reduction is over 80%. In Figure 5.11 the opposite can be observed for the initially feasible solutions. The maximum cost reduction is close to 6% while the plots are densest around the upper range of the graph.

In figures 5.9 and 5.10 a clear gradual decrease of the cost can be observed over the course of the iterations. This observation implies the functionality of the optimization algorithm to be able to reduce

the cost of the segments. However, in Figure 5.11 it can be seen that the cost of most of the segments remains very close to the initial cost, 52 out of 56 solutions. For the manipulator can be suggested that the minimum is close to the initial cost which would explain the insignificant cost reduction. This could also explain the achieved cost reduction when initialized at a random solution as in Figure 5.10.

It is important to note that the same algorithm is applied to multiple segments. The trajectories of each of the segments are defined by different parameters and optimization variables. This could explain why the results of the different segments vary.

5.5.3 Observations and how to continue

The goal of this section is to determine by how much the cost can be reduced by the second optimization in relation to its initial solution. Furthermore, from the results could be concluded if the algorithm works as it should by reducing the cost over the course of the iterations. The same algorithm is applied to segments with varying optimization variables and trajectories. Various relative reductions of the cost can be seen for the segments in which the MRP moves. In a number of cases a significant reduction is found which suggests the algorithm is functional for the MRP segments.

In the segments in which the manipulator moves, a different result is found. The initial solution is infeasible in some of the cases. Therefore, comparing the optimized solution with the initial solution is not possible. The costs of the segments which are initially feasible do not seem to be reduced significantly. This could mean two things; the algorithm is unable to find a better solution even though there does exist a better solution, or the initial solution is close to the best solution and optimization could be argued to be unnecessary. From the segments which are initially unfeasible it can be seen that the algorithm is able to reduce the cost over the course of the iterations which indicates functionality of the algorithm. However, from only four segments the results are too unreliable to draw a conclusion.

From the results it cannot yet be deduced if the optimization algorithm is unable to find a better solution or if the found solution is actually close to the minimum. Therefore, the optimization should be performed again but with other initial conditions. In Figure 5.10 it was seen that the random initialization results in significant reductions of the cost. Therefore, the new optimization is initialized by a random value within the range of the optimization variables. The result of the new optimization could also clarify if the optimized solutions are locally optimal solutions by comparing the optimized solutions of each individual segment.

5.6 The effect of randomized initial solutions

In the previous section it was seen that segments which are initialized by a random solution can show a significant reduction of the cost. However, for the remainder of the segments it is still unclear if the algorithm is stuck in a local minimum, the algorithm is either dysfunctional or actually the best solution is found. Therefore, in this section the same cases as in the previous sections are optimized. The second algorithm is not initialized with fixed optimization variables but instead, the initial solutions are chosen at random.

The same cases are considered as before in order to be able to compare the optimized solutions. Instead of initiating the algorithm with the fixed values in the middle of their range the algorithm is initialized with random values. The parameters describing the path of the MRP are still chosen to be the values found in the first algorithm. The configurations at which the mobile manipulator stops are the same or a comparison would not be possible.

The cases from previous section with the fixed initial solutions are denoted by $\mathbf{C}(i, \mathbf{F})$, the new cases with random initial conditions are denoted by $\mathbf{C}(i, \mathbf{R})$.

5.6.1 Expected differences

It is expected that the costs of the initial and optimized solutions of $\mathbf{C}(i, \mathbf{R})$ vary more than the solutions of $\mathbf{C}(i, \mathbf{F})$. This is because the initially infeasible and thus random solutions of $\mathbf{C}(i, \mathbf{F})$ have shown the largest improvement. The improvement results from the randomization during optimization in order to

find a feasible solution. Furthermore, it is expected that the algorithm is now able to reduce the cost significantly for all segments opposed to previous section. Since it is possible that the previously found solutions are at a local minimum, better, equal or worse solutions can be found by initializing randomly. Because a large number of solutions are possible it is not expected that the optimized solutions match perfectly with the previously obtained solutions.

5.6.2 Optimization results

In figures 5.12 and 5.13 the optimized solutions are again plotted against the benchmark. From the plots it is not possible to notice a difference between the comparison in Section 5.4.2.

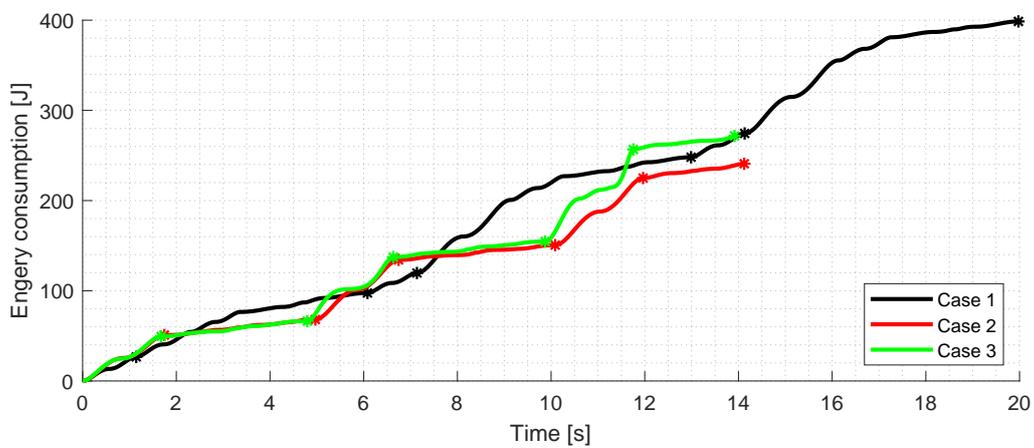


Figure 5.12: Energy consumption over time of cases 1, 2 and 3 initialized by $\mathbf{C}(i, \mathbf{R})$

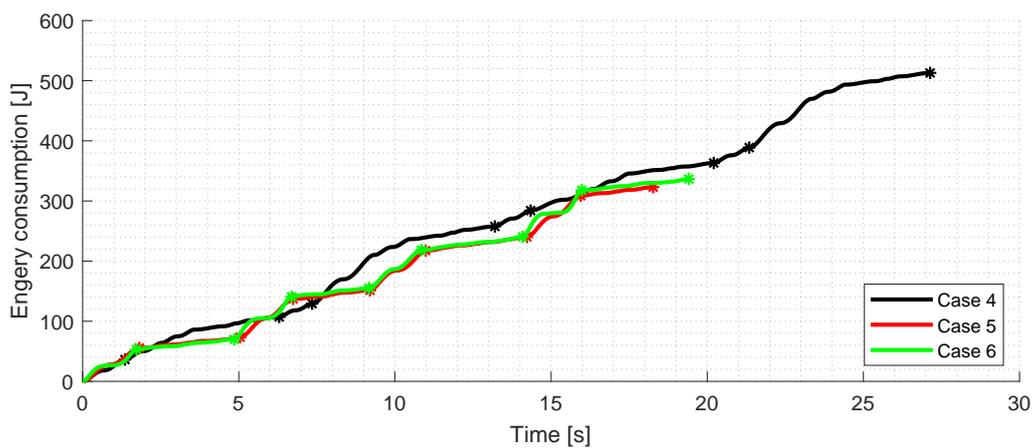


Figure 5.13: Energy consumption over time of cases 4, 5 and 6 initialized by $\mathbf{C}(i, \mathbf{R})$

In the figures 5.14 and 5.15 the energy consumption over time is plotted for all $\mathbf{C}(i, \mathbf{R})$. Compared to the previous section a significant difference between the first feasible solutions and optimized energy consumption can be seen.

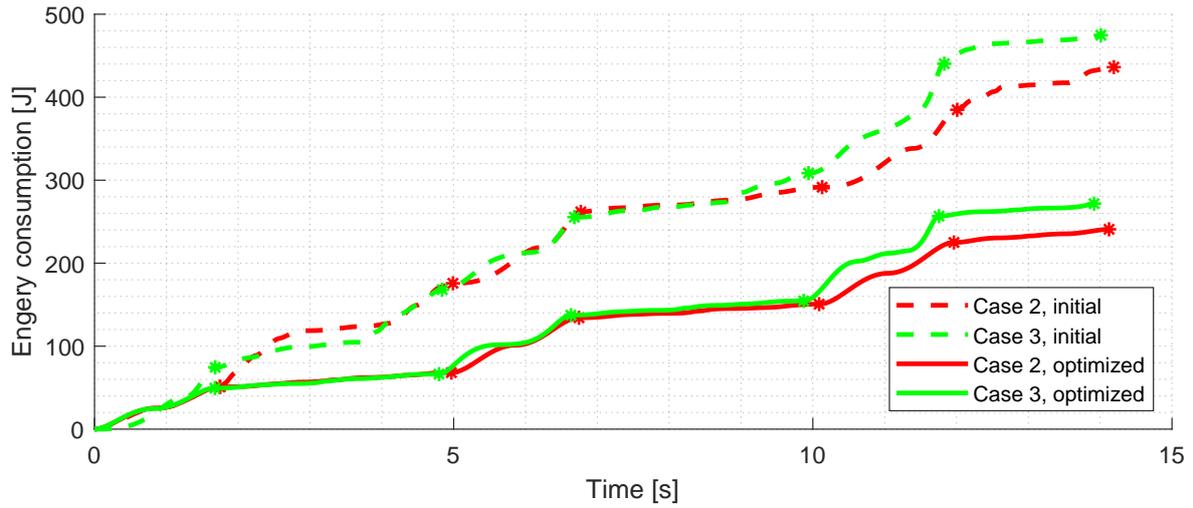


Figure 5.14: Energy consumption comparison of initial and optimized solutions for cases 2 and 3

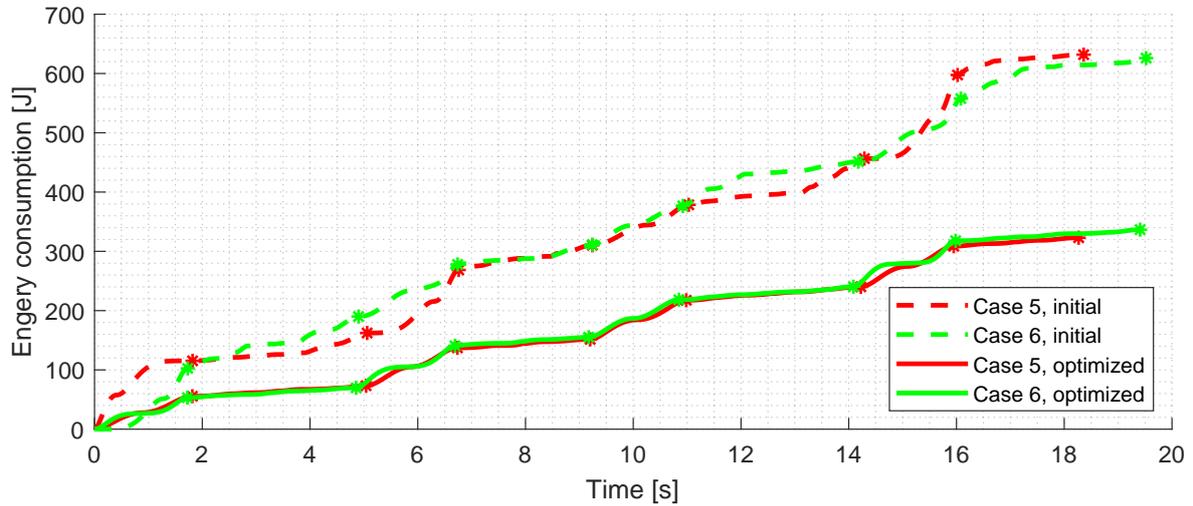


Figure 5.15: Energy consumption comparison of initial and optimized solutions for cases 5 and 6

The best cost per population as function of the iterations is again plotted in figures 5.16 and 5.17. It is expected that similar plots can be seen as in Figure 5.10. All segments in which the MRP drives and in which the task motion is performed are initially feasible. All segments in which the manipulator moves between the task configuration and rest configuration are initially infeasible.

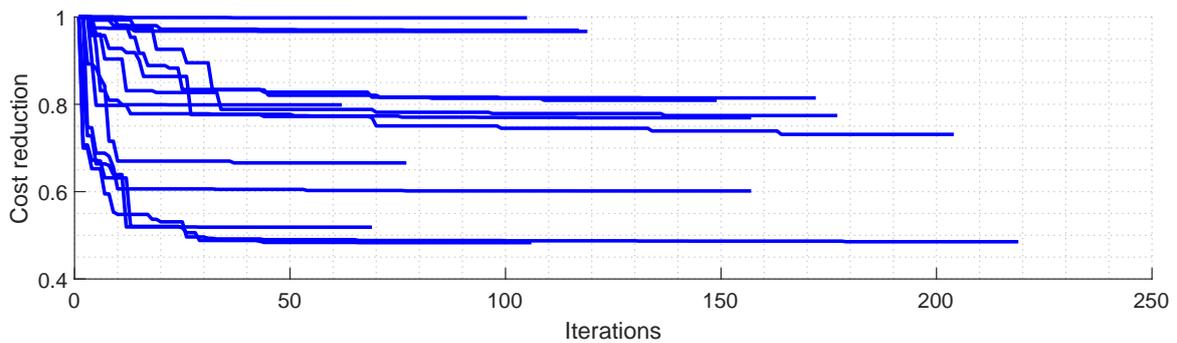


Figure 5.16: The cost reduction as function of the iterations of the MRP segments

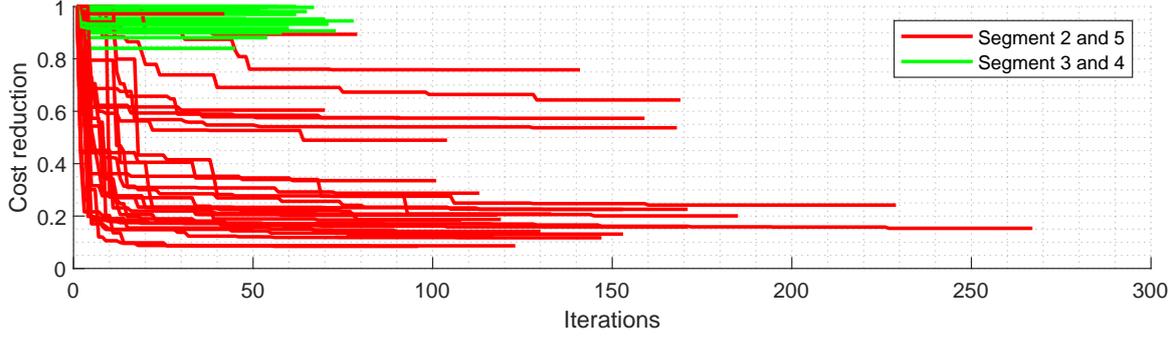


Figure 5.17: The cost reduction as function of the iterations of the manipulator segments

In figures 5.16 and 5.17 the achieved cost reduction is shown for the each of the segments for all $\mathbf{C}(i, \mathbf{R})$. The blue plots correspond to the MRP segments and the red and green plots to the manipulator segments. The red plots indicate the segments which are initially infeasible, which are all the initial solutions of segment 2 and 5. All task motion segments are initially feasible.

In Figure 5.16 the cost reduction of the optimized solution of the MRP trajectories is shown in relation to the initial solutions. Even though the initial path, as result of the first optimization, remains the same, a big difference in the cost reduction is found. This is because $s(t)$ is still randomized.

In Section 5.5.2 it was found that of all random solutions, which are the initially infeasible solutions, a significant reduction of the cost is found. This is again the case in Figure 5.17 indicated by the red plots. However it can be seen that by randomly initializing the segments of the task motion the cost reduction is limited. Opposed to the MRP trajectories the influence of the variable $s(t)$ is limited. This is because $s(t)$ is the only variable during task motions.

5.6.3 Comparison of the optimized solutions

The first comparison to make between the optimization with fixed and random initial variables is the resulting cost. The cost of the complete trajectory is considered in Table 5.7.

i	$E_{\mathbf{C}(i, \mathbf{F})}$ [J]	$E_{\mathbf{C}(i, \mathbf{R})}$ [J]	$\frac{E_{\mathbf{C}(i, \mathbf{F})}}{E_{\mathbf{C}(i, \mathbf{R})}}$ [-]
2	240.69	240.84	0.9994
3	271.83	271.74	1.0003
5	322.57	323.02	0.9986
6	336.67	336.61	1.0002

Table 5.7: The optimized cost of the cases with both initial variables

From the table above the conclusion could be drawn that the optimized results are sufficiently close to each other and thus the solutions must be near identical. However, the cost is a sum of the cost of a number of segments. Therefore, individual segments could vary greatly while still resulting in nearly the same total cost. In the next section the individual segments are compared as function of the cost and optimized solutions.

Individual segments

A comparison is made between the total cost of $\mathbf{C}(i, \mathbf{F})$ and $\mathbf{C}(i, \mathbf{R})$. It was found that the costs are nearly identical. The total cost is a function of a lot of variables which is not beneficial to directly compare results. Therefore, the cases and segments are separated and individual results are compared. The individual comparison can be used to evaluate if the optimized solutions are defined by the same

optimization variables for $\mathbf{C}(i, \mathbf{F})$ and $\mathbf{C}(i, \mathbf{R})$.

The difference between the optimization variables is indicated per segment. Each of the optimization variables is defined by a number of samples and a range from which the variable can take a value. Each variable is sampled by the indices $i \in [1, n_{o_i}]$. Since the number of samples per variable can vary, the difference between the solutions is divided by the number of samples such that

$$i_{o,D} = \frac{i_{o,F} - i_{o,R}}{n_{o_i}} \quad (5.1)$$

where $-1 \leq i_{o,D} \leq 1$. The values of $i_{o,F}$ and $i_{o,R}$ correspond to the optimized solutions of $\mathbf{C}(i, \mathbf{F})$ and $\mathbf{C}(i, \mathbf{R})$.

In Figure 5.18 $i_{o,D}$ is plotted for all optimization variables per segment. The complete trajectory consists of five segments per task resulting in fifteen or twenty segments. For the recurring first segment four variables are used, for the second and fifth segment seven variables are used and for the third and fourth segment only one variable is used. This can be seen by the number of dots per segment.

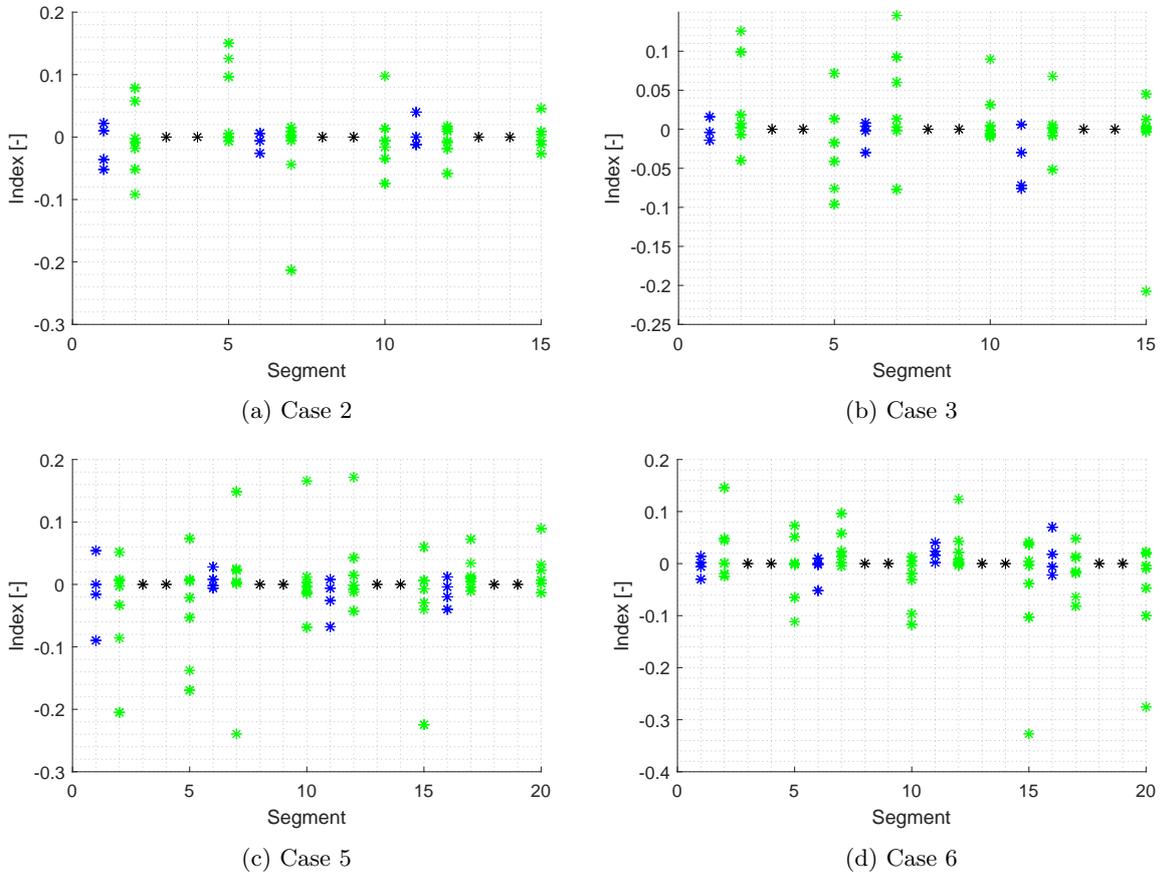


Figure 5.18: Comparison of the optimization variables per segment

When each $i_{o,D}$ is at 0 the optimized solutions resulting from both initial conditions are identical. This can be seen in Figure 5.18 by the black plots which are used for the segments of the constrained task motion. The only optimization variable is related to $s(t)$. Because of the small number of possible solutions both optimizations result in the same solutions for all segments. The blue plots are used for the segments of the MRP motion. In these segments four optimization variables are used which allows a larger number of possible solutions. It can therefore be seen that the solutions are never completely identical. The segments in which the manipulator trajectory is optimized shows the biggest variation of solutions as can be seen by the green plots. The manipulator trajectory is optimized by seven optimization variables and thus allows the largest variation of possible solutions. Therefore, the probability of finding the same solution twice is lowest and a higher variation of solutions is found.

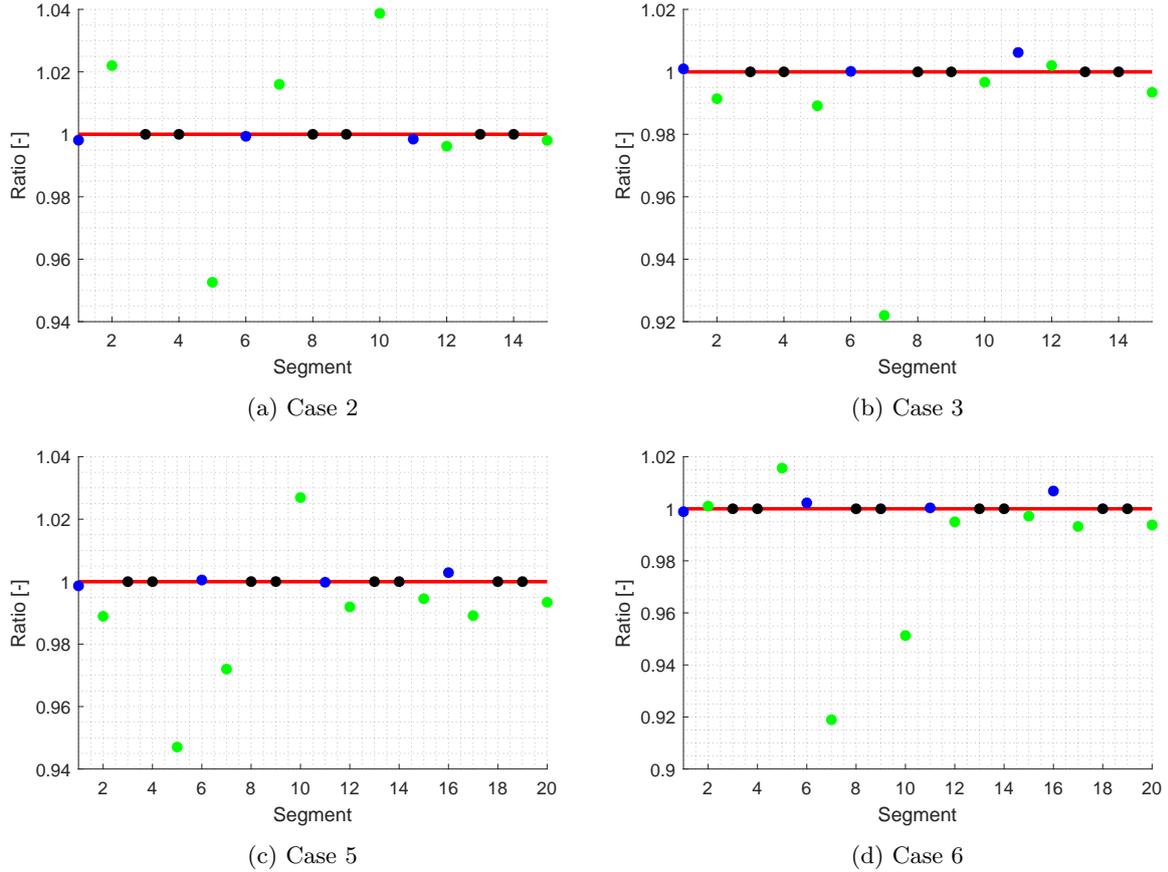


Figure 5.19: Comparison of the cost per segment

In Figure 5.19 the costs of each of the segments are compared as was done in Table 5.7. The costs $\mathbf{C}(i, \mathbf{F})$ are divided by the costs of $\mathbf{C}(i, \mathbf{R})$ and thus a value of 1 means that the cost is identical. This can be seen in the segments of the constrained task motion again indicated by the black dots. This was already known since the solutions were shown to be identical. Comparing the MRP segments and manipulator segments as before a relation can be found. If the variation between the solutions is larger also a larger difference between the cost is found.

It is likely that for a number of segments 1,2 and 5 the algorithm gets stuck in a local minimum which results in a larger difference between the solutions and possibly also the costs. For some of the segments a smaller difference between the solutions can be seen which indicates that both solutions converge to the same minimum.

With the current settings the algorithm does not always converge to the same solution with varying initial conditions. Therefore, it can be suggested that both solutions are not the absolute minimal solution. At least for the total trajectory can be concluded with certainty that neither solutions are optimal. This is because from Figure 5.19 it can be seen that for numerous segments the ratio of the cost of $\mathbf{C}(i, \mathbf{F})$ and $\mathbf{C}(i, \mathbf{R})$ is both above and below 1.

In Table 5.8 the best segments of $\mathbf{C}(i, \mathbf{F})$ and $\mathbf{C}(i, \mathbf{R})$ are combined. The resulting solution is a minimal improvement of the already existing solutions.

i	$\mathbf{C}(i, \mathbf{F})$ [J]	$\mathbf{C}(i, \mathbf{R})$ [J]	Best segments [J]	Improvement
2	240.69	240.84	240.27	0.172%
3	271.83	271.74	271.13	0.225%
5	322.57	323.02	322.23	0.104%
6	336.67	336.61	335.87	0.222%

Table 5.8: The optimized cost of the cases with both initial variables and best segments combined

5.6.4 Concluding remarks

From the results it can be seen that the algorithm is able to reduce the cost of each of the segments. Moreover, the optimized costs of $\mathbf{C}(i, \mathbf{F})$ and $\mathbf{C}(i, \mathbf{R})$ lie close to each other. This means that the optimization of $\mathbf{C}(i, \mathbf{F})$ was effective but the optimized solution is in most cases not significantly better than the initial solution. Therefore, it was falsely suggested that the optimization was not effective. The optimization of $\mathbf{C}(i, \mathbf{R})$ does result in significantly better solutions while comparing the optimized and initial solutions. This gives the impression that the algorithm is only functional while initialized with random values. However, the initial cost is significantly larger than the optimized solution.

5.7 Summary

The trajectory generation method described in chapters 3 and 4 is tested. During these tests results are gathered which are discussed in this chapter. The achieved energy reduction of the optimized trajectories have shown a significant decrease. However, while comparing the energy consumption of the benchmark and optimized trajectories it became clear that additional results were needed to evaluate the effectiveness of the algorithms. Therefore, comparisons were made between the initial and optimized solutions of the second algorithm. These have shown that the second algorithm is able to reduce the energy consumption as required and expected.

Chapter 6

Conclusions and recommendations

The goal of this project was to formulate a trajectory generation method for a mobile manipulator. The trajectory generator would have to be able to avoid obstacles and be an improvement of the current trajectory generation method. The quality of a trajectory, or cost, is measured by the energy consumption. It is shown in Chapter 5 that by implementing the trajectory generation method described in chapters 3 and 4 an improvement can be achieved. In the next section the elaborate conclusions are discussed. In the section thereafter recommendations are given for future studies and projects concerning similar subjects.

6.1 Conclusions

The optimization strategy results in a lower cost in relation to the benchmark. However, comparing the initial solution of the optimization with the optimized solution only a minor improvement can be seen. From these results it could not be concluded if the algorithm is functional and able to reduce the initial cost. When the initial solution is randomized approximately the same cost is found as for the fixed initial condition. From this result it can be concluded that the algorithm is able to optimize the trajectory while initialized by either fixed optimization variables or random optimization variables to arrive at nearly the same cost.

The number of iterations needed is affected by the initial condition. The collision check is performed by means of a boolean check. This means that upon detection of a collision the cost function is set to a value, a penalty, which results in a variation of the cost of predetermined values. An evolutionary search algorithm requires a variation of the cost between the solutions in order to compute a better solution. If the costs of all solutions are the same value, as function of the same penalty, only a random search is effective until solutions with a greater variation are found. When a feasible solution is found the algorithm effectively reduces the cost over the iterations.

The second optimization algorithm generates solutions in the same way for each of the segments. However, the number of optimization variables per segment varies, namely one, four and seven. Therefore, the manipulator segments defined by seven optimization variables show a bigger difference between the optimized solutions initialized by fixed or random parameters.

6.2 Recommendations

During the project it was uncertain which methods and strategies would work. A choice had to be made which method would result in a high success rate of being able to generate a feasible trajectory while still containing sufficient characteristics of a realistic situation. Furthermore, a number of different aspects of the optimization, simulation and dynamics would have to be considered. Focusing on merely one aspect of the project would mean an uneven distribution of time and effort as well as a risk of certain aspects to underperform or not be finished at all. Moreover, not enough time is available to work out every detail as would be desired. As a result a number of alternatives can be recommended to be included in a future

study concerning mobile manipulators.

The decision was made to separate the MRP and manipulator optimization completely in terms of energy optimization. The complete MRP path is used in the first algorithm to estimate stopping configurations which would benefit the energy reduction. The weight of the MRP is significantly larger than the weight of the manipulator and thus the manipulator motion is a function of the MRP configurations. The manipulator trajectory is not optimized for each possible MRP configuration to reduce the computational effort. However, in Section 5.5 it can be seen that the possible cost reduction of $\mathbf{C}(i, \mathbf{F})$ is insignificant during the second algorithm. Therefore, it is suggested to optimize the MRP trajectory and configurations for energy consumption using the manipulator initial solution of $\mathbf{C}(i, \mathbf{F})$. This can be done since it could be seen that in a majority of the cases optimization does not alter the trajectory significantly. Optimization is still required in the case of initially infeasible trajectories.

In Section 5.6.3 the optimized solutions are compared. It was found that the number of optimization variables results in various magnitudes of cost and solution differences. The segments for which the solutions are near identical show that the algorithm can approach the optimal solution effectively. It can therefore be suggested to increase the complexity of the optimization to increase the potential energy reduction. This can be done by increasing the order of the polynomials or opting for a spline formulation with more optimization variables. Alternatively, the algorithm stopping criterion can be eased by requiring less iterations of the same best cost to avoid unnecessary computations. The segments for which the optimized solutions deviate more significantly the stopping criterion could be altered such that the algorithm is allowed to increase the number of iterations to arrive at a potential minimum.

It was mentioned that evolutionary algorithms do not function efficiently when the resulting costs of the generated solutions are identical or do not consist of a large number of varying values. When the collision between the mobile manipulator and the environment are evaluated the penalties are used to indicate that a collision is detected. These penalties cannot differentiate between collisions and thus it is impossible for the algorithm to determine which solution is better or worse. A better collision detection method would be to determine by how much the mobile manipulator intrudes the environment. This can be done analytically by defining three-dimensional shapes to cover the system. The analytic description can also significantly reduce the computational effort since the mobile manipulator does not have to be described by a point cloud.

Appendix A

Toppling of the MRP during actuation

During extreme actuation it is possible that the reaction force between the wheels and the floor becomes negative. However, the MRP is not fixed to the floor and thus the reaction force can only be positive or zero. As a result the MRP could start to topple. When this is the case the models are no longer capable of describing the kinematics and dynamics of the system. Therefore, it should be guaranteed that toppling of the MRP can never be initiated. Since the motion of the MRP and the manipulator are separated it is possible to express the toppling point of the MRP in by three parameters opposed to the nine degrees of freedom.

To evaluate the toppling limits of the MRP an analytical description is derived of the normal forces of the wheels with the floor. If the moment applied to the MRP around the x - or y -axis is too large the MRP can topple, where the axis of rotation is the line between the contact points of two wheels on the same side of the MRP. It is only required to evaluate the toppling limits of the MRP and not the behavior thereafter. The forces working on the MRP are shown in Figure A.1.

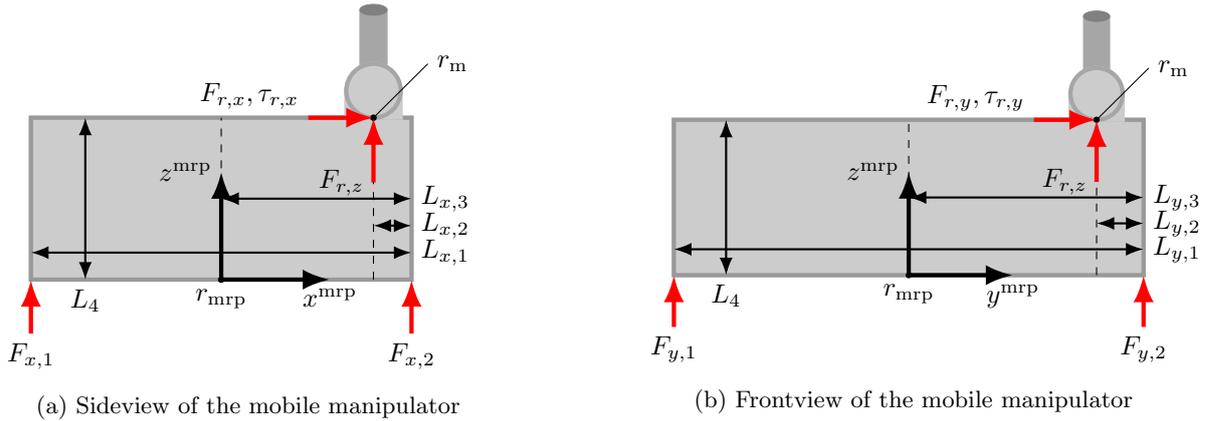


Figure A.1: Schematics of the mobile manipulator

The rest configuration of the manipulator visualized in Figure A.2. The tucked-in configuration reduces the height of the center of mass and thus also the resulting moment due to accelerations. The rest configuration $\theta_{\text{rest}} = \left[\frac{1}{2}\pi \quad -\frac{1}{6}\pi \quad \frac{2}{3}\pi \quad 0 \quad \frac{1}{2}\pi \quad 0 \right]$.

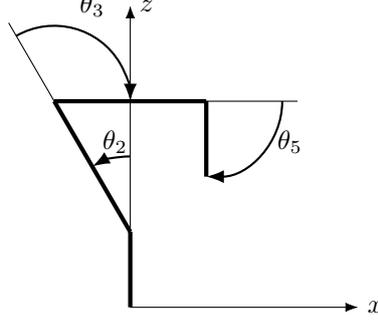


Figure A.2: Manipulator schematic of the rest configuration

The forces acting of the wheels are a function of the acceleration expressed in the orientation of the MRP with respect to the world fixed frame ${}^W\ddot{r}_{\text{mrp}}$ and the reaction forces and moments with the manipulator F_r and τ_r . The height of the center of mass of the MRP $L_5 = \frac{1}{2}L_4$.

$$\begin{aligned}
F_{x,1} &= \frac{1}{L_{x,1}} \left(m_{\text{mrp}}g L_{x,3} - F_{r,z}L_{x,2} - \tau_{r,y} - F_{r,x}L_4 + m_{\text{mrp}} {}^W\ddot{x}_{\text{mrp}}L_5 \right) \\
F_{x,2} &= \frac{1}{L_{x,1}} \left(m_{\text{mrp}}g(L_{x,1} - L_{x,3}) - F_{r,z}(L_{x,1} - L_{x,2}) + \tau_{r,y} + F_{r,x}L_4 - m_{\text{mrp}} {}^W\ddot{x}_{\text{mrp}}L_5 \right) \\
F_{y,1} &= \frac{1}{L_{y,1}} \left(m_{\text{mrp}}g L_{y,3} - F_{r,z}L_{y,2} + \tau_{r,x} - F_{r,y}L_4 + m_{\text{mrp}} {}^W\ddot{y}_{\text{mrp}}L_5 \right) \\
F_{y,2} &= \frac{1}{L_{y,1}} \left(m_{\text{mrp}}g(L_{y,1} - L_{y,3}) - F_{r,z}(L_{y,1} - L_{y,2}) - \tau_{r,x} + F_{r,y}L_4 - m_{\text{mrp}} {}^W\ddot{y}_{\text{mrp}}L_5 \right)
\end{aligned} \tag{A.1}$$

The forces F_x and F_y can be rewritten as function of the variables \ddot{x}_{mrp} , \ddot{y}_{mrp} , $\ddot{\theta}_{\text{mrp}}$ and $\dot{\theta}_{\text{mrp}}^2$. By doing so the relation between the forces and the variables can be evaluated to determine the operating limits. The forces F_x are only dependent on \ddot{x}_{mrp} and the forces F_y are only dependent on \ddot{y}_{mrp} . The values of $c_{\ddot{r}}$, $c_{\ddot{\theta}}$, $c_{\dot{\theta}}$ and c_g in (A.2) are constant and are related to the MRP and manipulator configuration, dimensions and masses. The values of c_g correspond to the constant force related to gravity and the positions of the masses.

$$\begin{aligned}
F_{x,1} &= \ddot{x}_{\text{mrp}}c_{\ddot{r},1} + \ddot{\theta}_{\text{mrp}}c_{\ddot{\theta},1} + \dot{\theta}_{\text{mrp}}^2c_{\dot{\theta},1} + c_{g,1} \\
F_{x,2} &= \ddot{x}_{\text{mrp}}c_{\ddot{r},2} + \ddot{\theta}_{\text{mrp}}c_{\ddot{\theta},2} + \dot{\theta}_{\text{mrp}}^2c_{\dot{\theta},2} + c_{g,2} \\
F_{y,1} &= \ddot{y}_{\text{mrp}}c_{\ddot{r},3} + \ddot{\theta}_{\text{mrp}}c_{\ddot{\theta},3} + \dot{\theta}_{\text{mrp}}^2c_{\dot{\theta},3} + c_{g,3} \\
F_{y,2} &= \ddot{y}_{\text{mrp}}c_{\ddot{r},4} + \ddot{\theta}_{\text{mrp}}c_{\ddot{\theta},4} + \dot{\theta}_{\text{mrp}}^2c_{\dot{\theta},4} + c_{g,4}
\end{aligned} \tag{A.2}$$

The reaction moments and forces of the manipulator result from the stationary manipulator. Therefore, the manipulator is considered as a single body. The center of mass is a function of the individual centers of mass of each link and their corresponding mass:

$$r_{\text{CoM}} = \sum_{i=1}^n \frac{m_i r_{\text{CoM},i}}{m_{\text{tot}}}, \tag{A.3}$$

where $m_{\text{tot}} = \sum_{j=1}^n m_j$ is the total mass of the manipulator. The center of mass of each link is located at the center of the link and thus the center of mass vectors are

$$\begin{aligned}
r_{\text{CoM},1} &= \begin{bmatrix} 0 & 0 & \frac{L_{m,1}}{2} \end{bmatrix}^\top, \\
r_{\text{CoM},2} &= \begin{bmatrix} -\frac{1}{2}s_1 L_{m,2} & 0 & L_{m,1} + \frac{1}{2}c_1 L_{m,2} \end{bmatrix}^\top, \\
r_{\text{CoM},3} &= \begin{bmatrix} -s_1 L_{m,2} - \frac{1}{2}s_{2,3} L_{m,3} & 0 & L_{m,1} + c_1 L_{m,2} + \frac{1}{2}c_{2,3} L_{m,3} \end{bmatrix}^\top, \\
r_{\text{CoM},4} &= \begin{bmatrix} -s_1 L_{m,2} - s_{2,3} L_{m,3} - \frac{1}{2}s_{2,3,5} L_{m,4} & 0 & L_{m,1} + c_1 L_{m,2} + c_{2,3} L_{m,3} + \frac{1}{2}c_{2,3,5} L_{m,4} \end{bmatrix}^\top
\end{aligned} \tag{A.4}$$

where $L_{m,i}$ is the length of link i . The reaction forces F_r and reaction moments τ_r result from the acceleration of the center of mass of the manipulator. The only angular acceleration is around the z -axis. The reaction moment around the z -axis does not influence the normal forces and thus the moment of inertia is not considered.

Rotation of the MRP results in accelerations of the manipulator relative to the MRP center as function of both $\ddot{\theta}_{\text{mrp}}^2$ and $\ddot{\theta}_{\text{mrp}}$. The total acceleration of the manipulator center of mass is

$${}^W \ddot{r}_{\text{CoM}}^{\text{mrp}} = \ddot{r}_{\text{mrp}}^{\text{mrp}} + \ddot{\theta}_{\text{mrp}} \times (r_{\text{m}}^{\text{mrp}} + r_{\text{CoM}}) + \dot{\theta}_{\text{mrp}} \times (\dot{\theta}_{\text{mrp}} \times (r_{\text{m}}^{\text{mrp}} + r_{\text{CoM}})) \tag{A.5}$$

where \ddot{r}_{CoM} utilizes the same notation as \ddot{r}_{mrp} and is therefore expressed in the orientation of the MRP fixed frame. In the MRP fixed frame the accelerations due to rotation, and thus their reaction forces, always work in the same direction since r_{m} and r_{CoM} are constant. The reaction moments result from the crossproduct of the center of mass position and the force due to acceleration.

$$\begin{aligned}
\tau_{r,x} &= m_{\text{tot}} z_{\text{CoM}} (\ddot{y}_{\text{mrp}} + \ddot{y}_{\text{CoM}}) + m_{\text{tot}} y_{\text{CoM}} g, \\
\tau_{r,y} &= -m_{\text{tot}} x_{\text{CoM}} g - m_{\text{tot}} z_{\text{CoM}} (\ddot{x}_{\text{mrp}} + \ddot{x}_{\text{CoM}}), \\
F_r &= -m_{\text{tot}} (\ddot{r}_{\text{mrp}} + \ddot{r}_{\text{CoM}}) - Mg
\end{aligned} \tag{A.6}$$

The following values are found for the parameters $c_{\ddot{r}}$, $c_{\ddot{\theta}}$, $c_{\dot{\theta}}$ and c_g of the mobile manipulator:

-	$c_{\ddot{r}}$	$c_{\ddot{\theta}}$	$c_{\dot{\theta}}$	c_g
1	-12.0714	-2.2854	-1.7775	153.00
2	12.0714	2.2854	1.7775	207.00
3	-9.3889	1.3825	-1.7775	153.00
4	9.3889	-1.3825	1.7775	207.00

Table A.1: The constant parameters of the normal forces of the MRP wheels with the floor

The goal is now to find the maximum and minimum values of \ddot{x}_{mrp} , \ddot{y}_{mrp} , $\ddot{\theta}_{\text{mrp}}$ and $\dot{\theta}_{\text{mrp}}^2$ such that the forces acting on the wheels remain positive. The maximum or minimum allowed values of the variables are given below in Table A.2 and A.3 and result from $\frac{c_{g,i}}{c_{\ddot{r},i}}$ or $\frac{c_{g,i}}{c_{\ddot{\theta},i}}$ or $\frac{c_{g,i}}{c_{\dot{\theta},i}}$. The values thus do not take into account the other parameters.

-	$\frac{c_g}{c_{\ddot{r}}}, \ddot{x}_{\text{mrp}} [\frac{\text{m}}{\text{s}^2}]$	$\frac{c_g}{c_{\ddot{\theta}}}, \ddot{\theta}_{\text{mrp}} [\frac{\text{rad}}{\text{s}^2}]$	$\frac{c_g}{c_{\dot{\theta}}}, \dot{\theta}_{\text{mrp}}^2 [\frac{\text{rad}^2}{\text{s}^2}]$
$F_{x,1}$	12.6746	66.948	86.0759
$F_{x,2}$	-17.1479	-90.5767	-116.4557

Table A.2: Maximum and minimum allowed values of the variables related to F_x

-	$\frac{c_g}{c_{\ddot{r}}}, \ddot{y}_{\text{mrp}} [\frac{\text{m}}{\text{s}^2}]$	$\frac{c_g}{c_{\ddot{\theta}}}, \ddot{\theta}_{\text{mrp}} [\frac{\text{rad}}{\text{s}^2}]$	$\frac{c_g}{c_{\dot{\theta}}}, \dot{\theta}_{\text{mrp}}^2 [\frac{\text{rad}^2}{\text{s}^2}]$
$F_{y,1}$	16.2959	-110.6691	86.0759
$F_{y,2}$	-22.0473	149.7288	-116.4557

Table A.3: Maximum and minimum allowed values of the variables related to F_y

The values in Table A.1 are of opposite sign. Therefore, when a variable increases $F_{x,1}$ or $F_{y,1}$ it decreases $F_{x,2}$ or $F_{y,2}$ by the same amount. For each force the variables are either limited at a maximum value or a minimum value while the opposite holds for the opposing force. Combined the maximum and minimum values of each variable are found.

-	$\frac{c_g}{c_{\ddot{r}}}, \ddot{x}_{\text{mrp}} [\frac{\text{m}}{\text{s}^2}]$	$\frac{c_g}{c_{\ddot{r}}}, \ddot{y}_{\text{mrp}} [\frac{\text{m}}{\text{s}^2}]$	$\frac{c_g}{c_{\ddot{\theta}}}, \ddot{\theta}_{\text{mrp}} [\frac{\text{rad}}{\text{s}^2}]$	$\frac{c_g}{c_{\dot{\theta}}}, \dot{\theta}_{\text{mrp}}^2 [\frac{\text{rad}^2}{\text{s}^2}]$
Maximum	12.6746	16.2959	149.7288	86.0759
Minimum	-17.1479	-22.0473	-110.6691	-116.4557

Table A.4: Maximum and minimum allowed values of the variables

The safest situation is the one where the actuation limits are formulated such that the variables can all operate at their limit and the forces are still all positive. This can be done by requiring each variable to not exceed a certain fraction $k_{c_{\ddot{r}}}$, $k_{c_{\ddot{\theta}}}$ and $k_{c_{\dot{\theta}}}$ of their limits.

$$F = k_{c_{\ddot{r}}}\ddot{r}_{\text{mrp}}c_{\ddot{r}} + k_{c_{\ddot{\theta}}}\ddot{\theta}_{\text{mrp}}c_{\ddot{\theta}} + k_{c_{\dot{\theta}}}\dot{\theta}_{\text{mrp}}^2c_{\dot{\theta}} + c_g > 0 \quad (\text{A.7})$$

The fractions should all be positive and sum to maximum 1, for instance each of them is $\frac{1}{3}$. The values of $c_{\ddot{r}}$, $c_{\ddot{\theta}}$ and $c_{\dot{\theta}}$ can be taken into account to formulate the fractions. The values of $c_{\ddot{r}}$ are relatively large in relation to $c_{\ddot{\theta}}$ and $c_{\dot{\theta}}$ due to the high weight of the MRP and low weight of the manipulator. The maximum and minimum allowed values of \ddot{r}_{mrp} are thus relatively small. By increasing $k_{c_{\ddot{r}}}$ in relation to $k_{c_{\ddot{\theta}}}$ and $k_{c_{\dot{\theta}}}$ the maximum allowed value of \ddot{r}_{mrp} is increased such that it lies closer to the other limits.

The values of $c_{\ddot{r}}$, $c_{\ddot{\theta}}$ and $c_{\dot{\theta}}$ are either positive or negative. This means that the contribution of the corresponding variable to F_x or F_y is either positive or negative. When a normal force is increased by one or more of the variables the remaining variables are allowed to exceed the limits given by the Tables A.2 and A.3. This approach can cause for problems when the accelerations suddenly change direction.

During the manipulator motion the accelerations are required to be exceptionally high to induce toppling and thus the risk of toppling is negligible. The acceleration of the end effector has to be in the range of $40 [\frac{\text{m}}{\text{s}^2}]$ where the angular acceleration of the joints is in the range of $200 [\frac{\text{rad}}{\text{s}^2}]$. Obviously the values are largely dependent on the configuration of the MRP and the manipulator but the order of magnitude remains similar. For an energy optimization, where accelerations are generally minimized, these exceptional accelerations are not attained.

Appendix B

Parameters

-	$L_{x,i}$ [m]	$L_{y,i}$ [m]
1	0.7	0.9
2	0.035	0.045
3	0.35	0.45

Table B.1: MRP dimensions as in Figure 3.3

The height $L_4 = 0.3$ [m] and mass $m_{\text{mrp}} = 30$ [kg]. The mass is assumed to be distributed homogeneous and thus the center of mass is located at the exact center of the rectangle describing the MRP. The mass moment of inertia is defined by

$$\begin{aligned}
 I_{\text{mrp},xx} &= \frac{1}{12} m_{\text{mrp}} (L_4^2 + L_{y,1}^2), \\
 I_{\text{mrp},yy} &= \frac{1}{12} m_{\text{mrp}} (L_{x,1}^2 + L_4^2), \\
 I_{\text{mrp},zz} &= \frac{1}{12} m_{\text{mrp}} (L_{x,1}^2 + L_{y,1}^2).
 \end{aligned} \tag{B.1}$$

-	$L_{m,i}$ [m]	$r_{m,i}$ [m]	m_i [kg]
1	0.1	.025	1
2	0.25	.025	2
3	0.25	.025	2
4	0.2	.025	1

Table B.2: The manipulator dimensions and masses

The dimensions $r_{m,i}$ refer to the radii of the links. The mass moment of inertia of each link is defined by

$$\begin{aligned}
 I_{i,xx} &= \frac{1}{12} m_i (L_{m,i}^2 + r_{m,i}^2), \\
 I_{i,yy} &= \frac{1}{12} m_i (L_{m,i}^2 + r_{m,i}^2), \\
 I_{i,zz} &= \frac{1}{2} m_i r_{m,i}^2.
 \end{aligned} \tag{B.2}$$

-	$q_{m,i}^+$ [deg]	$q_{m,i}^-$ [deg]
1	120	-120
2	15	-160
3	250	55
4	180	-120
5	91	-91
6	180	-180

Table B.3: The joint limits of the manipulator

The optimization variables of the first algorithm are sampled by 501 samples. In the first algorithm the position and orientation is defined by a minimum and maximum. The optimization variable is chosen between 0 and 1, ranging from the minimum to the maximum value. The paths are generated by a variation of the polynomial with a value between o_i^+ and $o_i^- = -o_i^+$, where $o_i^+ = 20$.

Per iteration 15 solutions are generated and the 4 best solutions are saved for the next generation. The solutions per iteration are as follows:

- one solution identical to the previous best solution
- four solutions with a small variation of the best solution
- two solutions with a small variation of the second best solution
- one solution considering the best configurations and paths as straight line
- one solution combining the configurations of the best solution with paths of the best segments
- one solution combining the configurations of the best solution with paths of the best segments with a small variation
- one solutions combining the best solutions per segment
- two solutions combining the best solution per segment with a small variation of the path
- two solutions combining the best configuration per segment with a random path

In the second algorithm the optimization variables defining the MRP path are copied from the first algorithm. The manipulator paths are sampled by 10001 samples. Consider a path defined by a fourth order polynomial with $\hat{q} = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$. The path is defined by s ranging from 0 to 1. By defining p_5 of the polynomial description a maximum deviation of the position is attained which is scaled by 0.0625. Therefore, the range of values for the joint path is defined by $o_i^+ = \frac{q_{m,i}^+ - q_{m,i}^-}{0.0625}$ and $o_i^- = -o_i^+$. As a result of the limits all possible paths as function of a fourth order polynomial can be described. During optimization $i_{o_i}^+$ is reduced if $q_{m,i}^+$ is exceeded and $i_{o_i}^-$ is increased if $q_{m,i}^-$ is exceeded. This means if a limit is exceeded a joint path is not generated in later iterations with a value which exceeds the limit equally or further.

The function $s(t)$ is defined by 501 samples with the range defined in Section 4.4.3.

In the second algorithm 7 solutions are generated per iteration and the 4 best solutions are saved for the next iteration. The solutions correspond to the best solution of the previous iteration and 2 solutions each for a small variation of the best solution and second best solution and a mutation of the best solutions.

Bibliography

- [1] A. Bharadwaj and M. A. Dvorkin, “The rise of automation: How robots may impact the u.s. labor market.” <https://www.stlouisfed.org/publications/regional-economist/second-quarter-2019/rise-automation-robots>. (accessed: 08.09.2020).
- [2] C. Ramer, S. Reitelshofer, and J. Franke, “A robot motion planner for 6-dof industrial robots based on the cell decomposition of the workspace,” *Ieee Isr 2013*, 2013.
- [3] C. F. Wu, S. K. Lin, J. Lambrecht, T. D. Nguyen, A. Vick, M. Kleinsorge, and J. Kruger, “Integrated object and path demonstration for industrial robots in adaptive handling applications,” *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014.
- [4] J. Blankendaal and H. Kiela, “Fieldlab flexible manufacturing.” <https://www.brainportindustries.com/en/technology/fieldlab-flexible-manufacturing>. (accessed: 23.06.2020).
- [5] F. J. Abu-Dakka, I. F. Assad, R. M. Alkhdour, and M. Abderahim, “Statistical evaluation of an evolutionary algorithm for minimum time trajectory planning problem for industrial robots,” *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 1-4, p. 389–406, 2016.
- [6] A. Duttdubey, R. B. Mishra, and A. K. Jha, “Task time optimization of a robot manipulator using artificial neural network and genetic algorithm,” *International Journal of Computer Applications*, vol. 51, no. 13, p. 26–33, 2012.
- [7] P. Zacharia and N. Aspragathos, “Optimization of industrial manipulators cycle time based on genetic algorithms,” *2nd IEEE International Conference on Industrial Informatics, 2004. INDIN 04. 2004*.
- [8] K. Baizid, A. Yousnadj, A. Meddahi, R. Chellali, and J. Iqbal, “Time scheduling and optimization of industrial robotized tasks based on genetic algorithms,” *Robotics and Computer-Integrated Manufacturing*, vol. 34, p. 140–150, 2015.
- [9] F. L. Chernousko, *Optimization in Control of Robots*, pp. 19–28. Basel: Birkhäuser Basel, 1994.
- [10] M. Ratiu and M. A. Prichici, “Industrial robot trajectory optimization- a review,” *MATEC Web of Conferences*, vol. 126, 2017.
- [11] C. Hansen, J. Öltjen, D. Meike, and T. Ortmaier, “Enhanced approach for energy-efficient trajectory generation of industrial robots,” *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, vol. 8, Aug 2012.
- [12] M. Bagheri and P. Naseradinmousavi, “Novel analytical and experimental trajectory optimization of a 7-dof baxter robot: global design sensitivity and step size analyses,” *The International Journal of Advanced Manufacturing Technology*, vol. 93, no. 9-12, p. 4153–4167, 2017.
- [13] D. Meike and L. Ribickis, “Energy efficient use of robotics in the automobile industry,” *15th International Conference on Advanced Robotics (ICAR)*, 2011.
- [14] D. M. Winzenick, “Electric drives – energy-efficiency is increasingly regulated.” <https://www.zvei.org/en/association/divisions/automation-division/electric-drive-systems/electric-drives-energy-efficiency-is-increasingly-regulated/>. (accessed: 23.06.2020).

- [15] T. Javied, T. Rackow, R. Stankalla, C. Sterk, and J. Franke, "A study on electric energy consumption of manufacturing companies in the german industry with the focus on electric drives," *Procedia CIRP*, vol. 41, p. 318–322, 2016.
- [16] P. Huang, J. Yan, Y. Xu, W. Xu, and B. Liang, "Genetic algorithms-based minimum torque path planning for space manipulator," *6th World Congress on Intelligent Control and Automation*, 2006.
- [17] W. Perez Bailon, E. Barrera Cardiel, I. Juarez-Campos, and A. Ramos-Paz, "Mechanical energy optimization in trajectory planning for six dof robot manipulators based on eighth-degree polynomial functions and a genetic algorithm," *IEEE Internet Computing - INTERNET*, pp. 446–451, 09 2010.
- [18] R. Saravanan, S. Ramabalan, and C. Balamurugan, "Evolutionary optimal trajectory planning for industrial robot with payload constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 38, no. 11-12, p. 1213–1226, 2007.
- [19] J. Sandoval, P. Vieyres, and G. Poisson, "Generalized framework for control of redundant manipulators in robot-assisted minimally invasive surgery," *IRBM*, vol. 39, no. 3, p. 160–166, 2018.
- [20] J. Wang, Y. Li, and X. Zhao, "Inverse kinematics and control of a 7-dof redundant manipulator based on the closed-loop algorithm," *International Journal of Advanced Robotic Systems*, vol. 7, no. 4, 2010.
- [21] C. A. Klein and C.-H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 3, p. 245–250, 1983.
- [22] S. Soylyu, B. J. Buckham, and R. P. Podhorodeski, "Redundancy resolution for underwater mobile manipulators," *Ocean Engineering*, vol. 37, no. 2-3, p. 325–343, 2010.
- [23] Q. Yu, G. Wang, X. Hua, S. Zhang, L. Song, J. Zhang, and K. Chen, "Base position optimization for mobile painting robot manipulators with multiple constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 54, p. 56–64, 2018.
- [24] S. Ren, Y. Xie, X. Yang, J. Xu, G. Wang, and K. Chen, "A method for optimizing the base position of mobile painting manipulators," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, p. 370–375, 2017.
- [25] G.-C. Vosniakos and E. Matsas, "Improving feasibility of robotic milling through robot placement optimisation," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 5, p. 517–525, 2010.
- [26] S. Mitsi, K.-D. Bouzakis, D. Sigris, and G. Mansour, "Determination of optimum robot base location considering discrete end-effector positions by means of hybrid genetic algorithm," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 1, p. 50–59, 2008.
- [27] A. Nektarios and N. A. Aspragathos, "Optimal location of a general position and orientation end-effectors path relative to manipulators base, considering velocity performance," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 2, p. 162–173, 2010.
- [28] T. Song, F. J. Xi, S. Guo, and Y. Lin, "Optimization of a mobile platform for a wheeled manipulator," *Journal of Mechanisms and Robotics*, vol. 8, Jun 2016.
- [29] S. Vafadar, A. Olabi, and M. S. Panahi, "Optimal motion planning of mobile manipulators with minimum number of platform movements," *2018 IEEE International Conference on Industrial Technology (ICIT)*, 2018.
- [30] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.
- [31] H.-Y. Zhang, W.-M. Lin, and A.-X. Chen, "Path planning for the mobile robot: A review," *Symmetry*, vol. 10, no. 10, p. 450, 2018.
- [32] M. Zhao, N. Ansari, and E. Hou, "Mobile manipulator path planning by a genetic algorithm," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [33] C.-J. Chen and C.-S. Tseng, "The path and location planning of workpieces by genetic algorithms," *Journal of Intelligent Manufacturing*, vol. 7, no. 1, p. 69–76, 1996.

- [34] P. Huang, J. Yan, Y. Xu, W. Xu, and B. Liang, "Genetic algorithms-based minimum torque path planning for space manipulator," *6th World Congress on Intelligent Control and Automation*, Jun 2006.
- [35] W. P. Bailon, E. B. Cardiel, I. J. Campos, and A. R. Paz, "Mechanical energy optimization in trajectory planning for six dof robot manipulators based on eighth-degree polynomial functions and a genetic algorithm," *2010 7th International Conference on Electrical Engineering Computing Science and Automatic Control*, Sep 2010.
- [36] R. Ziwu, L. Chunguang, and S. Lining, "Minimum-acceleration trajectory optimization for humanoid manipulator based on differential evolution," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 73, 2016.
- [37] D. P. Garg and M. Kumar, "Optimization techniques applied to multiple manipulators for path planning and torque minimization," *Engineering Applications of Artificial Intelligence*, vol. 15, p. 241–252, Jun 2002.
- [38] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley, 2006.
- [39] R.-F. Fung and Y.-H. Cheng, "Trajectory planning based on minimum absolute input energy for an lcd glass-handling robot," *Applied Mathematical Modelling*, vol. 38, no. 11-12, p. 2837–2847, 2014.
- [40] C.-C. Tsai, H.-C. Huang, and C.-K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, p. 4813–4821, 2011.
- [41] D. Berenson, J. Kuffner, and H. Choset, "An optimization approach to planning for mobile manipulation," *2008 IEEE International Conference on Robotics and Automation*, 2008.
- [42] S. M. LaValle, "Rapidly-exploring random trees : a new tool for path planning," 1998.
- [43] S. M. Lavalle and J. J. Kuffner, "Randomized kinodynamic planning," *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*.
- [44] A. Koubaa, H. Bennaceur, I. Chaari, S. Trigui, A. Ammar, M.-F. Sriti, M. Alajlan, O. Cheikhrouhou, and Y. Javed, *Introduction to Mobile Robot Path Planning*, pp. 3–12. Cham: Springer International Publishing, 2018.