Online Path Planing for Industrial Robots With Integrated Workspace Limits and Safety Criterion

Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik
der Universität Stuttgart zur Erlangung
der Würde eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Abhandlung

Vorgelegt von
Dipl.-Ing. Akos Csiszar
aus Satu Mare

Hauptberichter:	Prof. Dr.-Ing. Alexander Verl
Mitberichter:	Prof. Dr.-Ing. Mathias Liewald
	Prof. Dr.-Ing. Heinz Wörn
Tag der mündlichen Prüfung:	02.11.2016

Institut für Steuerungstechnik der Werkzeugmaschinen und
Fertigungseinrichtungen der Universität Stuttgart

2017

Acknowledgement

I would like to express my gratitude to Prof. Dr.-Ing. Cornel Brisan and to Prof. Dr.-Ing. Alexander Verl for taking me on as their PhD student and for being outstanding advisors. Their constant encouragement, support, and suggestions made this work successful.

My sincere thanks go to my colleagues at the institutions I had the honor to work with. Colleagues at the Technical University of Cluj Napoca, especially to colleagues at the Department of Mechatronics and Machine Dynamics, colleagues at the ISW, University of Stuttgart, especially colleagues in Group 2, colleagues at the Fraunhofer IPA, especially Group 322 of the Robot Systems Department, I am very grateful for your help and support! Furthermore, would like to express thanks to the DAAD (Deutsche Akademische Austausch Dienst) for honor that a DAAD fellowship means. Furthermore, I would like to thank also to the implementation team of the SiDoc project for their support during this work.

Moreover, I am deeply indebted to my family and friends for their support and encouragement.

This work was partially financed by the DAAD and by by the project "Doctoral studies in engineering sciences for developing the knowledge based society-SIDOC" contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013.

The Author

Short Summary

The demands on modern robot systems are ever increasing. Reaction to changes in the environment of robots, i.e. collision avoidance, especially in physical human robot interaction (pHRI), is becoming more and more important. This is, among others, due to the real-time requirements of roboti controllers not trivial. This work is dedicated to the the real-time capability of the path-planning algorithms by the use of a reactive path-planning method. As a solution, the potential field methos (APF) is proposed. The APF method has the desired reactive character and it is real-time capable. The published APF methods, however, have deficits regarding their application for industrial robots and for pHRI. In the course of this work, a new APF path-planner for industrial robots has been developed, suitable for human robot interaction. The developed path-planner takes into account the workspace limitations of the robot arms and safety criteria for pHRI.

In the first chapter, the general aspects of the work, such as motivation, problems and goals of the work are described. Also in this chapter is the work delimited. Then, in chapter 2, the relevant state of the art is presented. Existing methods for the definition of the workspace, path planning and published safety criteria for human robot interaction are analyzed and evaluated. In chapter 3, a comprehensive presentation of the already published aspects of the APF method, which will serve as a direct basis for the contributions of the dissertation, are be presented. At the end of the chapter the advantages and disadvantages of the method are explicitly listed. Here is also the motivation behind selecting this particular method for research. The contributions to the state of the art are described in detail in Chapter 4. These contributions address the problem defined in Chapter 1 and the open scientific questions in

Chapter 2. A modular approach for calculating the workspace of a robot is proposed. A method for integrating different types of workspace limits in the APF method has been developed. A novel safety criterion has been also developed. This novel safety criterion can guarantee the safety of a person involved in the interaction even in the case of a fault occurs in the control system of the robot. A method for integrating the safety criterion in the APF algorithm has also been developed.

In Chapter 5, the results of the research are presented. In particular, the real-time capability and fault tolerance of the developed path-planning method proves to be attractive for the field of industrial robots. The final chapter, Chapter 6, summarizes the important results. The work is concluded with the comments on possible future developments of the industrial application of the research results.

Kurzinhalt

Die Ansprüche an moderne Robotersysteme steigen immer an. Reaktion auf Änderungen in der Umgebung der Roboter, also Kollisionsvermeidung, vor allem in der Mensch-Roboter Interaktion (MRI), wird immer wichtiger. Das ist unter anderem wegen der Echtzeitanforderung der Robotersteuerungen nicht trivial. Diese Arbeit widmet sich der Echtzeitfähigkeit des Bahnplanungsalgorithmus durch den Anwendung einer reaktiven Bahnplanungsmethode. Als Lösungsansatz wurde die Feldpotentialmethode (APF) vorgeschlagen. Die APF-Methode besitzt den erwünschten reaktiven Charakter und ist echtzeitfähig. Die veröffentlichten APF-Methoden weisen jedoch Defizite auf in Bezug auf die Anwendung für Industrieroboter und für MRI. Im Rahmen diser Arbeit wurde ein neuer APF-Bahnplaner für Industrieroboter entwickelt, geeignet für die Mensch-Roboter Interaktion. Der entwickelte Bahnplaner berücksichtigt die Arbeitsraumbegrenzungen der Roboterarme und die Sicherheitskriterien für MRI.

Im ersten Kapitel sind die allgemeinen Aspekte der Arbeit beschrieben, wie Motivation, Problemstellung und Ziele der Arbeit. Hier erfolgt auch die Abgrenzung der Arbeit. Danach, im 2. Kapitel wird der forschungsrelevante Stand der Technik vorgestellt. Analysiert und bewertet werden existierende Methoden für die Bestimmung des Arbeitsraumes, die Bahnplanung und veröffentlichte Sicherheitskriterien für Mensch-Roboter Interaktion. Im 3. Kapitel erfolgt eine umfassende Vorstellung der bereits veröffentlichten Aspekte der APF-Methode, die als direkte Basis für den wissenschaftlichen Beitrag der Dissertation dienen wird. Zum Schluss des Kapitels werden ausdrücklich die Vor- und Nachteile der Methode benennt. An dieser Stelle findet sich ferner eine Begründung, weshalb diese besondere Methode für die Forschung ausgewählt wurde.

Die Beiträge zum Stand der Technik werden im Einzelnen im 4. Kapitel dargestellt. Diese Beiträge wenden sich der Problemstellung aus dem 1. Kapitel und den offenen wissenschaftlichen Fragen aus dem 2. Kapitel zu. Es wird ein modularer Ansatz zur Berechnung des Arbeitsraums eines Roboters vorgeschlagen. Ein Verfahren, wie unterschiedliche Arten von Arbeitsraumgrenzen in das APF-Verfahren integriert werden können, wurde entwickelt. Des Weiteren wurde ein neues Sicherheitskriterium erforscht. Dieses neue Sicherheitskriterium kann die Sicherheit eines an der Interaktion beteiligten Menschen auch dann garantieren, wenn ein Fehler in der Steuerung des Roboters auftritt. Ein Verfahren für die Integration des Sicherheitskriteriums in dem APF-Algorithmus wurde auch entwickelt.

Im 5. Kapitel werdem die Ergebnisse der Forschungen vogestellt. Insbesondere die Echtzeitfähigkeit und Fehlertoleranz der entwickelten Bahnplanung-Methode erweist sich als äußerst attraktiv für den Bereich der Industrieroboter. Das Schlusskapitel, Kapitel 6, fasst die wesentlichen Ergebnisse zusammen. Mit den Anmerkungen zur möglichen Weiterentwicklung zum industriellen Ansatz der Forschungsergebnisse wird die Arbeit abgeschlossen.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Current Trends

The factory of the future is envisioned by Warren G. Bennis, a distinguished professor at the University of Southern California, as:

> "The factory of the future will have two employees: a man and a dog. The man's job will be to feed the dog. The dog's job will be to prevent the man from touching any of the automated equipment."

This prediction is obviously intended as a humorous summarization of current trends, but it carries two important real-life aspects. The factory of the future:

- will be autonomous to a high degree. The role of the human workers will not be decided by technological limits. Today the inside of the workspace of an industrial robot is mostly a forbidden zone for a human worker.
- will still have human employees, and their safety and bodily integrity must be assured (although the humorous phrase foresees that the dog will protect the automated equipment, it is inherently acting as a safety barrier between the human and the technological equipment). Today an often used safety barrier is a safety fence, separating the worker from the workspace of the robot physically.

This prediction of Warren G. Bennis also agrees with other predictions on the subject of humans occupying parts of the workspace of a robot, as presented in Garcia et al. 2007; Bischoff et al. 2010. Comparing the envisioned future state and the current state, differences can be identified. These differences can be formulated as a problems. The solution of these problems will lead to the envisioned future.

1.2 Problem Statement

In order to have robots react to the actions of humans, their software must permit modifications of their "behavior", based on changes in the environment. The environment of the robot refers to all the elements inside the robots workspace. Humans are also considered part of the environment of the robot. The motions of the humans or other obstacles inside the workspace are considered changes in the occupancy of the environment. For industrial robots their "behavior" is in fact closely related to the paths their tool center points (TCP) execute.

The currently valid robotic safety standard, ISO 10218:2011, if certain criteria are met, permit the sharing of the workspace between a robot and a human. The criteria set by the standard refer to avoiding collisions between the human and the robot under any circumstances.

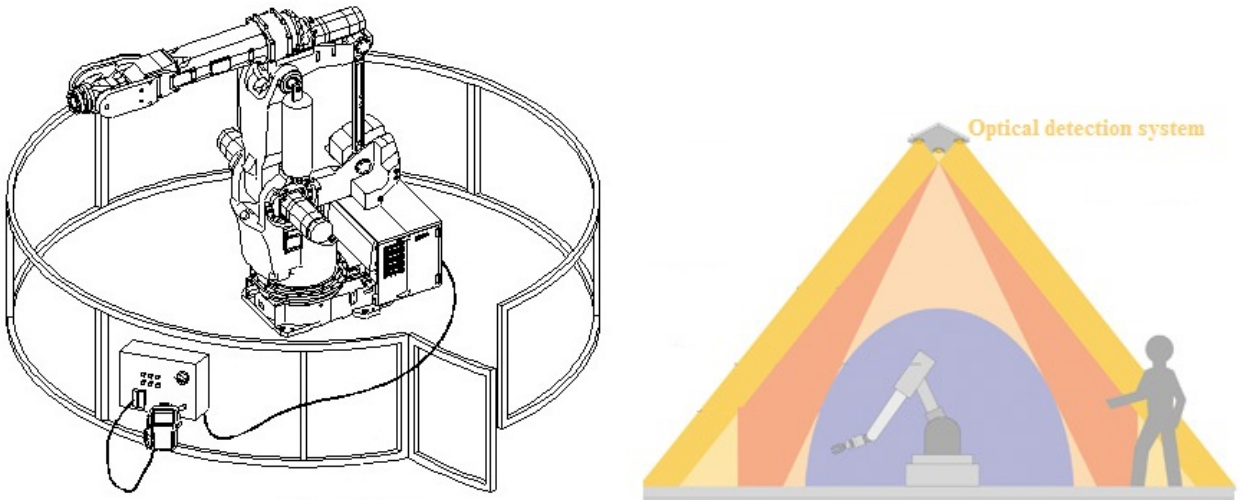
The current, offline approach in robot programming prohibits physical human robot interaction and collaboration. In the offline approach the paths, executed by the robot, are generated a priori. This way, these paths, cannot be altered while they are executed. This offline path generation of robots prohibits an interaction, the robots cannot respond to the actions of the human by altering their pre-generated paths. Currently, robots do not re-plan the path they are executing based on the motions of the human they are sharing a workspace with. The sharing of the workspace between a human and a robot is practiced only in a few, special scenarios, where the robot is stopped when the human is in its vicinity.

A safety barrier between the technological equipment and a human is necessary, as presented in Pedrocchi et al. 2009, to assure, that the human will not be harmed by the technical equipment under any circumstances. All technological equipment is prone to faults and malfunctions, either hardware or software. An often applied safety barrier is a safety fence, presented in figure 1.1a, used to separate areas where humans move freely, without restrictions, from the workspace of the robot. This approach restricts the coexistence of humans and machines in the same work area, and so prohibits interaction and collaboration

between the two. The optical barrier, presented in figure 1.1b is a close equivalent of the fence. In this case the workspace is more open, then in the case of the safety fence, however, interaction is very limited, since the robot has to stop when someone enters its workspace. A virtual barrier that permit interaction and collaboration, but also assures safety is desired.

A safety criterion can take the role of this virtual barrier. It has to be in the spirit of the valid safety standards, and has to be capable to assure the safety of the human, without restricting the interaction. Such a novel safety criterion can replace the currently used safety barriers and facilitate human robot interaction.

The path generation based on the safety criterion can be formulated as a constrained optimization problem. Given the requirements of the task, the workspace of the robot and other constraints (e.g. obstacles, humans, safety criterion, etc.) a path has to be found, which has to fulfill the task requirements, without violating the any constraints. These constraints can change during the execution of the path (i.e. humans and obstacles can move). This sets the requirement, that the optimization (i.e. the path planning) has to be carried out in real-time.



(a) Safety fence, photo source: ETS

(b) Optical safety barrier, photo source: Pilz

Figure 1.1: Currently used robotic safety barriers

1.3 Motivation

Factory equipment capable of coping with unforeseen situations can improve aspects of production, which usually requires human intervention. For an industrial manipulator these unforeseen scenarios are in fact unpredictable changes in the occupancy of the workspace. When using online path planning with collision avoidance the path planner algorithm can make changes to the task execution based on an optimality criterion (i.e. the cost function). Decisions based on an optimality criterion can be formulated as an optimization problem. In the case of a path planning algorithm the result of this decision making is the path of the TCP. More explicitly the result are the reference coordinates sent to the robot controller.

The advantages of such an approach are:

- Production systems making decisions in unforeseen situations contribute to an efficient production process.

- Humans and robots sharing the same workspace safely can solve tasks more efficiently then in the case of humans working independent of each other.

Cohabitation of humans and robots is not only important to the production industry. Other applications are also possible (e.g. robots helping in household environments can improve quality of life). This research is focused on the industrial environment and in particularly on how to avoid collision inside the workspace pf a robot.

1.4 Goals

A valid path is a path that can be executed by the robot. For this, first of all, the generated path has to lay completely inside the workspace of the robot.

Online path planning refers to computing paths in real-time, as they are executed, with a cycle time of $1 - 20ms$, as customary for motion control

applications. Generating the path this way will ensure, that the robot can react to changes in the environment in a timely manner.

A valid path refers to a path that can be executed by the robot, without violating any internal (e.g. workspace limits) or external (e.g obstacles) constraints.

A safe path is defined as a path which is executed by the robot, and during this the execution of the path the bodily integrity of a human inside the workspace of the robot is not put in danger. The violation of a safety criterion characterizes a path as not safe.

The main goal of this thesis is to develop an online path planning algorithm, with an integrated safety criterion, that is capable of planning valid, safe paths in real-time. This main goal can be divided into subgoals based on functional correlation. The fulfillment of the each subgoal is a prerequisite for the fulfillment of the next subgoal:

- Analysis the motion limits of the robot (i.e. workspace limits)
- Development of a real-time capable path planning algorithm for generating valid paths
- Development of a novel safety criterion and integrate it in the path planning algorithm

The workspace of the robot represents the search space for the path planning algorithm. Developing a novel safety criterion independently from the path planner algorithm and integrating it afterwards in the path planner, has at least two benefits:

- The safety criterion, since it is independent of the developed planning algorithm, can be integrated into other planning algorithms.
- The way, how the independently developed safety criterion is integrated in the developed planning algorithm, can be of guidance, if a different criterion is desired to be integrated.

Research during the development of this thesis was focused on meeting the above mentioned goals.

1.5 Contributions

The contribution of this thesis addresses two aspects: how to plan a path, and what makes a path safe. These contributions are closely related to the goal of the thesis. The contributions can be summarized as follows:

- new insights on the structure of the workspace of a manipulator
- extension of an existing path planning method,
- development of a novel safety criterion,
- integration of the safety criterion to the path planning algorithm

In this thesis the functionality of an already published real-time capable path planning method, with obstacle avoidance, is extended. The extension of the method makes it better suited for applications in the domain of pHRI (physical Human Robot Interaction) using robotic manipulators. The extension refers in particular to the integration of all the workspace constraints. Furthermore, a novel safety criterion is defined and integrated in the mentioned path planning algorithm. This way, the developed planning algorithm plans paths, in real-time, based on this novel safety criterion. These aspects are also graphically presented in figure 1.2. Although there are many other aspects of research involving human robot interactions, this work is limited to the above mentioned ones.

1.6 Delimitation of the Work

Research regarding pHRI requires expertise in different domains of science and technology. This way it is a truly interdisciplinary domain. It unites many major disciplines like robotics, signal processing, simulation technology, communication technology and others.

This work is focused on assuring safety inside the workspace of a manipulator by planning path based on a safety criterion. In order to achieve the defined

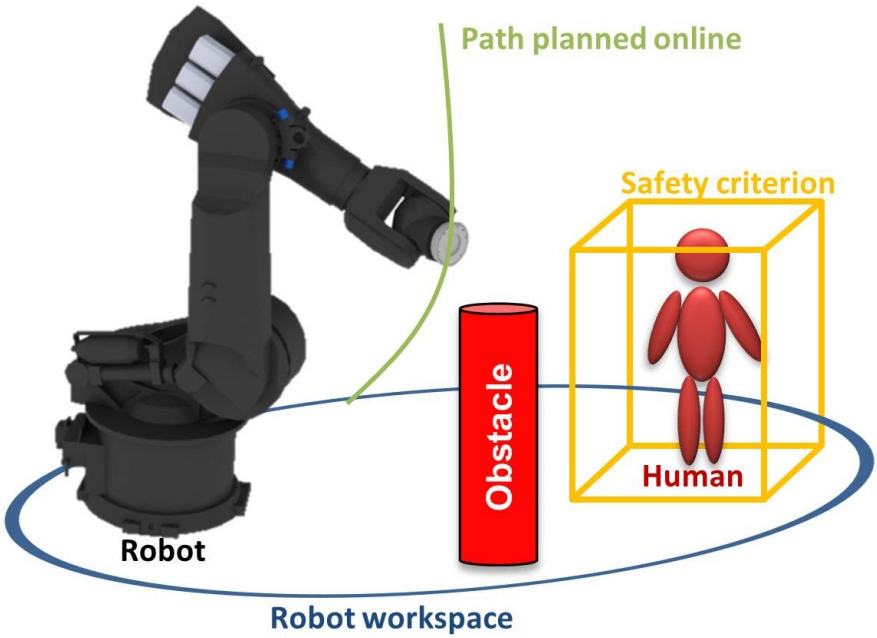


Figure 1.2: Graphical representation of thesis contributions

goals, two simplifications are considered. These simplifications refer to complex components, that are required as an input to this work. It is considered, that:

- The environment of the robot is known, and information about the current state of the environment is available in real-time. No predictions of future state are required, only the current state is considered as input. Research concerning sensor systems used in human-robot cooperation is presented in Gecks et al. 2005; De Santis et al. 2007; Ebert et al. 2002.

- Information about the braking characteristics of the robot are considered known and available in real time. Research concerning the braking characteristics of robots is presented in Dietz et al. 2010; Dietz et al. 2011.

These two assumed simplification, delimit this work to the domain of robotic path planning.

1.7 Thesis Structure

The structure of the thesis is defined having in mind the easy understanding of the presented concepts and the clear delimitation between contributions of this

research and the state of the art. A structure of this thesis is presented visually in figure 1.3.

The first chapter, Introduction presents the general aspects of the work, its goal and its motivation. It offers a brief overview of the contents and structure of the thesis.

The second chapter, entitled State of The Art, presents the state of the art of the different subdomains of robotics, to which contributions are made in this thesis. First of all the aspects concerning the workspace of a robot are presented. From path planning and safety point of view the workspace of a robot defines "the space of interest". Paths cannot be planned outside this workspace, humans cannot be harmed if are outside the workspace. Publications dealing with path planing and safety aspects are also presented in this chapter. A global overview of these subdomains is presented, at a conceptual level. Scientific open problems are identified and presented at the end of the chapter.

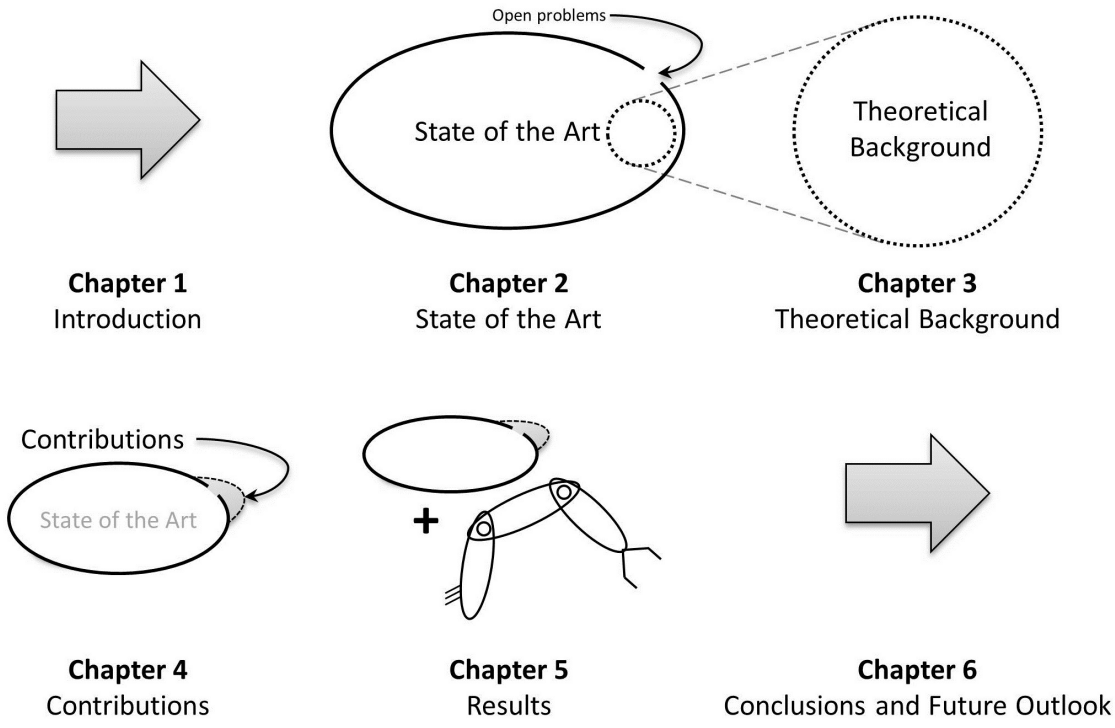


Figure 1.3: Graphical presentation of the thesis structure

An in depth presentation of already published methods is done in Chapter 3, entitled Theoretical Background. Only those theoretical aspects are described, which are considered important, either because the contributions of the thesis are built upon these, or they present limitations, which the contributions of this thesis improve. Some basic aspects of robotics, which are related to the thesis, are also described briefly in this chapter. Also, at the end of the chapter, open problems of the presented methods are presented.

Contributions of this thesis are presented in detail in Chapter 4 Contributions. The contributions address the problem statement in Chapter 1 Introduction and the open scientific problems identified in chapter 2, State of The Art. A solution to these open problems is proposed. The workspace limits of the robot have been identified. A correlation between the motion limits of the components in the structure of the robot and the workspace limits has been presented. Workspace limits have been integrated in the planning algorithm. More types of workspace limits have been considered than in the state of the art. Furthermore, a novel safety criterion has been developed and integrated in the path planning algorithm. This novel criterion can prevent collision also in the case of a malfunction of the robot. This quality cannot be found in the state of the art. The proposed solutions are presented in detail, emphasizing both their advantages and disadvantages. Some aspects of the possibility of industrial deployment are also described.

Results of the research, as well as how these results have been obtained are presented in Chapter 5 Numerical and Experimental Results. Both numerical and experimental results are shown. Numerical results have been obtained from numerical simulations. These validate the contributions in a simulated, virtual environment. Experimental results are shown to offer an experimental validation of the contributions. The architecture of the experimental setups used are described. The experimental results have been obtained using industrial hardware. Conclusions of the research and future prospects, how the research can be continued, are presented in the last chapter, Chapter 6 Conclusions and Future Outlook.

2 State of The Art

2.1 The Workspace of a Robot

Workspaces of different types are defined in the scientific literature e.g. reachable workspace, constant orientation workspace, inclusive orientation workspace, dexterous workspace, etc. Merlet 2002; Kanaan et al. 2006. The reachable workspace is the set of points that the TCP of the robot can reach, without any additional constraints (e.g. constant orientation of the TCP, etc.). Throughout this thesis the notion workspace refers to the reachable workspace of the robot, unless otherwise specified.

The limits of the workspace represent the barrier between poses the TCP can and cannot attain, based on the mechanical structure of the robot. The most used technique for generating the workspace of a mechanism is search by sampling. This is based on the discretization of the search space and point by point verification. This technique has the advantage of given a large enough search space the whole workspace is always found, but has the disadvantage of a high computational burden. Wave propagation based technique, in Macho et al. 2009, has a smaller computational burden when compared to sampling, but it can only find one integral part of a workspace. CAD geometry variation technique, presented in Lu et al. 2008b; Lu et al. 2008a, uses commercial CAD programs to find the reachable workspace based on the direct kinematics, solved by the CAD program, not by an implemented algorithm (e.g. numerical solving of nonlinear equations for the case of parallel robots).

The workspace is usually computed as a volume in Cartesian space. The surface of this volume represents the workspace limits.

From the point of view of a path planning algorithms, these limits and their integration in the path planning algorithm are important.

For global planning algorithms, the integration of these limits is straight forward. Their mathematical formulation has to be considered as a constraint for the optimization method. For local methods, on the other hand, their integration is more complex. The way these limits have to be integrated is strongly dependent on how these limits are defined. In order to integrate workspace limits in local path planning algorithms it is important not to only obtain the workspace as a volume and its limits as a surface, but also to define what causes these limits.

2.2 Obstacles

Any object inside the workspace of the robot is considered an obstacle. Obviously, collisions between the robot and obstacles should be avoided. In other words, obstacles define areas in the Cartesian workspace, which must not be occupied by the robot. Path planning with collision avoidance refers to planning paths which avoid collisions with the obstacles inside the workspace.

The most convenient way to characterize a robot, from the point of view of path planning algorithms, is with one point, namely the TCP. Unfortunately, this is inconvenient from the collision avoidance point of view, since in this case the structure of the robot could collide with the obstacles. To overcome this, the configuration space was introduced. In configuration space the robot is described by one point only, usually the TCP, and the size of the obstacles are enlarged. This enlargement of the obstacles is strongly correlated with the robot structure. Basically, the information about the geometry of the robot structure is added to the obstacle, not to the TCP. This way it can be guaranteed, that even though the robot is described by only one point, no collision between obstacles and the robot structure can occur, according to Lozano-Perez 1983. Configuration space does not represent a path planning method, but a manifold which represents the search space for path planning algorithms. It is

used in many cases to plan collision free paths, as presented in LaValle 2006. Unfortunately it is computationally demanding to generate the configuration space at every time step (or whenever obstacles or the robot moves), and so it is not well suited for real-time applications.

2.2.1 Human Obstacle Model

Modeling the human body as an obstacle in pHRI applications is also of interest. The most used method is defining a bounding volume around the human body. In Kulic et al. 2005 the shape of the human body is approximated by spheres, in Najmaei et al. 2009 superqudratic functions are used. In Chen et al. 2009 just the palms of the hand are modeled as point obstacles. A model of the human body with parametrized level of detail is presented in Najmaei et al. 2011. This parametrized model is presented in figure 2.1.

All these models aim to represent the human body with a certain level of detail. There are no safety considerations integrated in these models.

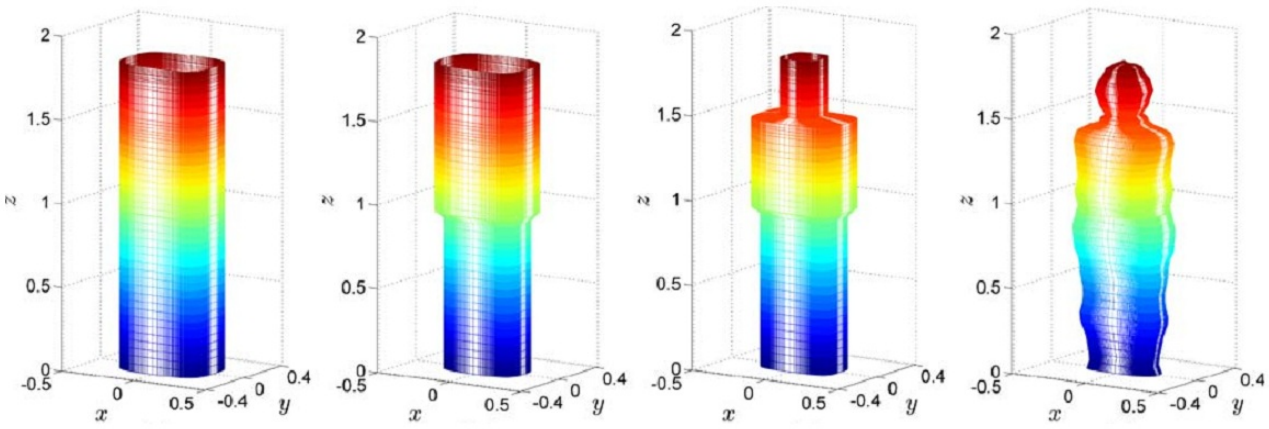


Figure 2.1: Model of the human body with variable level of detail, as presented in Najmaei et al. 2011, ©IEEE 2011

2.3 Assuring Safety Inside the Workspace

An important aspect of pHRI is to assure the safety of a human inside the workspace of a robot. Assuring the safety of the human participant means the reduction of the risks of injury and the reduction of the severity of possible injuries. This can be achieved through many methods. In table 2.1 the different approaches of safety assurance are shown, as presented in Ikuta et al. 2003.

<table>
<tr><th></th><th></th><th>Control Strategy</th><th>Design Strategy</th></tr>
<tr><td rowspan="2">Pre-Collision</td><td>avoid collision</td><td>distance</td><td>-</td></tr>
<tr><td rowspan="2">minimize impact force</td><td>speed</td><td rowspan="2">weight</td></tr>
<tr><td>moment of inertia</td></tr>
<tr><td rowspan="4">Post-Collision</td><td rowspan="4">attenuation diffusion</td><td rowspan="4">stiffness</td><td>cover</td></tr>
<tr><td>surface</td></tr>
<tr><td>joint compliance</td></tr>
<tr><td>shape</td></tr>
</table>

Table 2.1: Approaches to assure human safety in pHRI, as presented in Ikuta et al. 2003

These approaches can be categorized by different criteria. Pre-collision strategies aim to reduce injuries before the collision occurs, post-collision strategies aim to reduce injuries after a collision occurred. A similar categorization can be also found in Heinzmann et al. 2003; Morita et al. 1999; Kulic et al. 2006. In order to better understand the difference between the two approaches, in Ikuta et al. 2003 an analogy from the automotive world is presented. The aim of the pre-collision strategies is similar to the aim of the automotive ABS (Anti-Blocking System) systems, while post-collision strategies can be compared to the automotive airbags.

- Pre-collision design approaches to safety:
  - There is no design strategy to avoid collisions. A robots mechanical architecture cannot avoid collisions by design.
  - Weight reduction of the robots helps to reduce the impact force De Santis et al. 2008; Haddadin et al. 2008.

- Post-collision design approaches to safety:
  - Covering the whole robot with an elastic material (e.g. rubber) helps to reduce injuries after a collision has occurred Yamada et al. 2005.
  - Friction of the surface of the robot is also a factor which can reduce injuries, after a collision has occurred. Low friction coefficient of the surface will cause a low friction force which is desired Yamada et al. 2005; Duchaine et al. 2009.
  - Passive compliance of the joints of the robot also help to minimize injuries after collision Ahmed et al. 2010; Tonietti et al. 2005.
  - Sharp edges of the linkages of the robot are not desired, since these can cause cuts and bruises Kim et al. 2008.

The design strategies have to be applied at the design stage. These strategies cannot be applied to already existent structures, this why they are not generally applicable. The control strategies, unlike design strategies, do not require modifications of the mechanical structure. These strategies does not have to be considered at the mechanical design stage. These affect only the control system of the robot. They can be included as a software component, or, in some cases, it is possible to attach them as an external control hardware and software component to an existing robot controller. A common disadvantage of all control strategy approaches is that these require complex external sensor systems to detect the environment.

- Pre-collision control strategies modify the robot behavior based on the state of the environment of the robot.
  - Avoiding collision between the robot and the human means in fact always keeping a distance between the two Liu et al. 2005.
  - Minimizing the impact force upon collision can be done by reducing the speed of the robot. The reduced speed leads to a reduced accumulated kinetic energy, which upon impact will lead to a reduced impact force Laffranchi et al. 2009.
  - The inertia is an attribute of the mechanical structure. Reducing inertial effects from the control system is based on the different values

of inertia of the multiple solutions of the inverse kinematic function, for a given TCP pose Kulic et al. 2004.

- Post-collision control strategies
  - The stiffness of a robot is also posture dependent, this way this approach is similar to control strategy reducing inertia, described above Kulic et al. 2004.

There is one approach that stands out: avoiding collisions between the human and the robot, by keeping a distance between them. The safety aspect behind this consideration is very straight forward. If no collision occurs, no injury can occur. Quantifying a safe distance as a threshold value is a complex task.

2.3.1 Safety Criteria

The safety criterion is the condition that assures that the robot will not harm the human participant during the interaction. It is strongly linked to the different approaches to safety presented above. It is important to consider the human as a special type of obstacle, protected by a safety criterion. While collisions with different objects in the workspace can only cause material damage, collision with a human can potentially cause injury or even death. In this section an analysis of the published safety criteria for pHRI are presented.

The key aspect of the analysis is how these safety factors perform in a worst-case scenario. The worst-case scenario is considered a scenario where a malfunction in the robot hardware or software occurs. It is expected that in case of such a malfunction the brakes of the robot are engaged.

In the simplest case this safety criterion is a safety clearance, a minimal distance around the human body that the robot cannot violate. A safety clearance of $2cm$ is suggested in Liu et al. 2005 in the case when a robotic manipulator with 6 degrees of freedom share their workspace with human co-workers. This $2cm$ threshold is a rather empirical limit. The MAROCO framework for human robot cooperation, presented in Graf et al. 2009, also suggests Euclidean distance as safety criterion, but no threshold value is specified. The $2cm$, or other empirical limit should be correlated with the braking capabilities of the robot,

since in case of a fault, the brakes of the robot are engaged. This is considered the worst-case scenario. Only a static safety clearance will not guarantee the avoidance of collision with the human if the braking capabilities are not correlated with the safety clearance.

In Svenstrup et al. 2009 and Svenstrup et al. 2010 this safety clearance approach has been extended and applied for mobile robots. Four zones have been defined around the human body, public, social, personal and intimate. The zones are not symmetric to the coronal plane of the body. This has been explained by the fact that humans are more comfortable with the situation where a robot is in front of them, and not behind them.

A potential field can be defined, that can be used to describe these zones. The equation describing this field, as presented in Svenstrup et al. 2010 and in equation (2.3.1), is a summation of four bi-variate Gaussian distributions.

$$g_2\left(\boldsymbol{x}_{1:2}\right) = \sum_{k=1}^{4} \exp\left(-\frac{1}{2}\left[\boldsymbol{x}_{1:2} - \mathbf{0}\right]^T \sum\nolimits_k^{-1} \left[\boldsymbol{x}_{1:2} - \mathbf{0}\right]\right) \tag{2.3.1}$$

Where c_k is a normalizing constant, $\boldsymbol{x}_{1:2}$ represents the relative distance vector between the human and the point where the function is evaluated (usually the robots position), $\mathbf{0}$ is the position of the person, in the presented case it is the origin, $\sum_k$ are the covariances of each Gaussian distributions. The potential field created by this potential function is presented visually in figure 2.2.

The delimitation of the mentioned four zones was done by using threshold values of the potential field. From the four zones, the latter two, personal and intimate, should not be violated by the mobile robot. The zones around the human body represent preference of the human. The avoidance of the mentioned zones will make the human feel safe, in the presence of a robot, but they do not present a technical limit for safety.

A general framework for path planing has been presented in Brock et al. 2002. The elastic strip framework can be used for motion planning in the case of human robot interaction, but no safety criterion other then distance has been mentioned in the article.

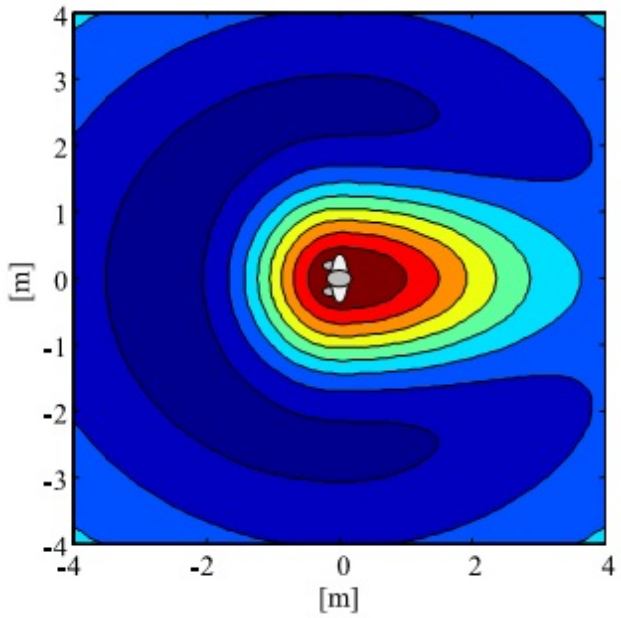


Figure 2.2: Zones around the human body, described by potential fields, as presented in Svenstrup et al. 2010, © IEEE 2009

A safety clearance based criteria has the advantage of being computationally simple, and it assures that there will be no collision between the robot and the human when everything is working without fault. However, in the case of a software or hardware fault the criterion cannot be assured anymore.

A more complex safety criteria can be found in Nokata et al. 2002. The safety criterion proposed in this paper was named Danger Index (DI), denoted by α. Its definition is force based, it is defined as the ratio between the current TCP force F and minimum impact force F_c that would cause harm to the human body upon impact. This is presented in equation 2.3.2.

$$\alpha = \frac{F}{F_c} \tag{2.3.2}$$

In the same publication, Nokata et al. 2002, another formulation of the DI is also proposed, which is an approximation of the force based definition, and is better suited for path planning application. The approximate formulation is proportional to the relative velocity v_h and the distance l.

$$\alpha = \frac{F}{F_c} = \frac{mav_h dt/l}{F_c} = k\frac{v_h}{l} \tag{2.3.3}$$

Where m represents the moving mass, a represents the acceleration and t represents time.

The DI is more complex than the safety clearance, because it quantifies the threat the robot imposes upon the human. Unlike safety clearance, DI allows collision with the robot if the impact force does not harm the human. In a situation where a fault causes the robot to brake, since through braking the robot cannot gain more energy, if a collision with the human would occur, it would not cause injury, because of the initial force limitation considered. However, a collision is not preferred, even if no injury occurs during the collision.

Similarly to the DI, the Kineto-Static Danger Field (KSDF), presented in Lacevic et al. 2010a, is also based on velocity. The KSDF has a potential field like formulation.

$$\mathrm{DF}\left(\boldsymbol{r}, \boldsymbol{r_t}, \boldsymbol{v_t}\right) = \frac{k_1}{\|\boldsymbol{r} - \boldsymbol{r_t}\|} + \frac{k_2 \|\boldsymbol{v_t}\| \left[\gamma + \cos \angle \left(\boldsymbol{r} - \boldsymbol{r_t}, \boldsymbol{v_t}\right)\right]}{\|\boldsymbol{r} - \boldsymbol{r_t}\|^2} \tag{2.3.4}$$

Where k_1, k_2 and $\gamma \geq 1$ are positive constants, $\boldsymbol{r}$ is a point in space at which the field is being computed, and $\boldsymbol{r_t}$ and $\boldsymbol{v_t}$ are position and velocity of the moving element.

Equation (2.3.4) is applied to the robot by integrating the points contained in the robot linkages (characteristic points). In Lacevic et al. 2010a the KSDF of a 2 DoF (Degrees of Freedom) planar manipulator is presented moving in a counter-clockwise direction. This is reproduced in figure 2.3.

In most cases the path of an industrial manipulator is known. This brings the possibility of prediction from the robot side. However the human counterpart in the interaction makes predictions difficult. In Najmaei et al. 2011 a predictive version of the DI has been suggested, called Cumulative Predictive Danger Index (CPDI). A neuronal network based estimator estimates the movement of the human. This way, both the path of the robot and the path of the human are considered known, with a given prediction horizon. This makes possible the calculation of the DI not just for the instantaneous state, but also for the predicted future. This approach considers more information than the methods above for estimating the level of threat the robot imposes. Using

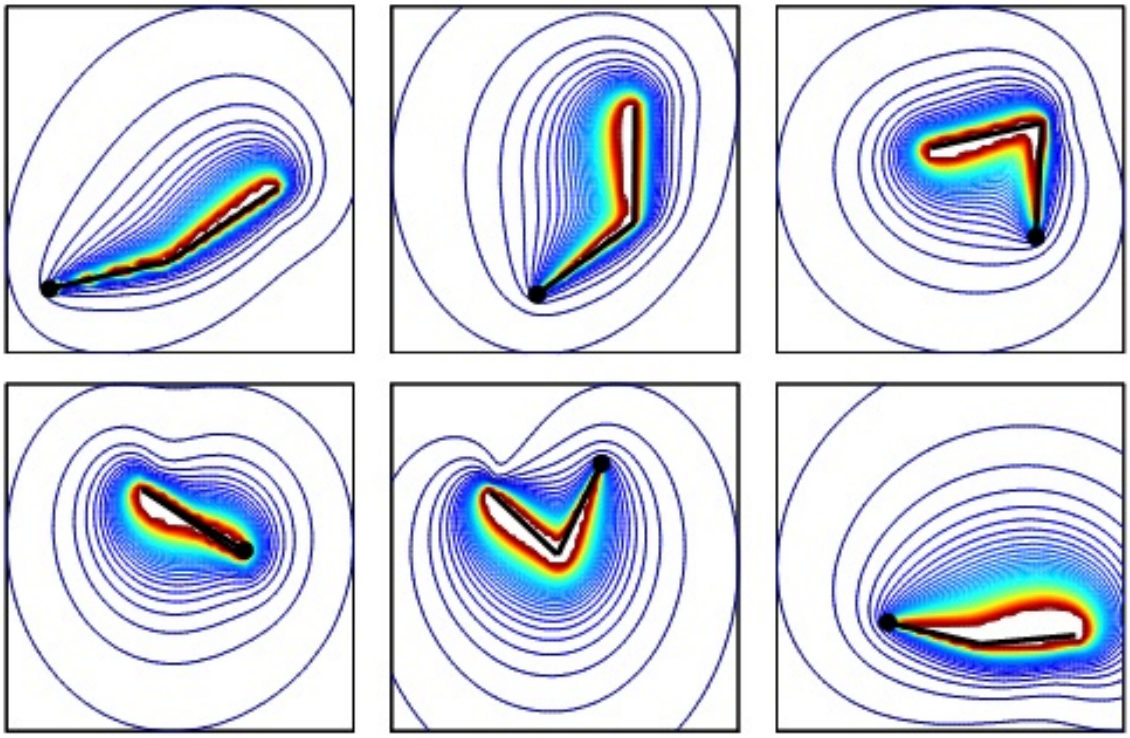


Figure 2.3: Kinetostatic Danger Field around a planar 2 DoF robot moving in a counter-clockwise direction, as presented in Lacevic et al. 2010a, © IEEE 2010

prediction, it characterizes the current and also the future states However, in the prediction only best case scenarios are considered. The human has a predicted trajectory and the robot has the programmed trajectory. The scenario, where the movement of the human is not conforming the prediction, is handled by the non-predictive DI, presented above. No worst case scenario is considered, where neither the human nor the robot follow the predicted or programmed trajectory. This in the case of the human means a distraction from its routine, which happens where easily. In the case of the robot, the execution of a trajectory which differs from the programmed one, is usually caused by a malfunction or fault. This case is not handled by the criterion.

The Head-Injury Criterion (HIC) was developed for crash testing applications in the automotive industry, described in Bicchi et al. 2004. It is calculated as

the integral of the acceleration of the center of mass of the head during a crash, as presented in Ogorodnikova 2009.

$$HIC = \Delta t \left(\frac{1}{\delta t} \int_{t_1}^{t_2} a_h dt \right)^{2.5}, \Delta t = t_2 - t_1 \tag{2.3.5}$$

Where Δt represents the duration of the impact which should be no longer then $15ms$, a_h represents the acceleration of the head during the crash.

A threshold value of 1000 for a frontal impact expresses the case when skull fractures begin to appear. Similar indexes to asses the severity of a crash are also used in the automotive industry (e.g. $3ms$-Criterion in Haddadin et al. 2008; Chawla et al. 2000). Although these are an effective measure of car crashes robotic path planning aims to avoid collisions.

Some of the above cited papers use complex safety criteria for path planning in human environments, but all of them consider that the robot will function properly. In order to assure safety for every scenario, the worst case scenario has to be considered.

2.3.2 Safety Standards

Safety standards are sets of rules that must be respected in industrial applications, having as role the assurance of the safety of humans by reducing the risk of injury and the severity of a possible injury below an acceptable threshold. The most relevant safety standard, from the path planning point of view is the ISO 10218:2011 Robots and robotic devices - Safety requirements for industrial robots ISO 10218:2011. However, other safety standards have to be respected also. According to Tan et al. 2009 the most important safety standards are ISO 10218:2011; ISO 12100:2011; ISO 13849-1:2006; ISO 13854:1996; ISO 13855:2010; ISO 14118:2000; ISO 14119:2013; ISO 14120:2002; ISO 14121-1:2007; ISO/TR 14121-2:2012; DIN EN 1175-1:06-2011; ANSI/RIA R15.06-2012.

Industrial pHRI solutions, based on these standards are available commercially (e.g. Kuka SafeRobot, ABB SafeMove), but the interaction possibilities

are very limited. The paths are not replanned online, only their velocity component is changed, either to stop the robot, either to slow it down, when the human is close to the robot. No actual collision avoidance is carried out, the robot does not avoid the collision, by choosing another path, instead it stops every time when the human worker is close, even if no collision would occur.

2.4 Path Planning

The path of the TCP is planned by so called path planning algorithms. These are in fact optimization methods Shiller et al. 1989. Optimization is mathematically defined as a minimization (or sometimes maximization) of a function, called the cost function. Optimization algorithms rarely have no additional conditions or criteria besides the cost function, that the solution candidates, also called feasible solutions, have to satisfy. In most cases these criteria, called constraints, limit the manifold of feasible solutions. A general example for such a constraint optimization is, as shown in Pike 2001:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c_i(x) \leq 0 \\ & c_j(x) = 0 \end{aligned} \tag{2.4.1}$$

Where x is the argument of the optimization, $f(x)$ is the cost function, $c_i(x)$ is a constraint which has the form of an inequality, and $c_j(x)$ is a constraint which has the form of an equality.

Computing a path for the TCP of a robot is a constraint optimization problem LaValle 2006. The argument of the optimization is the path itself, usually described as a vector of discreet points in space. If an initial path (or in some cases a set of initial paths) is required as input depends on the optimization method used. Other components (e.g. as velocity, acceleration, etc.) can also be added to the description of the path.

The inputs to the optimization algorithm come from the task of the robot. The cost function, $f(x)$ in the case of path planning can be the length of the

path, or other, more complex criteria can also be used (e.g energy consumption Ur-Rehman et al. 2009, or combination of a number of criteria, as presented in Ur-Rehman et al. 2010). The cost function is used to compare different paths with each other, in a quantified manner, to decide which is optimal (e.g. is the shortest, requires the least energy, etc.)

Constraints have to be integrated in the planning algorithm. These constrains limit the set of feasible paths. They characterize a path as valid or invalid, an so, if one of the constraints is violated during the execution of a candidate path, the path is considered invalid. In the case of path planning the constraints usually delimit areas in the environment of the robot which, for some reason, cannot or should not be occupied by the TCP and/or by the linkages of the robot.

When planning paths for industrial robots, there are certain constraints that are specific to this application. There is one inherent constraint, which is the key aspect of path planning. All feasible paths have to have as the first point the current pose of the TCP and as last point the target pose of the TCP. Important constraints are for path planning are:

- workspace limits
- obstacles
- safety criteria

It is arguable if the target point is a constraint or part of the path planning procedure. The result, the output of the optimization procedure (i.e. the path planning procedure) is the optimal path.

Path planning methods are usually categorized based on two criteria. Both criteria divide the methods into two-two large groups. Path planning methods are divided into:

- online or offline methods Raja et al. 2012, based on at which point in time is the path generated
- global or local methods, based on how the path is generated.

Generally global path planning methods are well suited for offline path generation, and local path planning methods are well suited for online path generation, but this is not always the case. Global path planning methods have been proposed also for online path planning.

Offline path generation refers to generating paths for the robot a priori. In this case path defined during the commissioning of the robot are executed repeatedly at the time of production, as presented in figure 2.4. Currently this is the most used industrial practice Pan et al. 2012. Unfortunately this approach prohibits pHRI. Since the path are generated before execution, the robot cannot react to changes in its environment by altering its path.

Online path planning opens the possibility of pHRI. In this case paths are generated "on the fly". The path or portions of the path generated in one real-time cycle are executed in the next real-time cycle, as presented in figure 2.5. This way the robot can react to changes in its environment.

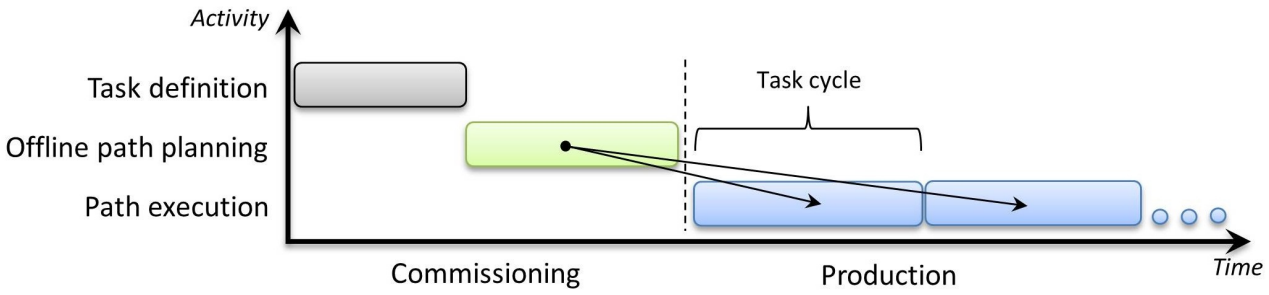


Figure 2.4: Scheduling of offline path planning

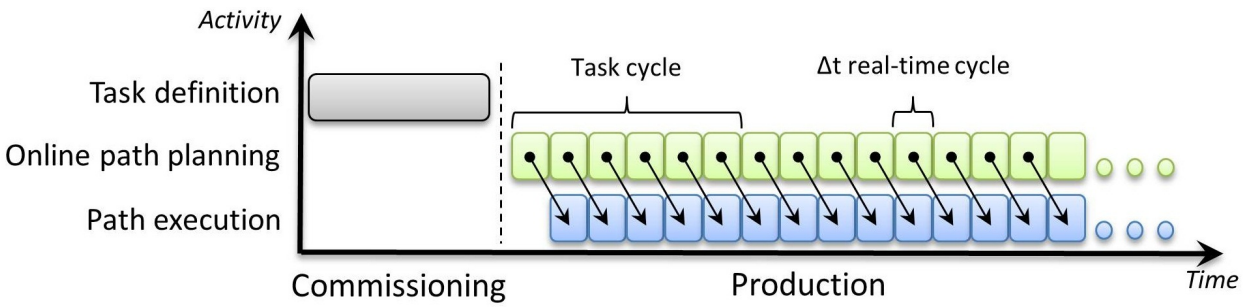


Figure 2.5: Scheduling of online path planning

Different path planning methods are recommended for different applications. When analyzing safety path planning methods, usually there are two important aspects:

- the path planning methodology, which defines how the path is generated, and
- the safety criterion, which acts as one of most important constraints to the path planning.

An extensive review of general path planning methods is presented in LaValle 2006; Latombe 1991. In the following different path planning methodologies, which have been proposed for path planning in pHRI applications, are presented and analyzed, with emphasis on real-time execution capabilities.

2.4.1 Global Methods

Global planning algorithms generate the entire path from the starting point to the target point. It is fairly straightforward to include different constraints (e.g. obstacles) and optimality criteria (e.g. shortest path), which will be minimal for the generated path, obtaining this way optimal or close-optimal paths. Their main disadvantage is that they are not suitable for real-time execution, because of their non-deterministic character (their execution time is strongly dependent on the inputs to the algorithm) and their long execution times. The entire path has to be replanned every time the environment changes (e.g. an obstacle moves). This limitation of the global planners is also described in Petti et al. 2005.

In figure 2.6 the functionality of a generic global path planner is shown. The non-deterministic nature can easily be observed, considering that the calculation time of a path is proportional to the distance between the current TCP position (TCP_i) and the target TCP position (TCP_n). The red circle representing the obstacle starts moving from position X_0 at time $t = t_0$. At time $t = t_{i+1}$ it arrives to the position $X_{t_{i+1}}$ and stops until $t = t_n$. The global path

planner plans the paths ($\wp_{t_i}$) between the current position of the TCP and the target position, in every real-time cycle (Δt).

Mostly global planning algorithms have been suggested for safety path planning in the state of the art, despite their disadvantages (e.g. non-determinism, large computational time and power needed).

An example where the calculation of the path takes significantly longer then its execution can be found in Svenstrup et al. 2010. In this article the safety criterion considered by path planning is distance based. Four zones have been defined around the human body, two of which cannot be occupied by the robot, as presented in section 2.3.1. In this article the RRT algorithm (Rapidly-Exploring Random Trees) has been proposed for planning. This algorithm is an iterative planning algorithm, it has been described in detail in Kuffner et al. 2000.

The RRT algorithm constructs a tree like structure in the search space. Given the start point, at every iteration the tree is expanded, a new node is added, if the node, and the path leading to the node is obstacle free. The new node has a

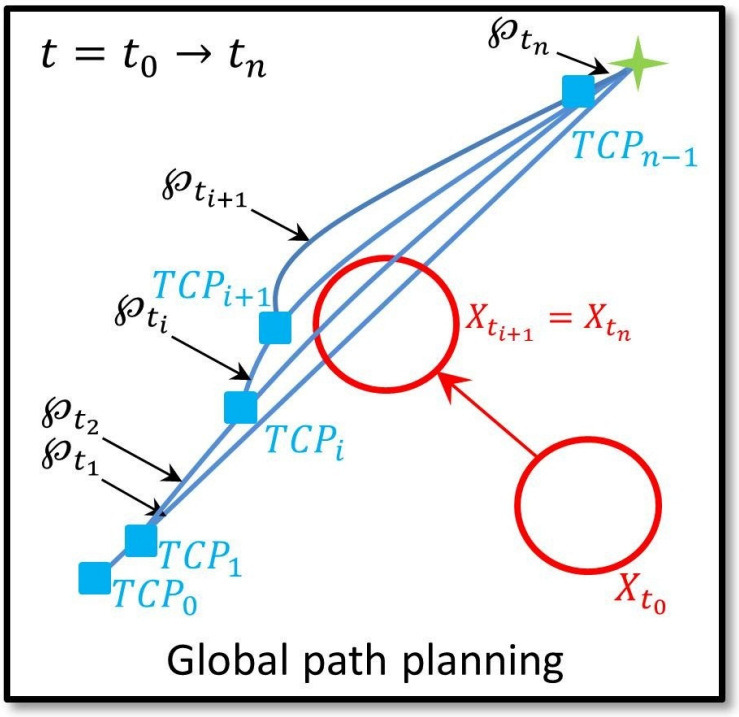


Figure 2.6: Conceptual representation of global path planning methods.

predetermined distance from an already existent node, to which it is connected to, and the expansion is done towards a random point in the workspace. When adding a new point, it is verified that kineto-dynamic constraints of the robot are respected, and that the point does not lie in restricted zones (e.g. zones occupied by obstacles). This way the search space is randomly explored. In order to compute a path faster another tree is "grown", from the target point of the path planning. This is expanded in the same manner, at the same time, but the direction of growth is not random, it always is towards a newly added point to the first tree. The algorithm is not ran until a first solution is found, instead it is ran until a specified time. This way, after a first solution is found, it is improved, until the time is up. The calculation time is presented to be $20s$. In these $20s$ the entire path between current robot position and target robot position has been generated, however only $2s$ of this trajectory is executed. This assures a $2s$ reaction time to the environmental changes, although it can be applied only in simulation since $20s$ are needed to compute the path for these $2s$. The results are presented in figure 2.7. The green path in the figure represents the least cost path, selected as the path that will be executed. The paths represented in red are "collateral" paths generated by the RRT algorithm. These paths are valid paths, generated by the RRT, but these are not optimal.

In order to match the execution time of the real-time cycle with the long execution time of the A-star algorithm, the pre-emptive version of the A-star algorithm has been suggested in Graf et al. 2009. The pre-emptive version of the A-star algorithm plans the same way as the non-pre-emptive version, the difference lies in the fact that the calculations are not carried out in one real-time cycle, but they are continued from one cycle to another. This way the pre-emptive A-star algorithm can be used together with a real-time cycle of a few milliseconds, but the actual planning will not have a result in every cycle, but after a number of cycles. A detailed description of the A-star algorithm can be found in Yao et al. 2010; Fei et al. 2004. The A-star is an iterative graph search algorithm. It selects a new part of the path or abandons parts of the path selecting a new direction at every iteration, until the goal is reached.

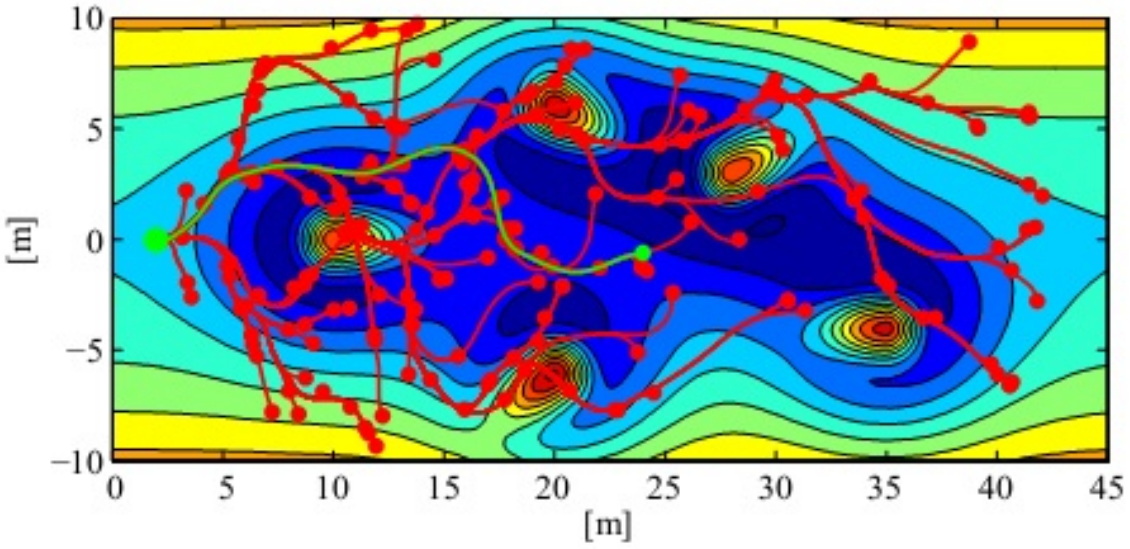


Figure 2.7: The potential map of the environment, as presented in Svenstrup et al. 2010, © IEEE 2010. Presented paths (red) are a part of the paths planned by the RRT planner (every 10^{th} vertex is presented), the green path is selected as optimal path.

The heuristic function, $f(x)$, which is used to select new parts of the path, as presented in equation (2.4.2), is usually the sum of two components.

$$f(x) = g(x) + h(x) \tag{2.4.2}$$

The first component, $g(x)$, represents the cost of the already obtained,"known" part of the path, while the second component, $h(x)$, represents an underestimate of the cost of the "unknown" part of the path, also called admissible heuristic estimate.

The total execution time of the algorithm is not reduced when using the pre-emptive version of the algorithm, it is just divided among many real-time cycles. This way the algorithm does not plan the entire path in one cycle, the planning continues from one cycle to another. The exact number of cycles required by the pre-emptive A-star algorithm for completion, due to the non-deterministic nature of the algorithm, is not known.

In Petti et al. 2005 the concept of partial motion planing (PMP) with global planing algorithms was introduced for human-robot interaction. Under hard real-time constraints the global planners cannot compute the entire path, instead, these are used to compute portions of the path. The length of the planned portion, a partial path, is limited by the time constraint. This is similar to the functionality of local planning algorithms.

In Graf et al. 2009 a method is proposed in which the initial path, planned by the A-star global planner is replanned by a pre-emptive A-star algorithm every time the path is blocked by moving obstacles. The safety criterion used for replanning is Euclidean distance. The advantage of the approach is that the real-time cycle, and so the time critical operations, are maintained. The trade-off of the pre-emptive approach is that the reaction of the robot to the change in the environment does not come in the next real-time cycle, but after a number of cycles (the exact number is hard to foresee), when the planning is finished.

The A-star algorithm has been used for safety path planning also in Liu et al. 2005. In order to speed up the planning process, the A-star algorithm does not search in configuration space or in Cartesian space, but in a pre-generated graph structure, the probabilistic road map. The road map is generated prior to execution, using random points in the workspace. These random points become the nodes in the graph. The edges of the graph are generated using a hybrid distance based criterion, which includes both configuration space and Euclidean distances. This way, a graph called the probabilistic road map (PRM) is obtained. More details and some performance improvements of the PRM have been presented in Wagner et al. 2012. THe edges in this graph represent transitions of the robot between the positions (states) corresponding to the nodes the edges connect. The path is generated using the A-star algorithm, which searches in the road map. The safety criterion is distance based. The edges in the graph are marked as valid, dangerous or invalid, based on the distance between the robot and the obstacles while executing the transition represented by the edge. For a small number of points (nodes in the road map) real time performance is achieved, but enhancing the resolution of the

search space (increasing the number of nodes in the road map graph) increases significantly the computation time.

Instead of pre-generated roadmap, in Lacevic et al. 2010b a different approach was used to limit the search space of the A-star algorithm and to reduce the execution time this way. In this case the nodes in the solution graph do not represent points in the workspace, but obstacle-free bubbles in configuration space. Bubbles have been defined as compact, diamond or hypercube shaped regions of the configuration space that are restrictions free. For better results the hypercube based definition of the bubbles have been suggested, although both approaches are presented. The manipulator can freely move around in a bubble without any collisions. The algorithm searches for a succession of neighboring bubbles that start from the start position and lead to the target position. Selection of the bubbles is done by a modified heuristic function of the A-star algorithm. The heuristic function (2.4.2) was completed with two additional components.

- The first additional component is the maximum value of the CKSDF the robot induces over all obstacles in the workspace when it is in the bubble that is considered to be added to the path. This is how the safety criterion CKSDF (described in section 2.3.1) is included in the path planning.

- The second additional term expresses if other bubbles in the vicinity of the current bubble have been verified and found unsuitable to the path.

These two additions to the heuristic function have two positive effects on path generation.

- Since free regions, where the value CKSDF is lower, and therefore are further from the robot, tend to be larger, selecting these as part of the path speeds up the execution of the algorithm (larger chunks of paths are added in one iteration, hence fewer iterations are necessary).

- The other effect is that these paths are safe, since they have been selected based on the CKSDF criterion.

The main disadvantage of this approach is its execution time. As stated in the paper, the execution time of the non-optimized implementation of the algorithm, which should terminate in a real-time cycle (i.e. in less than $1 - 20ms$) is "a couple of seconds".

A prediction based safety planning method has been presented in Najmaei et al. 2011. In this case the path is modified not just based on the current situation, but also on a predicted, future situation. The approach was named impedance based reactive control strategy. The initial path of the robot is considered to be known. The path of the human obstacle is predicted by artificial neuronal networks. The safety criterion CPDI (Cumulative Predictive Danger Index, described in section 2.3.1) is used to predict the amount of threat the robot will impose in the future and the robot is moved, by a virtual force, to a pose which has a CPDI value below a safety threshold. Since the CPDI is strongly dependent on the prediction, also the DI (Danger Index, described in section 2.3.1) is considered when the path is altered. The disadvantage of this approach is that this method uses a best case prediction of human movement, which is unfortunate, since the assurance of safety in a worst-case scenario is necessary.

The use of fuzzy logic is suggested for safety path planning in Graf et al. 2010. The paper states the method is real-time capable and uses a safety criterion, but the paper does not describe this criterion. The method itself is described with few details, but the experimental validation suggests good results.

2.4.2 Local Methods

Local path planning methods have a "one-shot" nature. These are not iterative algorithms; the result of the planning is given after executing the algorithm, which includes no iterative cycles. However, this result, is not the path from the current state to the target state, but it is the next state the robot should transition to in order to reach the goal at a later point in time. Their most important advantage is their fast execution time, real-time capability and deterministic nature. The time needed for generating the next state is not dependent on the

current and the target state. Their major drawback is that local minima points can impede the algorithm to find the path to the target state, and that a global optimality criteria (e.g. minimization of energy consumption, etc.) is usually not reached.

In figure 2.8 the functionality of a local path planning algorithm is presented. It can be observed, that in every cycle only a small portion of the path ($\Delta\wp_{t_i}$) is planned. The deterministic nature can easily be observed considering that the calculation time of a path is proportional to the distance of the planned portion ($\Delta\wp_{t_i}$). The red circle representing the obstacle starts moving from position X_0 at time $t = t_0$. At time $t = t_{i+1}$ it arrives to the position $X_{t_{i+1}}$ and stops until $t = t_n$. The the local path planner plans small portions of the path ($\Delta\wp_{t_i}$) in every real-time cycle (Δt).

Local methods, despite their characteristics that favor real-time implementations are seldom suggested for safety path planning.

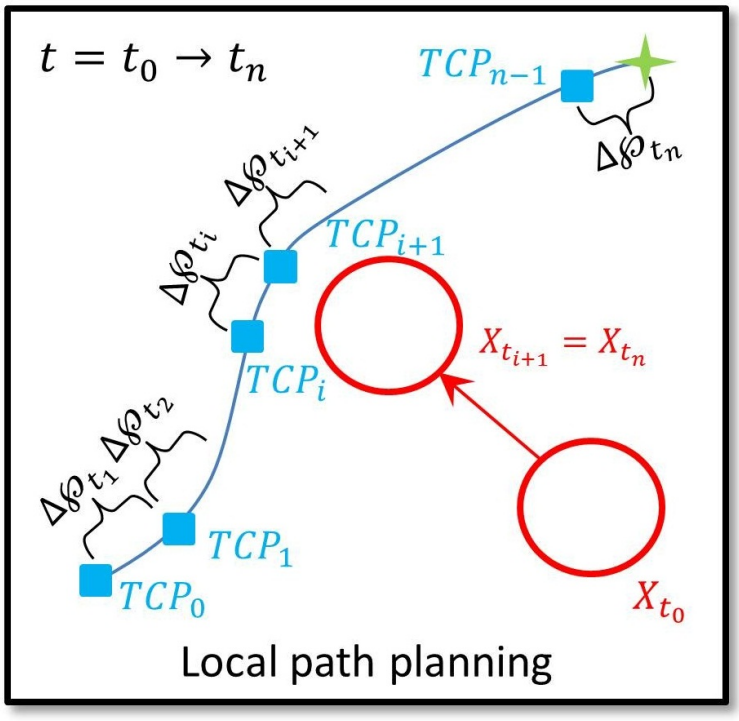


Figure 2.8: Conceptual representation of local path planning methods.

The Artificial Potential Field (APF) method is a local path planning method where different elements of the workspace are modeled as potential fields, generated by potential functions. The potential functions attributed to different elements of the workspace can either attract (e.g. target pose) or repel (e.g. obstacles) the robot. The superposition of these potential functions and considering their virtual effect on the robot generates a virtual force that drives the robot away from the obstacles (repellent component) and towards the target pose (attractive component). Many publication demonstrate the advantages of the APF method for different applications, such as path planning for mobile robots, in Sfeir et al. 2011; Lee et al. 2006 and for unmanned aerial vehicles in Helble et al. 2007, car-following application in Tao et al. 2011, autonomous camera control in Burelli et al. 2009, planning paths in computer games in Yannakakis et al. 2004; Hagelback et al. 2008, particle swarm optimization in Xu et al. 2010, decentralized path planning of swarm systems in Kim et al. 2004 and also haptic feedback in Ren et al. 2007.

The potential function plays a key role in the method. Many publications deal with improving different aspects of the potential functions. Variants of the potential functions include:

- the FIRAS function, presented in Khatib 1986, which improves the representation of the shape of the obstacle
- the Generalized potential function, presented in Krogh 1984, which takes into account also the velocity of the obstacles but keeps spherical symmetry.
- the Superquadratic potential function, presented in Volpe et al. 1990, which improves the representation of the shape of the obstacle, but keeps
- the Harmonic potential function presented in Kim et al. 1992, which eliminates the local minima problem (at the cost of real-time execution)
- the “new” potential function, presented in Ge et al. 2002, solves the so called GNRON problem (Goal Not Reachable when Obstacles Nearby).

The APF method is mostly used for path planning for mobile robots. Just a few papers deal with planing paths with the APF method for manipulators and even fewer with integrating kinematic constraint into the APF planning algorithm for manipulators. Workspace limits of mobile robots cannot be compared to workspace limits of the manipulators. From the three workspace limits the joint limits have been integrated to the algorithm in Khatib 1986 using polynomial potential function. A detailed description of the functionality of the method is presented in Khatib 1986; Csiszar et al. 2012 and in chapter 3.

Only one paper was found which deals with integrating safety criteria to the APF method. In Chen et al. 2009 the APF method, enhanced by GA (Genetic Algorithm) has been suggested for safety planning. This approach, named E-APF (Evolutionary Artificial Potential Field) was also presented for general applications (not for safety planning) in Vadakkepat et al. 2000. In the E-APF method an additional global optimization method completes the local APF method. A scenario of application is shown in figure 2.9, as presented in Chen et al. 2009. The geometrical characteristics in the virtual potential model are given by the real world system, but the charge intensities, are in most cases attributed empirically. These empirical parameters have a major effect on the generated path. A GA optimization algorithm has been suggested in Chen et al. 2009 is be used to eliminate the empirical assignment of these parameters. The cost function for GA optimization (called fitness function in GA terminology) is the inverse of the safety criterion considered in this paper. The minimum distance to all obstacles along the path has to be maximized. Figure 2.9 shows this visually, as presented in Chen et al. 2009.

A description of the GA can be found in Gao et al. 2008; Hui et al. 2009. At first, an initial population is selected. The members of this population (the chromosomes) are sets of possible solutions. In this case these are sets of values of the empirically assignable parameters. In order to evaluate the fitness function (the cost function) the entire path for all the members of the population has to be generated. After evaluating the fitness function for the whole population the GA specific operators (crossover, mutation, etc.) are applied and the second generation population is obtained. The fitness function

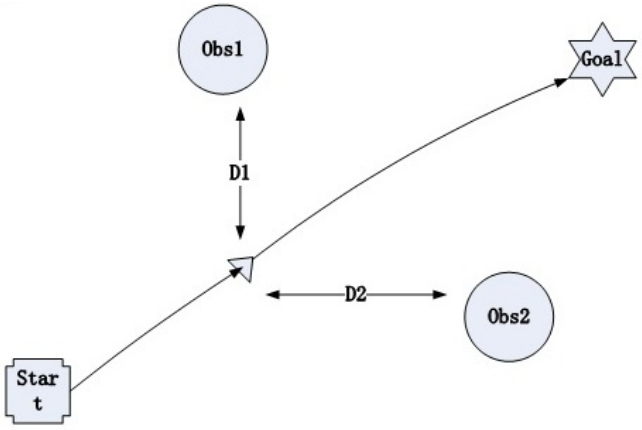


Figure 2.9: Use-case of the E-APF algorithm, as presented in Chen et al. 2009, © IEEE 2009. The scaling constants of the potential functions attached to the obstacles Obst1 and Obst2 are optimized. The cost function is the inverse of the sum of distances $D1$ and $D2$.

evaluation is restarted for the second generation population, similarly as for the case of the initial population. The third generation population is obtained from the second generation population, by applying the GA specific operators, and the algorithm continues to make new generations until stopped. In the paper it is mentioned that the path which is considered optimal (or close-optimal, having in mind the stochastic nature of the GA) is obtained from the 100th generation (a value chosen empirically). The advantage of such an approach is that it assures global optimality, even when using local planning method. The disadvantage is, that the parameters calculated by the global optimization method are dependent on the obstacle positions, and so the global optimization has to be carried out at every obstacle movement. There is no indication of the execution time in the paper, but it is assumed that this approach is not real-time capable and it is not deterministic.

A framework for path planning using elastic strips is described in Corke et al. 2000; Brock et al. 2000. Its application to human environments is presented in Brock et al. 2002. The elastic strip framework incorporates both local and global behavior. The global behavior is related to the goal point, thus it is related to the task of the robot. The local behavior is related to obstacle avoid-

ance, thus it represents the reaction to environmental changes. The framework applies incremental modifications to a previously planned path.

The path in the case of the elastic strips framework is not seen as a curve described by the TCP, but as a volume in the workspace that is occupied by the robot during the execution of the path. A slight modification in the path will cause a slight modification in the occupied workspace volume. The elastic tunnel has been defined as a workspace volume that is calculated by considering the free (e.g. without obstacles) space around a workspace volume that is occupied by the robot during the execution of a path. The elastic tunnel, together with its "generator" path is called an elastic strip. Basically it represents a volume of the workspace which is free of obstacles and may be occupied by the robot, this way, giving indications about how much the initial path can be altered and still be collision free. Given the elastic strip, a planner similar to the APF (Artificial Potential Field) planner was used to generate a new path wich is contained inside the elastic tunnel. In this case the APF like planner is not planning the next step of the robot, but it is applied to a discretized initial path, deforming it. Once a path is deformed this way, after the obstacle has moved, the path would not be "deformed back" to its initial state. To facilitate deformation of the path, in the absence of obstacles, virtual springs have been defined that deform the newly obtained path to the original state when obstacles are not anymore present. This way the path behaves like a band of rubber that is deformed by the proximity of the obstacles. No real-time cycle times are presented in the paper, but the presented experimental demonstrations suggest real-time capability at least for the path deformation phase. The collision avoidance in the elastic framework is distance based, and no other safety criterion is suggested. The advantage of this approach is the real-time capability. The disadvantage is the initial planning of the path, for which real-time generation does not seem plausible.

The most important characteristic of a path planner algorithm for safe pHRI is real-time capability. Although in some cases real-time execution of global path planning algorithms can be achieved, these are iterative solutions, hence they are not deterministic, and real-time performance in every scenario cannot

be guaranteed. It can be observed, that local planning methods, although they are real-time capable, have not been preferred for pHRI. When used these have been completed by global optimization methods. The reason behind this can be that the integration of additional constrains in the local planning algorithms is a difficult task.

2.5 Scientific Open Problems

Based on the presented analysis of the state of the art, scientific open problems can be identified. These open problem are:

- Most path planning methods proposed for pHRI face difficulties when considering real-time execution.
- Different constraints (e.g. workspace limits, safety criterion, etc.) have not been fully implemented in local planning algorithms.
- The safety criteria proposed face challenges when considering worst-case scenarios.

These open problems have been analyzed and solutions to these problems are proposed thought this work. The proposed solutions offer new insights to these problems. The idea behind the solutions has not been to create an industrially deployable solution, but to research novel ways how the identified problems can be solved, and offer a starting point and new insights for a possible industrial deployment.

3 Theoretical Background

3.1 The APF Path Planner

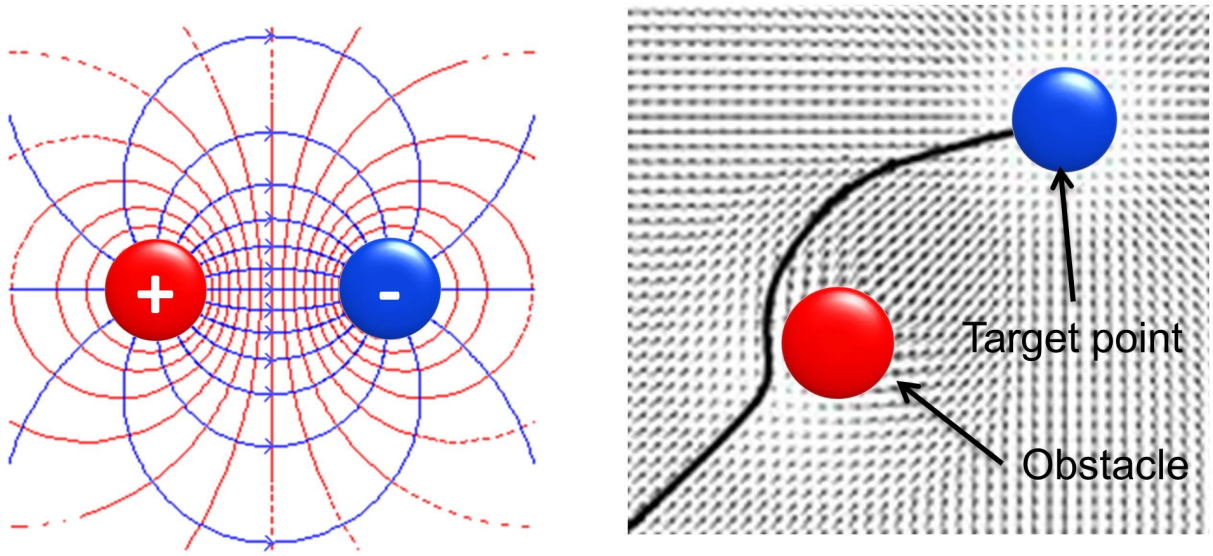
The APF method can be easily explained through an electrostatic analogy, as presented in Hwang et al. 1992; Valavanis et al. 2000. Every electrostatic charge creates an electrostatic potential field U around itself. Two charges of the same sign repel each other while two charges having different signs attract each other. This is represented in figure 3.1a The magnitude of the electrostatic force $\vec{F}_{\text{ES}}$ acting upon the charges, by the means of which they are attracted or repelled, is proportional to the potential field U, according to Alonso et al. 1992. The direction and the orientation of the electrostatic force ($\vec{F}_{\text{ES}}$) is the same as the direction and orientation of relative position vector.

$$\left\|\vec{F}_{\text{ES}}\right\| = U \cdot \eta_q \tag{3.1.1}$$

$$\vec{F}_{\text{ES}} = \left\|\vec{F}_{ES}\right\| \cdot \hat{r} \tag{3.1.2}$$

Where η_q is the electrostatic charge the potential field is repelling or attracting and $\hat{r}$ represents the unit vector of the relative position vector between the point where the field is evaluated and the charge causing the potential field. The units of measurements are: N for the force and N/C for the potential, C for the charge. The unit vector has no unit of measurement. As in many papers in the scientific literature about potential field methods, the unit of measurement of the potential field is neglected also in this thesis.

This nature inspired phenomena can be applied to path planning for robots, as shown in figure 3.1b. Obviously robots do not operate in strong electrostatic fields, that could influence their motion, but a virtual world can be created



(a) Electrostatic charges and potential field

(b) Path planning based on potential fields

Figure 3.1: Electrostatic system and APF method analogy

where the elements of the real world are modeled similarly to electrostatic charges.

In published papers using the APF method, the virtual force, $\vec{F}_{\text{Virtual}}$, induced by a artificial (virtual) potential field is defined differently as in the electrostatic analogy case, equations (3.1.1) and (3.1.2). It is defined as the negative gradient of the potential field, as presented in (3.1.3). This approach makes the electrostatic analogy only conceptually true.

$$\vec{F}_{\text{Virtual}} = -\nabla U \tag{3.1.3}$$

As in the case of electrostatic potential fields, virtual potential fields can also be categorized as attractive or repellent fields, otherwise known as as potential sinks or potential sources.

3.1.1 Target

Potential sinks attract the TCP of the robot. The value of the potential in the point where the potential sink is defined is the lowest; this is the global minimum point. The target for the TCP is modeled as a potential sink, it

attracts the TCP. The potential function U_{Target} attributed to the potential sink can have the same expression as the potential field created by a negative electrostatic charge, but not necessarily. Different types of potential functions can be attributed to the potential sink in order to achieve a desired behavior. Since there is only one target for the TCP, usually, only one potential sink is defined.

$$U_{\text{Target}}\left(X_{\text{TCP}}\right) = \frac{1}{2} k_{\text{Target}} \cdot D\left(X_{\text{Target}}, X_{\text{TCP}}\right)^2 \tag{3.1.4}$$

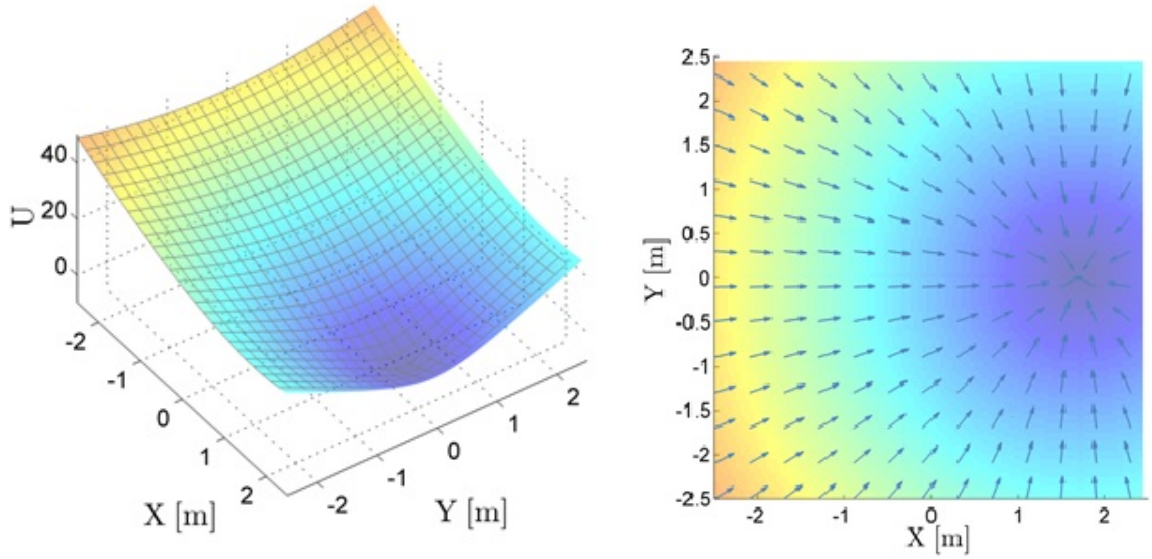
$$-\nabla U_{\text{Target}}\left(X_{\text{TCP}}\right) = -k_{\text{Target}} \cdot D\left(X_{\text{Target}}, X_{\text{TCP}}\right) \tag{3.1.5}$$

Where $D(X_{\text{Target}}, X_{\text{TCP}})$ represents the Euclidean distance between the point where the target is defined, X_{Target}, and the point where the potential is evaluated, which for path planning coincides with the coordinates of the TCP, X_{TCP}. k_{Target} is a scaling constant.

Figure 3.2 presents the potential field created by a potential sink. In figure 3.2a the value of the potential is presented for a target point defined in two dimensions, while in figure 3.2b the gradient of the potential function is visualized. In figure 3.2b the arrows represent the unit vector of the negative gradient, $-\nabla \hat{U}_{\text{Target}}$.

3.1.2 Obstacles

Potential sources repel the robot. Obstacles in the workspace are modeled as potential sources, so these repel the robot. The value of the potential U_{Obst} in this point is high, and the gradient of the potential field $-\nabla U_{\text{Obst}}$ points away from the potential source. In a workspace there can be more than one obstacle. Each obstacle has its own potential function. Similarly as in the case of the potential source, it is not necessary to attribute a potential function that resembles the potential function of a positive electrostatic charge. Different potential functions will cause different collision avoidance behaviors. Figure 3.3 presents the potential field created by a potential source. In figure 3.3a the value



(a) (b)

Figure 3.2: Visual representation of the attractive potential field, generated by the target potential function

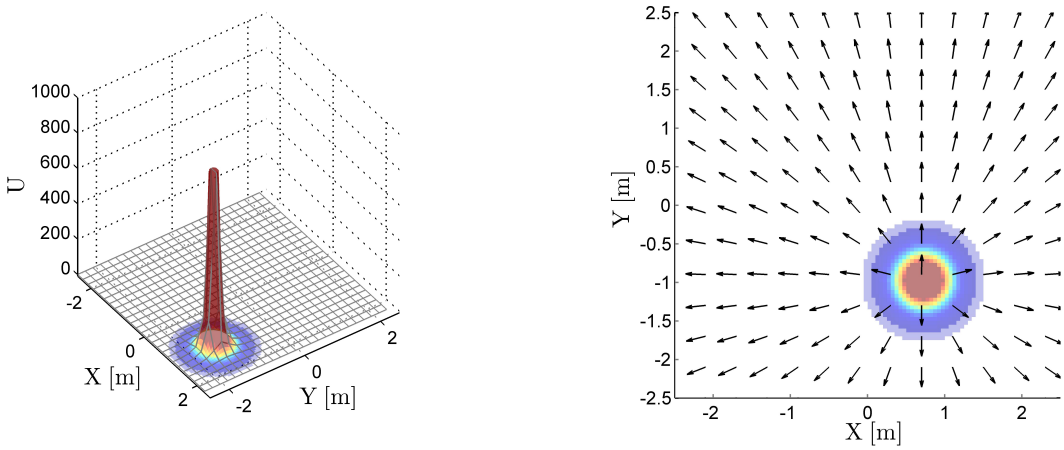
of the potential is presented for an obstacle defined in two dimensions, while in figure 3.3b the gradient of the potential function is visualized. In figure 3.3b the arrows represent the norm of the gradient $-\nabla \hat{U}_{Obst}$.

$$U_{\text{Obst}}\left(X_{\text{TCP}}\right) = \begin{cases} \frac{1}{2} k_{\text{Obst}} \left(\frac{1}{D(X_{\text{TCP}}, X_{\text{Obst}})} - \frac{1}{D_0} \right)^2, & D\left(X_{\text{TCP}}, X_{\text{Obst}}\right) \leq D_0 \\ 0, & D\left(X_{\text{TCP}}, X_{\text{Obst}}\right) > D_0 \end{cases} \tag{3.1.6}$$

Where $D(X_{\text{TCP}}, X_{\text{Obst}})$ represents the Euclidean distance between the obstacle and the TCP, D_0 delimits the effect of the obstacle on path planning to a given distance, and k_{Obst} represents a scaling constant.

Obstacles should also repel the linkages contained in the robot structure, not only the TCP, in order to avoid collisions with the robot itself. The method to achieve this behavior is based on evaluating the potential field not only at the TCP, but also in different characteristic points of the robot as described in Khatib 1986.

Superposition of the different potential functions is done, as in the case of electrostatic field, by summation. The value of the total potential, U_{Total}, in



(a) (b)

Figure 3.3: Visual representation of the repellent potential field, generated by the obstacle potential function

one point is the sum of all potentials U_i in that point. Also, the value of the gradient of the total potential field, ∇U_{Total}, in one point, is the sum of the gradients of all potential fields ∇U_i.

$$U_{total} = \sum_{i=0}^{n} U_i = U_1 + U_2 + \cdots + U_n \tag{3.1.7}$$

$$\nabla U_{total} = \sum_{i=0}^{n} \nabla U_i = \nabla U_1 + \nabla U_2 + \cdots + \nabla U_n \tag{3.1.8}$$

Where n is the total number of potential fields that are superositioned.

Applying equation (3.1.3) to (3.1.7) and (3.1.8) leads to

$$\vec{F}_{total} = -\nabla U_{total} = -\nabla U_{target} - \nabla U_{obstacle} \tag{3.1.9}$$

Equation (3.1.9) describes the virtual force resulting from the superposition of all the potential fields. This is equivalent to the superposition of all the effects of the potential fields (target attracts, obstacles repel).

3.1.3 Conversion of the Virtual Force to Reference Value

All the information about the surrounding environment is condensed into the value and the gradient of the total potential field, which equals the total virtual force, $\vec{F}_{\text{Total}}$ based on equation (3.1.9). This is the reason why obstacles (a constraint for the path planning) have to be formulated as a potential function. Their effect of limiting the feasible solutions has to be reflected in this vector.

The planning of the next step is based the total virtual force, $\vec{F}_{\text{Total}}$. The virtual world is modeled at every time step after the current state of the real world. This is presented also in figure 3.4. All coordinates describing the current state of the robot, of the obstacles, of the target point, etc. are input to the virtual world. Output from the virtual world is the next state of the robot, which will be the reference for the robot controller.

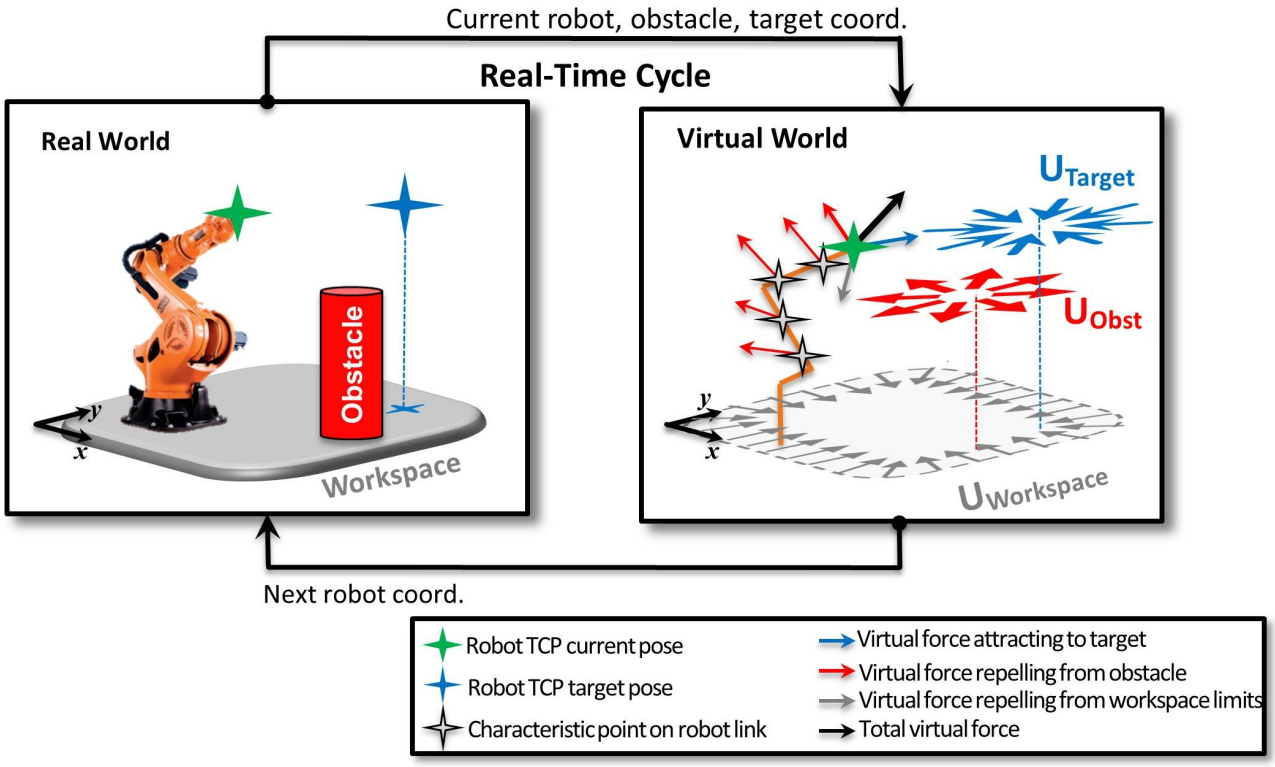


Figure 3.4: Overview of the path planner functionality

The virtual force, $\vec{F}_{\text{Virtual}}$, defined in the virtual Cartesian space is then mapped to the joint space, using the inverse Jacobi matrix.

$$\tau = J^{-1} \times F_{\text{Virtual}} \tag{3.1.10}$$

Where τ represents the vector of joint torques, J^{-1} represents the inverse Jacobi matrix.

The obtained joint torques are then applied as reference to the torque controlled joint motors, as presented in Khatib 1986. For mobile robots this approach is well suited. However, in the case of robot manipulators this requires the precise correlation between the dynamic model of the robot and the parameters of the potential field, as well as a good compensation of the gravitational and friction effects (which also require a precise dynamical model).

Applying the virtual force to the robot, produces the desired movement only if if the dynamical model and the potential fields are sufficiently well correlated. Unfortunately precise dynamical models for commercially available manipulators are not easy to obtain. This can be one explanation why most papers concerning the APF method only present validations in simulation, and not also experimentally.

Another possible solution for obtaining velocity from potential fields is presented in Reimann et al. 2010. In this case the virtual force is mapped as acceleration reference.

3.1.4 Workspace Limits

Workspace limits delimit the workspace of the robot. In Brisan et al. 2011 these are categorized as:

- Geometrical workspace constraints, representing the limit which originates from the sum of the length of all linkages of the robot. This type of workspace limit is strongly related to the inverse kinematic functions. It delimits the space surrounding the robot to zones which can be reached by the TCP and zones which cannot be reached by the TCP, based on

the inverse kinematic function, f^{-1}. If the inverse kinematic function has solution for a given point in space, then it is a point which satisfies the geometric workspace constraint and the point can be included in a feasible path. This way this constraint has the form of an equation. This constraint reflects only the theoretical structure of the robot.

$$Q = f^{-1}(X) \tag{3.1.11}$$

Where Q represents the vector of all joint coordinates, having the dimensions $1 \times n$. f^{-1} represents the inverse kinematic function, and X represents the coordinate of the TCP, having the dimensions $1 \times n$.

- Mechanical workspace constraints, representing the mechanical limits of the joints, and possible collisions of linkages. This type of workspace limit is related to the joints used in the structure of the robot. It further restricts the zone delimited by the geometrical constraint, based on the joint limits. If the coordinate of every joint is between the given minimum and maximum threshold values, then coordinates of the TCP satisfy this constraint. This way, the point represented by the coordinates of the TCP can be included in a feasible path. This way this constraint has the form of an inequality. The constraint reflects the limits of the practical solution applied when designing and/or constructing the robot.

$$Q_{min} < Q < Q_{max} \tag{3.1.12}$$

Where Q_{min} represents the vector of all inferior joint limits, Q represents the vector of actual joint coordinates, and Q_{max} represents the vector of all superior joint limits.

In many cases the collision of linkages is not possible due to the mechanical joint limits. If linkage collisions are possible, they can be treated similarly to mechanical joint limits, since these also have the form of inequalities.

The distance between linkages should always be greater then zero, or a minimal threshold value.

$$d\left(e_i, e_j\right) > 0 \tag{3.1.13}$$

Where $d\left(e_i, e_j\right)$ represents the euclidean distance between two linkages.

- Singularities, representing poses of the TCP where the robot can become uncontrollable. This type of workspace limit is also related to Jacobian matrices of the robot. Singular points in the workspace cannot be included in feasible paths. These can be identified based on the rank deficiency of the Jacobian matrices, according to Yang et al. 2005a; Yang et al. 2005b. If the Jacobian matrices at given TCP coordinates have no rank deficiencies then the constraint is satisfied and the point can be included in a feasible path. Rank deficiency can be simply expressed as

$$det\left(J_A\right) = 0 \tag{3.1.14}$$

or

$$det\left(J_B\right) = 0 \tag{3.1.15}$$

considering the derivate of the implicit form of the kinematics function:

$$J_A \cdot \dot{X} + J_B \cdot \dot{Q} = 0 \tag{3.1.16}$$

Where:

$$J_A = \frac{\partial F}{\partial X}$$

$$J_B = \frac{\partial F}{\partial Q}$$

F represents the implicit form of the kinematic function , $F(X, Q) = 0$, X represents the TCP coordinates and Q represents the joint coordinates.

Note that if the elements of the Jacobian matrices are not calculated exactly, but with a certain precision (i.e. obtained with numerical methods) other methods than the one using the determinant should be used, to analyze rank deficiency (e.g. Singular Value Decomposition, SVD, as presented in Csiszar et al. 2012; Yanai et al. 2011).

Throughout this work theoretical aspects are presented in a general form. In the case of the APF method this general form is applied to a 2 degrees of freedom (DoF) planar, serial manipulator for the purpose of visual representation and numerical simulation. For experimental validation a 6 DoF serial structure was used. Applying the generalized theory to a simple structure makes the graphical representations more intuitive. The 2 DoF type of robot was chosen, because it is simple, intuitive and the movements of this type of manipulator are in two dimensions, this way the third dimension can be used to visually represent other data (e.g. the potential fields analyzed during research). Geometrical characteristics of this robot, used as example, can be found in table 3.1.

First linkage length	$l_1 = 1m$
Second linkage length	$l_2 = 1m$
q_1 joint minimum value	$q_{1_{min}} = {}^{-\pi}/_{2}$
q_1 joint maximum value	$q_{1_{max}} = {}^{\pi}/_{2}$
q_2 joint minimum value	$q_{2_{min}} = {}^{-\pi}/_{2}$
q_2 joint maximum value	$q_{2_{max}} = {}^{\pi}/_{2}$

Table 3.1: Geometrical characteristics of the two degrees of freedom serial robot used as example throughout this paper

The inverse kinematic function of a robotic manipulator expresses the joint coordinates in function of the TCP coordinates. Its general form is:

$$\{Q_{\mathrm{TCP}_1} \dots \ Q_{\mathrm{TCP}_n}\} = f^{-1}\left(X_{\mathrm{TCP}}\right) \tag{3.1.17}$$

Where n is the number of possible solution of the inverse kinematic function f^{-1}, Q_{TCP_i} represent one solution to the inverse kinematic function and X_{TCP} represents the coordinates of the TCP.

When considering a 6 DoF industrial manipulator type robot, for the majority of points in the workspace, the inverse kinematic problem has 4 solutions, acooring to Angeles 2002. The inverse kinematic problem of the 2 DoF serial manipulator has two solutions, for the majority of points in the workspace. Let these two solutions be called lower solution and upper solution. This way equation (3.1.17), when applied to the 2 DoF serial manipulator, takes the form:

$$\{Q_{\mathrm{TCP}_1}\ Q_{\mathrm{TCP}_2}\} = f^{-1}\left(X_{\mathrm{TCP}}\right) \tag{3.1.18}$$

Where Q_{TCP_1} represents the lower solution to the inverse kinematic function f^{-1}, Q_{TCP_2} represents the upper solution to the inverse kinematic function f^{-1}, and X_{TCP} represents the coordinates of the TCP.

Figure 3.5 presents the workspace of this 2 DoF serial robot. Figure 3.5a shows the workspace of the 2 DoF robot in configuration space. The inverse kinematics of this type of robot has for almost all points inside the workspace two solutions. The point marked with the star in the Cartesian space (figure 3.5b, 3.5c, 3.5d) will have two equivalent points in the joint space figure 3.5a, one corresponding to the upper solution (figure 3.5c) and one to the lower solution (figure 3.5d). By comparing figures figure 3.5c and figure 3.5d it can also be observed that the certain areas of the workspace can only be reached either by the upper solution either by the lower solution. This is explained by the joint limits present. In some papers these are neglected, but, as the figure shows, they have an important influence.

This causes also the different workspace limits in Cartesian space for upper solution and lower solution of the robot. The workspaces, both in Cartesian space and joint space are used throughout this work to visually represent specific aspects of the research.

Applying the potentials presented in equations (3.1.5)-(3.1.6) and in figures 3.2 and 3.3, brings the following conclusion: Planed paths do not respect the workspace limits. This way the path planner algorithm plans paths that are outside of the workspace of the robot, and cannot be executed, thus the target point will not be reached by the TCP.

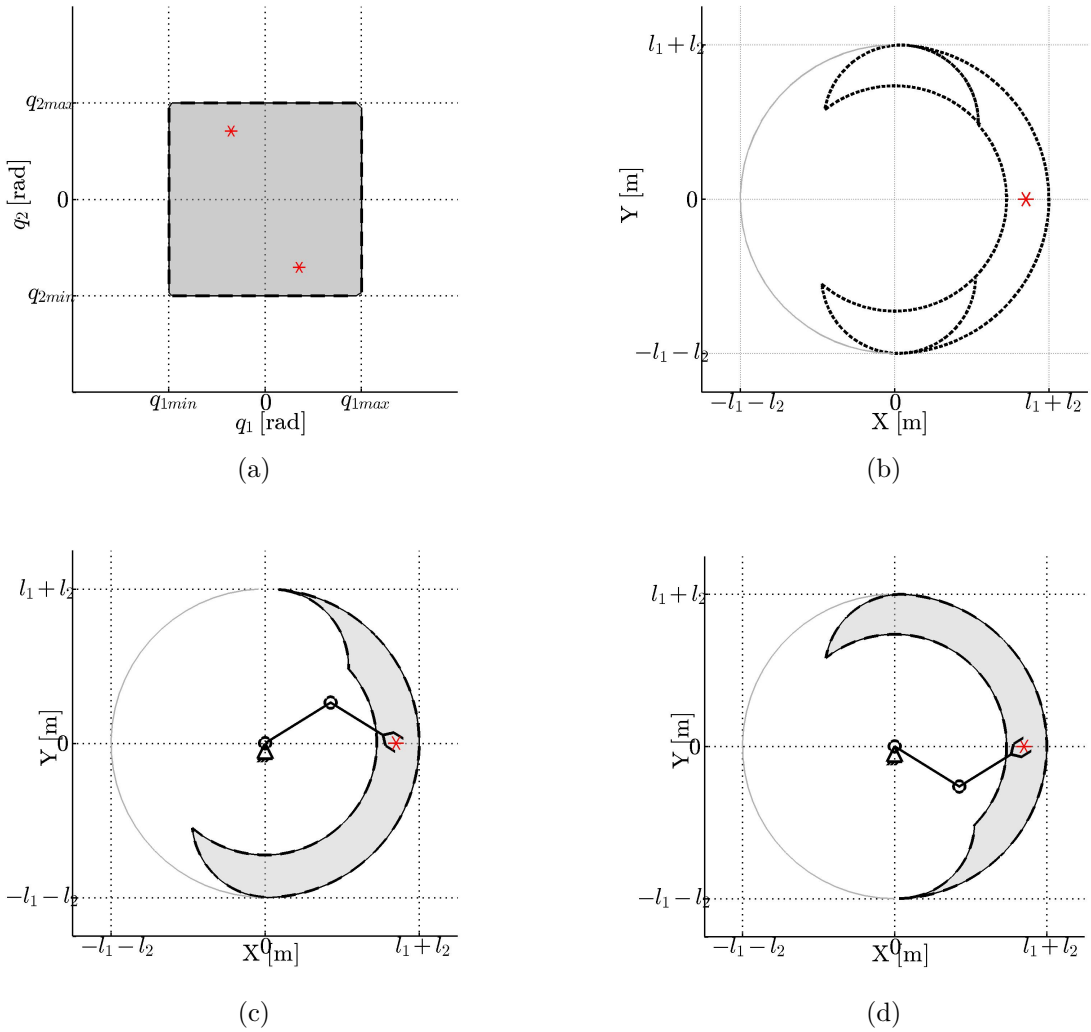
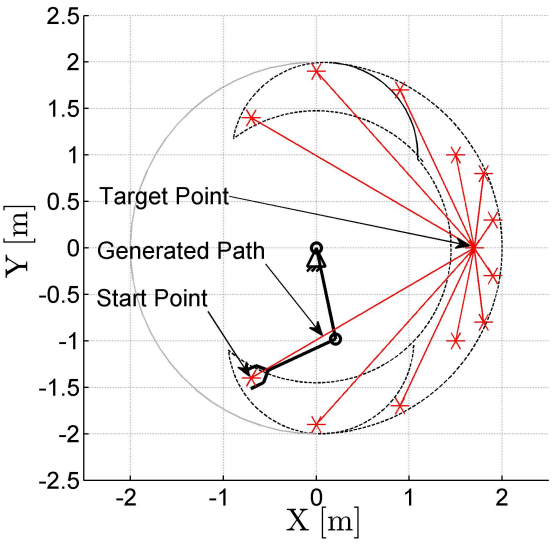


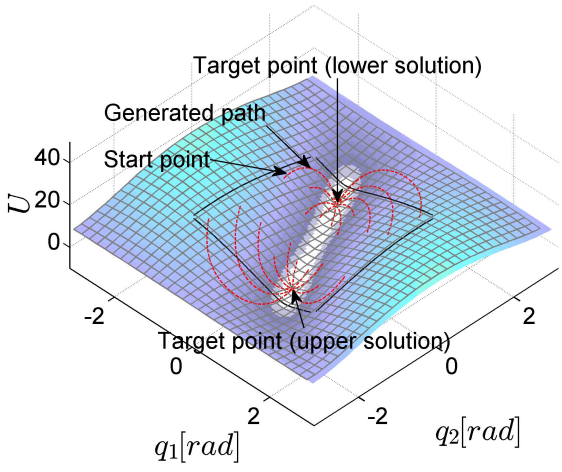
Figure 3.5: Workspace of a two degrees of freedom planar manipulator (a) in joint space, (b) in Cartesian space, for both solutions, (c) in Cartesian space, for the upper solution, (d) in Cartesian space, for the lower solution

The potential field planner having a starting point X_{start} will plan a path, following the gradient of the potential functions. When the potential function is defined in the Cartesian space as in equation (3.1.5), the path planning without any obstacles will take place as presented in figure 3.6. All red lines in figure 3.6 represent one planned path from a starting point to a common target point. In figure 3.6a the robot workspace and the paths are presented in Cartesian space. In figure 3.6b the potential field of the target point is presented in joint space, and in figure 3.6c the potential field of the target point are presented in

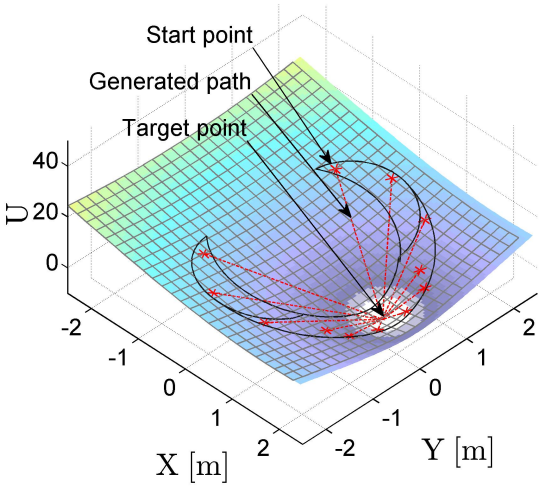
Cartesian space. In the cases presented in 3.6b and 3.6c the third dimension is the potential field U_{Target}.



(a)



(b)



(c)

Figure 3.6: Path planning in Cartesian space represented (a) in Cartesian space, (b) joint space, with target potential function, (c) in Cartesian space, with target potential function

It can be observed that the planned path are straight lines in the Cartesian space (since there are no obstacles to avoid) and that these lines do not regard the workspace limits. These paths cannot be executed by the robot.

On solution for generating paths inside the workspace is to plan the path in joint space instead of Cartesian space. It is possible to plan in joint space with the APF planner, by modifying how the target (attractive) potential is defined.

Since usually in robotics a target is defined in Cartesian space, the inverse kinematic function has to be used to define the Cartesian space target in joint space. In the general case this can be done based on equation (3.1.17). Applied to 2 DoF robot it becomes:

$$\{Q_{T_1}\ Q_{T_2}\} = f^{-1}\left(X_{Target}\right) \tag{3.1.19}$$

Where the size of Q and X is $n \times 1$, n represents the DoF of the robot, in the case of the 2 DoF serial manipulator $n = 2$. The vector $Q_{T_1} = \left[q_{1T_{low}}\ q_{2T_{low}}\right]^T$ represents the lower solution and the vector $Q_{T_2} = \left[q_{1T_{up}}\ q_{2T_{up}}\right]^T$ represents the upper solution of the inverse kinematics function f^{-1} for the Cartesian coordinates of the TCP target point $X_{\text{Target}} = \left[x_{\text{Target}}\ y_{\text{Target}}\right]^T$.

The potential function attributed to the target point is defined in joint space. A potential function defined in joint space has a similar form as the potential function in Cartesian space. Defining the target potential function in joint space, assures, that the generated path, in the absence of obstacles, will not violate the workspace limits of the robot. The potential field created by the target point, defined in joint space is presented in equation (3.1.20).

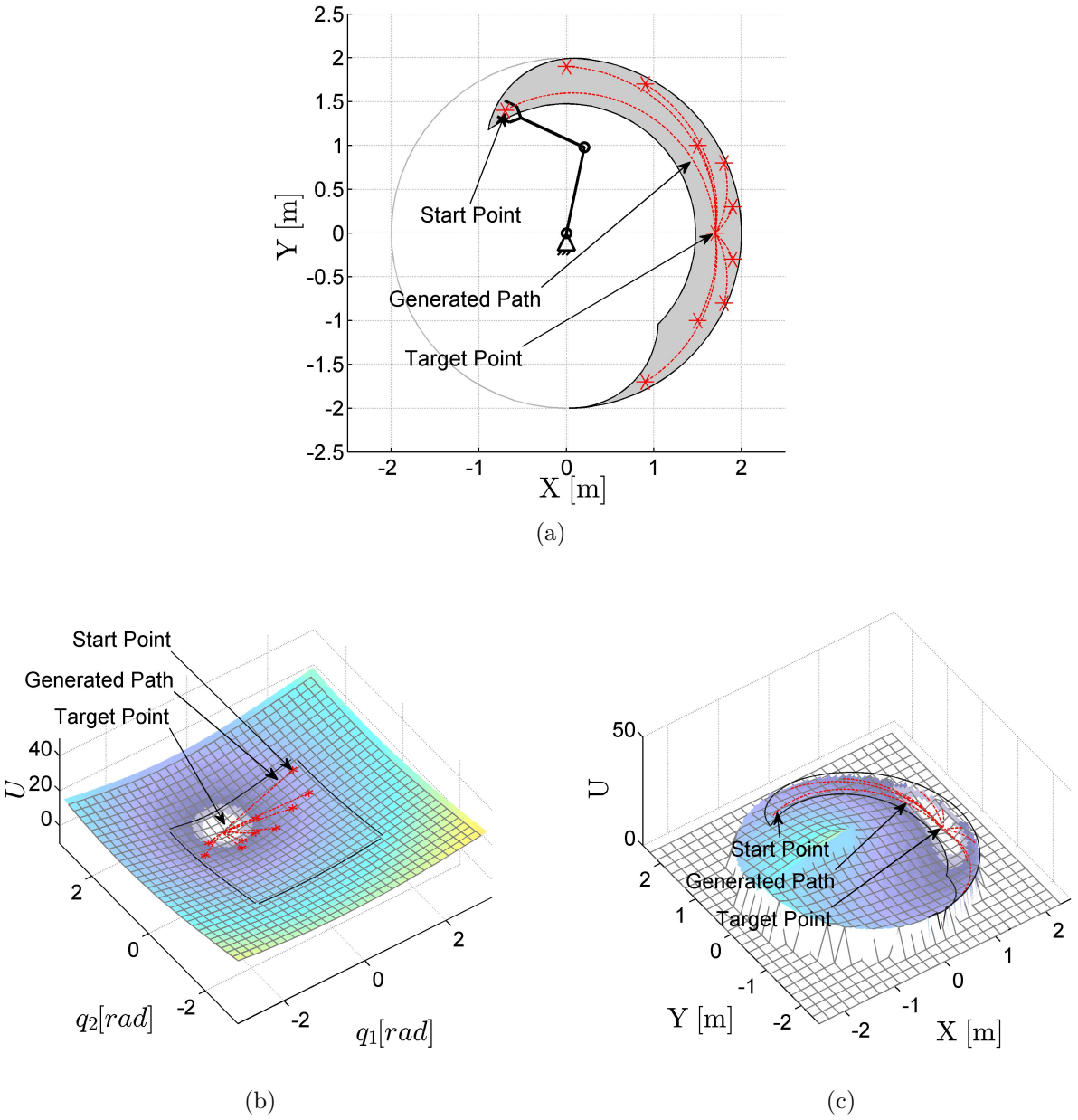
$$U_{\text{Target}}\left(Q_{T_i}\right) = \frac{1}{2} k_{\text{Target}} \cdot D\left(Q_{T_i}, Q_{\text{TCP}}\right)^2 \tag{3.1.20}$$

$$-\nabla U_{\text{Target}}\left(Q\right) = -k_{\text{Target}} \cdot D\left(Q_{T_i}, Q_{\text{TCP}}\right) \tag{3.1.21}$$

Where Q_{T_i} is the target point in joint space. It has the i index, because targets are usually defined in Cartesian space, and have more then one equivalent in joint space. Q_{T_i} represents either the lower either the upper solution of the 2 DoF robot based on equation (3.1.19). For other robot types more then two solutions are also possible. One has to be chosen a priori. k_{Target} is a scaling

constant and $D\left(Q_{T_i}, Q_{\text{TCP}}\right)$ represents the distance between the current TCP coordinates and the target TCP coordinates in joint space.

Figure 3.7 presents the planned paths in the joint space using the upper solution Q_{T_1}.



(a)

(b)

(c)

Figure 3.7: Path planning in joint space using upper solution, represented (a) in Cartesian space, (b) in joint space, with target potential function, (c) in Cartesian space, with target potential function

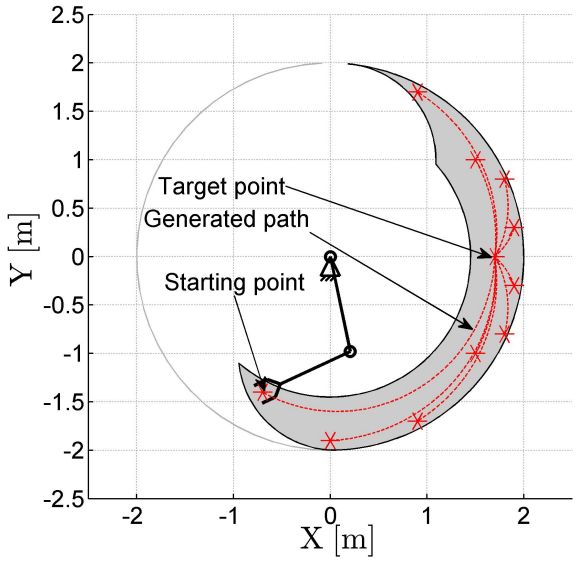
Paths generated by the planner from different starting points are marked by continuous lines. These paths, planed in joint space have the advantage of respecting the workspace limits. Figure 3.7a shows these paths in Cartesian space. Figure 3.6b presents the potential field in joint space and figure 3.7c shows the potential field in Cartesian space.

Figure 3.8 is similar to figure 3.7, but instead of the upper solution, the lower solution, Q_{T_2}, is selected.

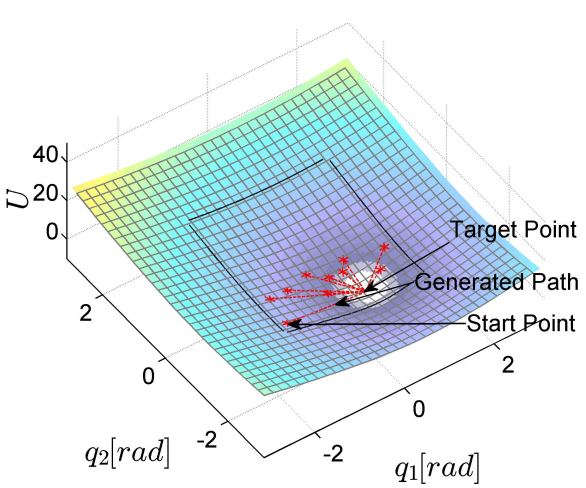
The difference between the paths generated in joint space (figure 3.7a and figure 3.8a) and the path generated in Cartesian space (figure 3.6a) can be easily observed visually. The paths generated in joint space by the potential field planner do not cross the limits of the workspace, when no obstacle is present. However, it is not guaranteed that the workspace limits will not be crossed when obstacles are also considered.

When one or more obstacles are present, these are avoided by the robot, and so the presented paths are modified, in order to avoid collisions. This way, an obstacle can cause the violation of the workspace path. The path planner generates paths that are outside of the workspace, if it has to avoid an obstacle. This example is shown in figure 3.9.

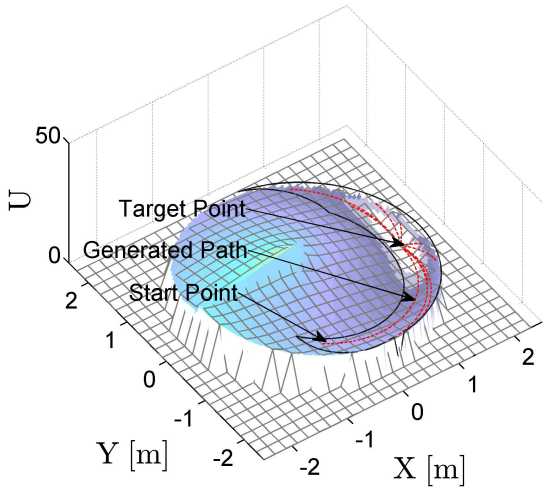
The presented scenario proves the need for integrating workspace limits in the potential field path planner, and that planning in joint space is not a general solution. In Khatib 1986 the joint limits are integrated in the path planing algorithm, but the other two workspace limits have not. Other publication dealing with workspace limits in the case of the APF method have not been found. Although in reality parts of the path outside the workspace cannot be executed, it is best to handle such a situation at the path planner level, and not let the robot controller handle the false reference coordinates generated by the path planner. In order to handle invalid paths in the path planning algorithm, the algorithm has to be aware of the robot motion limitations. This is only possible by integrating the workspace limits as constraints in the path planning algorithm.



(a)



(b)



(c)

Figure 3.8: Path planning in joint space using lower solution, represented (a) in Cartesian space, (b) in joint space, with target potential function, (c) in Cartesian space, with target potential function

3.2 Why the APF Planner?

The APF method is deterministic, non-iterative and does not require large computational power, hence it is well suited for real-time execution. These advantages of the method outweigh its disadvantages.

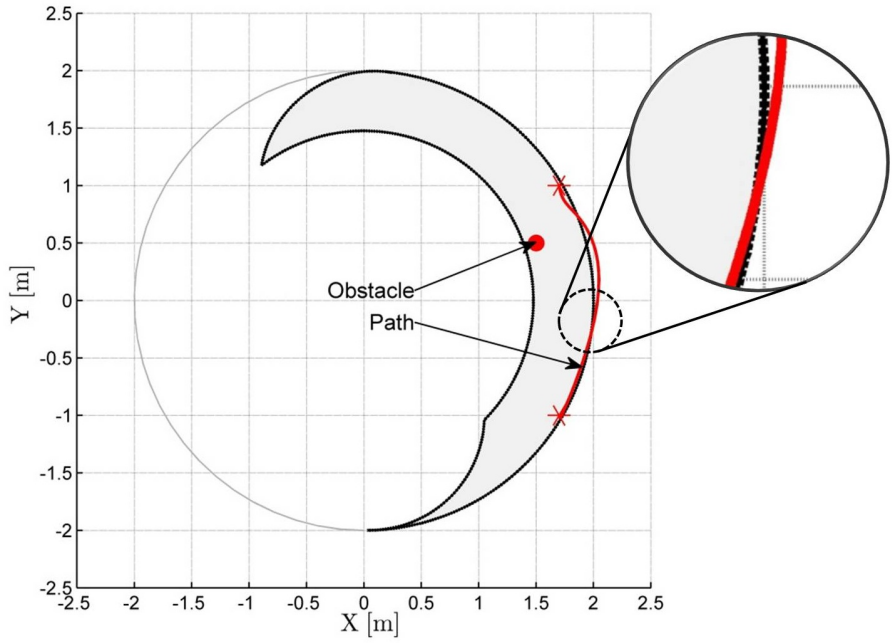


Figure 3.9: Path planning in Cartesian space with obstacle present. In the presence of the obstacle the planned path violates the workspace limit.

Their major drawback is the local minima problem. In some cases these algorithms fail to find the global minimum point (i.e. the target point for the TCP). Solutions to this problem is suggested in Bell et al. 2005; Wahid et al. 2008; Zou et al. 2003; Yun et al. 1997; Chatterjee 2011; Zhang et al. 2006. For the APF method the local minima problem is caused by unfortunate alignment of obstacles and target point. Examining the situation more closely leads to interesting possibilities. The obstacle in the case of pHRI is the human participant. If the alignment of the human-obstacle and the target point impedes the robot from finding a path from the current position to the target point, the human participant in the interaction will recognize this scenario. Very likely it will be perceived as he or she is in the way of the robot, and will change position, similarly to human-human interaction, when one human is in the way of another human, the first one will give way to the later, assuming normal interaction scenarios. This eliminates the major disadvantage of local path planning methods for path planning in pHRI context.

4 Contributions

4.1 Workspace Limits of a Robot

4.1.1 Workspace Determination

The workspace of a robot is defined by its joints, linkages, and by the way these are interconnected. Their interconnection is the key to how the joints and linkages influence the workspace. If the robot is broken down to its structural elements, each structural element has input and output coordinate systems, though which it is interconnected with the other structural elements. A set of reachable points can be obtained for every output coordinate system, relative to the input coordinate system.

This set of reachable points is very similar to a workspace for just one joint (e.g. the set is a circle, or an arc of a circle, if the structural element is a rotational joint). Let this set of reachable points called a workspace module. An iterative way of generating the workspace of the robot can be defined based on these modules.

These workspace modules and the way they are interconnected define the workspace of a robot. If the structural elements are connected in:

- series, the workspace is generated by the Minkowski sum of the workspace modules
- parallel, the workspace is generated by the intersection of the workspace modules

Combining two workspace modules gives a new workspace module. It is equivalent to the workspace module of the substructure formed by the two

structural elements. Combining all workspace modules gives the workspace of the robot.

This approach is well suited for the comparison of workspaces of modular reconfigurable robots, but not only. It has been published in Brisan et al. 2011. It emphasizes the role of different structural elements in the overall workspace.

Applying this approach to a modular reconfigurable parallel manipulator demonstrates best its functionality. Figure 4.1 shows the modules and the possible configurations of the Partner modular reconfigurable robotic system. As the figure shows the basic modules are connected serially, to build up two types of legs. These legs are then connected in a parallel manner to build up a configuration of the Partner robot.

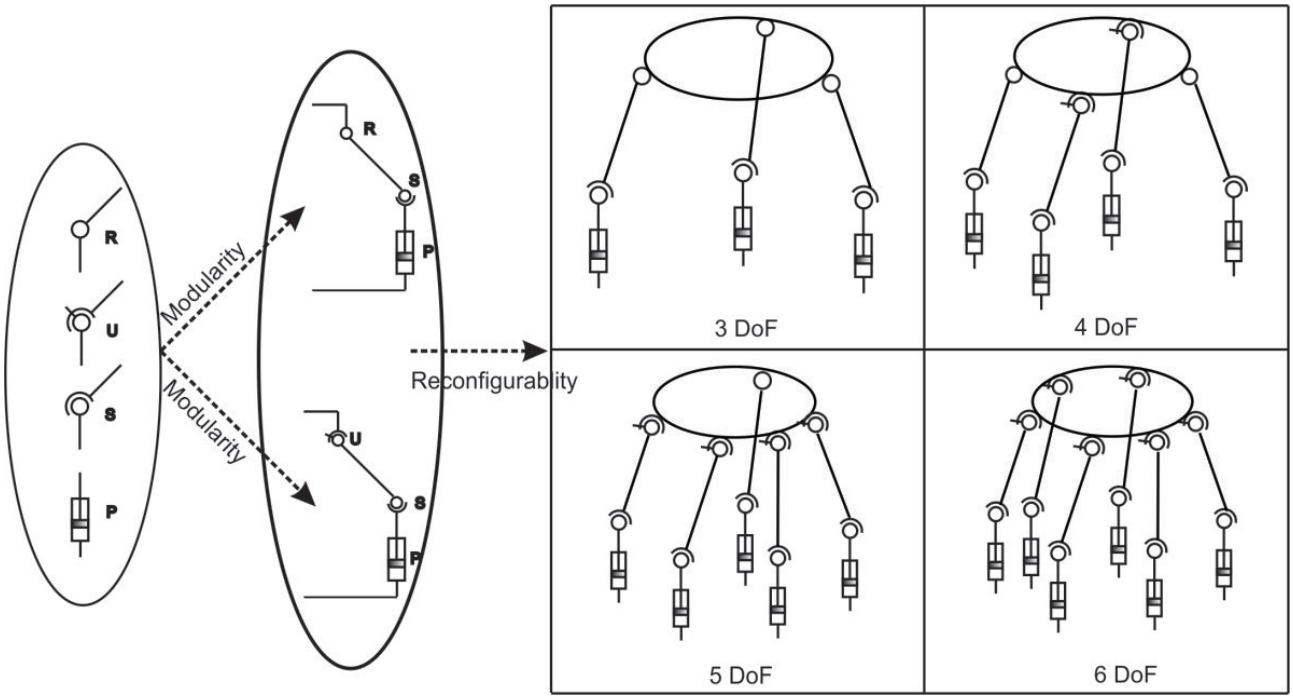


Figure 4.1: Modularity end Reconfigurability properties of the Partner robots

Figure 4.2 presents the PSU (Prismatic-Spherical-Universal) leg of the Partner robot in detail. This leg, when viewed independently, is in fact a serial kinematic chain, constructed from a prismatic joint module, a spherical joint module and a universal joint module. As such, is has 6 DoF.

By taking the workspace of each module, and adding them, using the Minkowski sum, iteratively, the workspace of this leg module can be obtained.

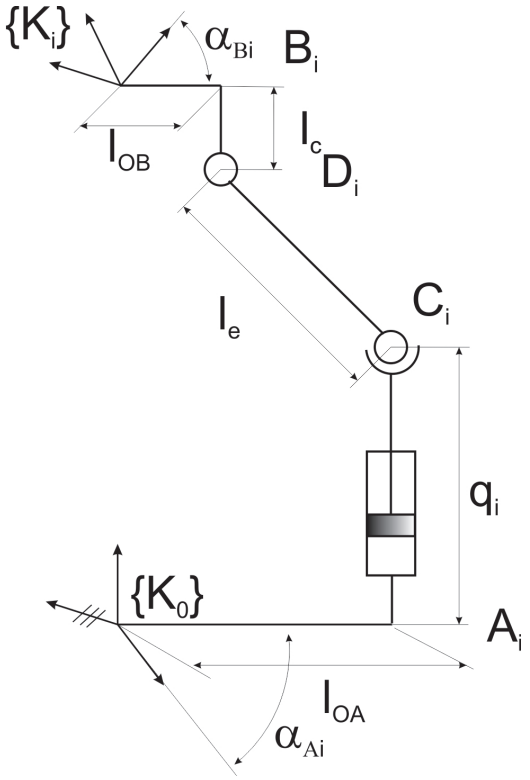


Figure 4.2: The PSU leg of the Partner robot

In figure 4.3a the structure of the universal joint is presented. This joint has two rotational degrees of freedom. Let these two rotations be around the x-axis, ϕ_x and the y-axis, ϕ_y. The workspace of its theoretical model is the surface of a sphere.

$$\begin{cases} x = x_0 + r \cdot \cos\phi_x \cdot \sin\phi_z \\ y = y_0 + r \cdot \sin\phi_x \cdot \cos\phi_z \end{cases} \tag{4.1.1}$$

However, the motion amplitudes of this joint are limited. This can be observed in figure 4.3b. Furthermore, the amplitude of the rotational motions are dependent on each other. The interdependency of the motion amplitudes is described by:

$$-\gamma_l \leq \arccos\left(\frac{\tan(\phi_y)}{\tan(\phi_x)}\right) \geq \gamma_l \tag{4.1.2}$$

The geometrical interpretation of this equation is presented in 4.4. The area on the surface of the sphere which is marked with gray represents the workspace

of this universal joint. The surface is discretized to a set of points. This set of points represents the discretized workspace module.

(a) Theoretical Structure (b) Universal joint module

Figure 4.3: Universal joint

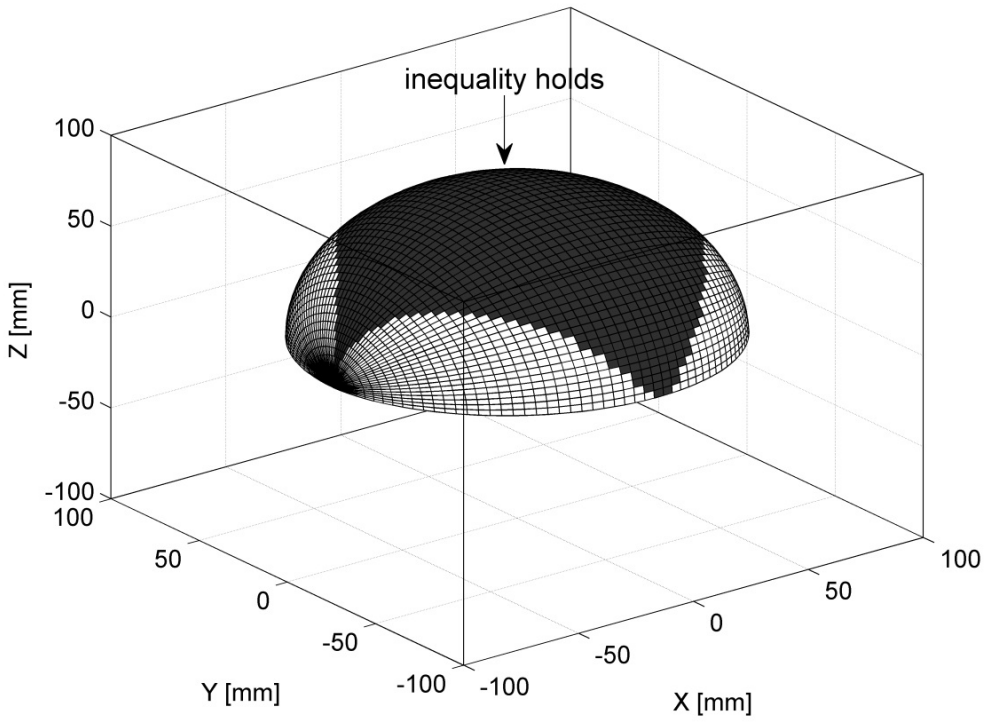


Figure 4.4: The workspace of the universal joint

A similar procedure can be applied to the prismatic joint and the spherical joint found in the structure of the PSU leg. After all workspace modules have been identified, their Minkowski sum generates the workspace of the PSU leg. This workspace is represented in figure 4.5. Similarly the workspace of the PSR

leg can be also generated, having in mind, that instead of the universal joint it has a rotation joint in its structure.

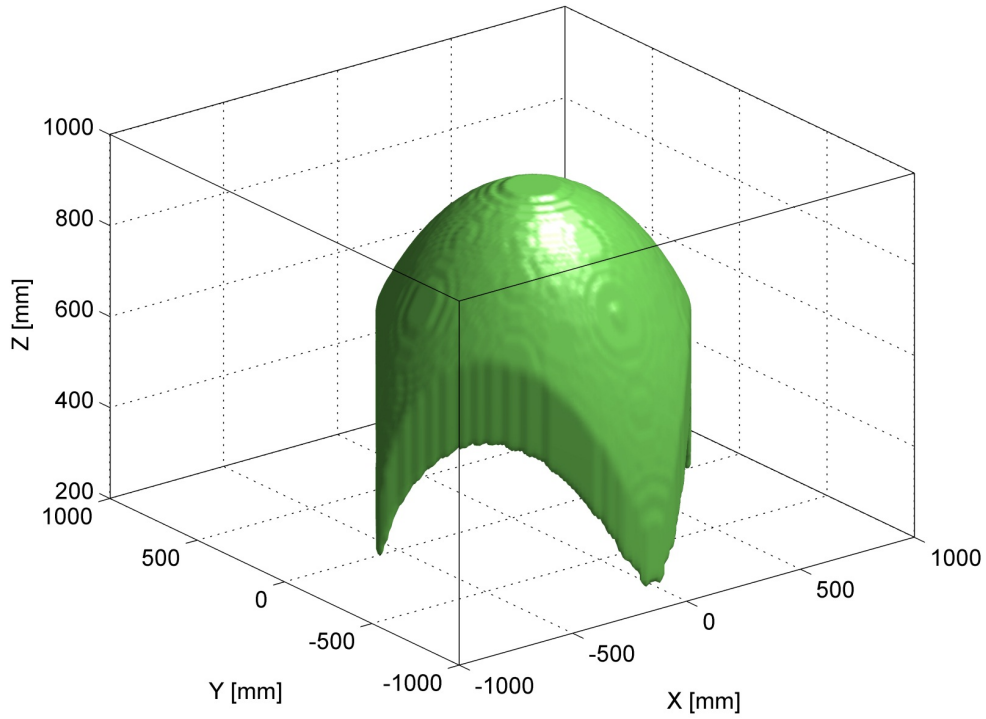


Figure 4.5: Workspace of the PSU leg

Six of these legs are found in the structure of the 6 DoF configuration of the Partner robot. These legs are interconnected in a parallel manner. By taking the intersection of the workspaces of the six legs, the workspace of the Partner robot is obtained. This procedure is shown in figure 4.6. A section of all the workspaces has been taken at a given Z coordinate, and the figure shows how the intersection of the leg workspaces for the workspace of the robot.

The workspace in 3 dimensions of the 6 DoF configuration of the Partner robot is shown in figure 4.7. The cube inside the workspace is the largest cube that can be inserted in the workspace. It offers a quantification of the workspace size, the length of an edge of the cube is $130mm$

The presented procedure can be used to generate the workspace of parallel or serial robots. It is well suited for modular robots, but not only. It offers a new insight to the workspace limits. The exterior boundaries of the workspace can

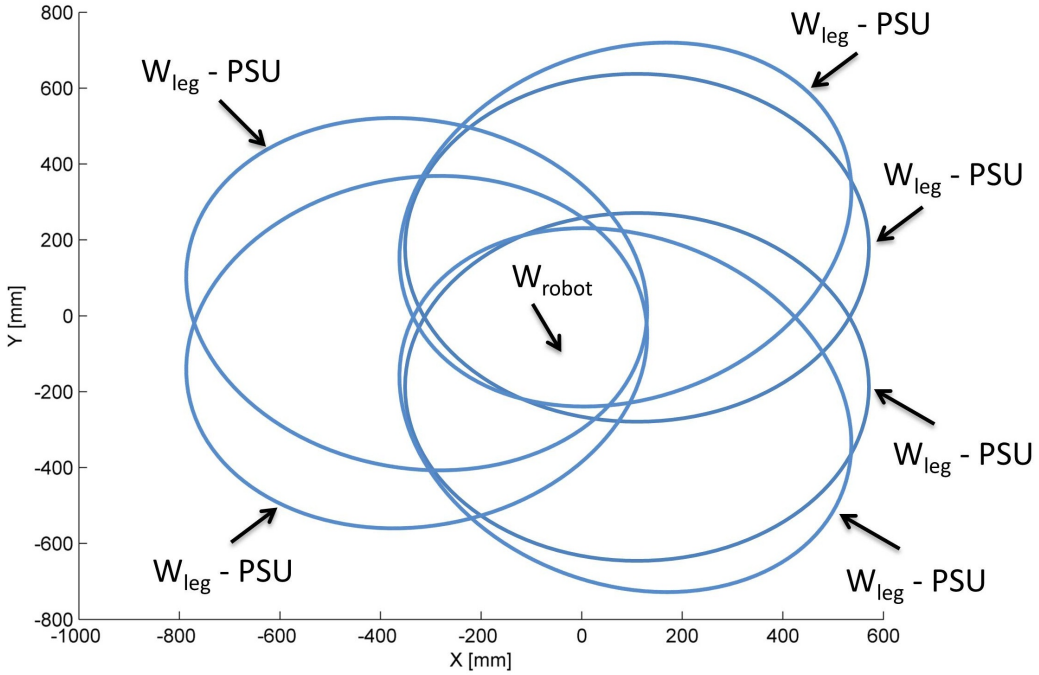


Figure 4.6: Section of the workspace of the 6 legs and the workspace of the robot

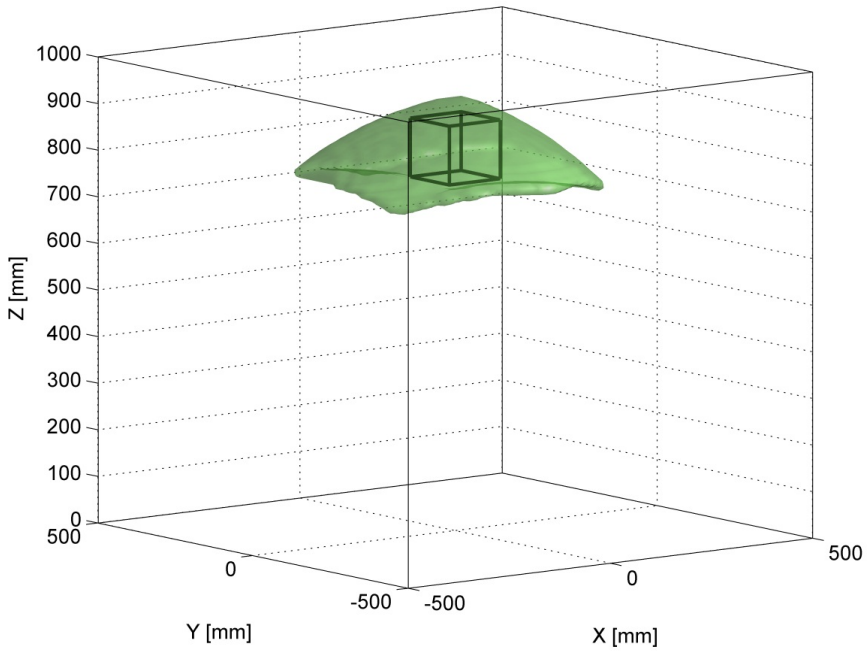


Figure 4.7: Workspace of the 6 DoF Partner robot

be traced back to motion limits of the individual joints found in the structure of the robot. It is more advantageous to include these limits independently when using local path planning algorithms. This way the computation burden for calculating distances for between non-regular surfaces can be avoided.

4.2 Workspace Limits in the APF Method

The integration of all workspace limits guarantees that the path planner will not plan (or will not attempt to plan) paths outside the workspace of the robot. A path planed outside the workspace, from the point of view of the robot controller, means that the controller receives a reference that cannot be executed. This situation surly leads to a deadlock, on the level of the robot controller. Damage to the robot is unlikely, but possible, if the workspace limits are not handled correctly by the robot controller. It is best to handle workspace limits by the path planner, this way only executable (valid) paths are computed.

The APF planner operates based on potential fields. In order to integrate workspace limits in the APF planner, these have to be formulated as potential functions. Similarly to obstacles, these should repel the TCP of the robot. The reason why the workspace limits are treated independently, and not treated simply as obstacles, is that these limits, unlike obstacles, have to be avoided only by the TCP, and obstacles have to be avoided by the whole robot structure. These limits have been defined differently as obstacles.

The three workspace limits are considered in this research, based on Brisan et al. 2011, are:

- joint limits,
- geometrical workspace limits,
- singularities.

In some cases other workspace limits may be considered, due to the surroundings of the robot. (i.e. the robot is placed near a wall). Such a limit can be

either treated and modeled as an obstacle, since it has to be avoided by the entire structure of the robot, not only by the TCP or the joint limits or the geometrical limit can be modified, considering the surroundings of the robot and also the structure of the robot.

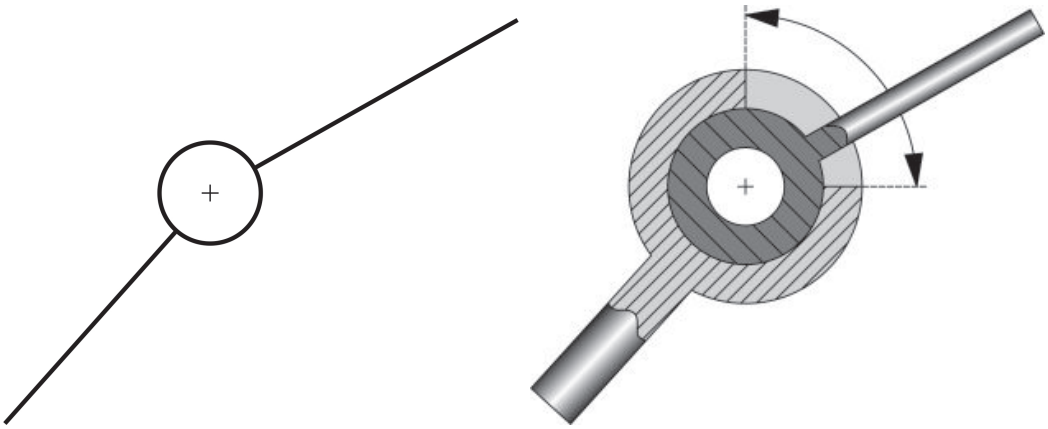
In order to integrate workspace limits in the potential field planner these have to be formulated also as potential functions.

4.2.1 Joint limits

Joint limits are mechanical limits of the joints in the structure of a robot. In many theoretical models of mechanisms and robots these limits have not not taken into consideration. Although the theoretical models of these joints do not present any motion limits their physical counterpart can present motion amplitude limits. Figure 4.8 presents this applied for a rotational joint.

The potential function, formulated as a polynomial function in Khatib 1986, attached to these motion limits has been reformulated as a sigmoid function. These limits can be formulated mathematically in the form of inequalities.

$$q_{i_{min}} < q_i < q_{i_{max}} \tag{4.2.1}$$



(a) Theoretical Rotational Joint Schematic (b) Rotational joint schematic

Figure 4.8: Rotational joint schematics

Where q_i denotes the instantaneous value of the joint parameter, $q_{i_{min}}$ denotes the lower limit of the joint parameter, and $q_{i_{max}}$ denotes the upper limit of the joint parameter.

These limits can also be observed in figure 3.5. The joint limits delimit a rectangular area of the joint space figure 3.5a. This rectangular area in joint space becomes the workspace in Cartesian space presented in figure 3.5b. The limits inside the circular area represent the joint limits, while the circumference of the circle represents the geometrical workspace limit, presented in the next subsection.

As equation (4.2.1) shows, the limits $q_{i_{min}}$ and $q_{i_{max}}$ refer only to one joint variable, q_i. All joint variables have different limits, which are independent of each other. The vector Q can be written as:

$$Q = [q_1 \dots q_n]^T \tag{4.2.2}$$

Similarly the vectors

$$Q_{min} = [q_{1_{min}} \dots q_{n_{min}}]^T \tag{4.2.3}$$

and

$$Q_{max} = [q_{1_{max}} \dots q_{n_{max}}]^T \tag{4.2.4}$$

can also be defined.

This way, equation (4.2.1), in vector form, becomes:

$$Q_{min} < Q < Q_{max} \tag{4.2.5}$$

The vector Q describes the current pose of the robot in joint space. It could also be denoted as Q_{TCP}, since it defines the TCP coordinates in joint space. Using the direct kinematic function f it can be converted to Cartesian space. The size of Q is $n \times 1$ where n is the number of DoF of the robot.

The vectors Q_{min} and Q_{max} and the inequality (4.2.5) define a region in joint space. Since the variable in inequality (4.2.5) is Q_i, the joint coordinates, it has

been concluded that the linkages of the robot have no effect on this workspace limit, just the mechanical joint limits. This way these limits have to be avoided only by the TCP. Hence the main difference between obstacles and joint limits: Obstacles have to be avoided by the whole robot structure, joint limits have to be avoided only by the TCP.

The fact that joint limits are positions that should not be reached by the TCP shows, that these should be modeled as repellent potential fields (i.e. potential sources). The fact that they are independent of each other, in the case of serial robots, determines that they have to be be modeled as potential sources independently (a potential function is attached to each joint).

$$U_{\text{joint}_i} = U_{\text{joint}_i}(q_i) \tag{4.2.6}$$

Where U_{joint_i} represents the potential function attached to one joint.

In the case of parallel robots, passive joints can be found, with more then one DoF. There are two difficulties when dealing with such joints.

- They are not active joints, so their actual coordinates depend on the coordinates of the active joints.

- In the case of a universal joint, or a spherical joint the there are more then one joint variables, and the limit values of the joint variable depends on the other joint variable(s).

An example for a joint with more then one degrees of freedom is the universal joint, often found in the structure of parallel robots. Figure 4.3 presents this type of joint. The motion limits of this joint are described by equation 4.1.2. It can be observed that the form of equation (4.1.2) is very similar to equation (4.2.1). This way the motion limit of the universal joint has been written using only one variable, instead of two, hence it can be treated similarly as in equation 4.2.1, by substituting $-\gamma_l$ with $q_{\min}$ and γ_l with $q_{\max}$.

The marginal values represent the joint limits, which have to be avoided. These limits have to repel the robot.

Since the there are two marginal values of q_i to avoid, U_{joint_i} is composed from two potential fields, $U_{\text{joint}_{i_{\min}}}$ attached to the inferiors joint limit $q_{i_{\min}}$, and $U_{\text{joint}_{i_{\min}}}$ attached to the superior joint limit $q_{i_{\max}}$.

$$U_{\text{joint}_i}(q_i) = U_{\text{joint}_{i_{\min}}}(q_i) + U_{\text{joint}_{i\max}}(q_i) \tag{4.2.7}$$

The potential function has to be zero or close to zero when the actual joint coordinate, q_i, is not near the joint limit value.

$$U_{joint_{i_{max}}}(q_i) < \epsilon \quad \text{where} \quad \epsilon \cong 0 \quad \text{and} \quad q_i < q_{i_{max}} - \delta \tag{4.2.8}$$

$$U_{joint_{i_{min}}}(q_i) < \epsilon \quad \text{where} \quad \epsilon \cong 0 \quad \text{and} \quad q_i < q_{i_{min}} + \delta \tag{4.2.9}$$

Where δ expresses the distance from the joint limit, above this distance the path planning should not be effected by the joint limit constrains.

Obstacles usually are modeled by polynomial potential functions. In Ren et al. 2007 sigmoid functions have been suggested to model obstacles instead of polynomials. Sigmoid functions can also be used to model joint limits, but for different reasons then in the case of obstacles. The general form of a sigmoid function is presented in equation (4.2.10). Defining the potential function as a sigmoid function corresponds to the general requirements for potential functions stated in Al-Sultan et al. 1996.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.2.10}$$

Adopted for the joint limiting equation (4.2.10) takes the form:

$$U_{joint_{i_{\min}}}(q_i) = \frac{1}{1 + e^{-\alpha - \beta \cdot \left(q_i - q_{i_{\min}}\right)}} \tag{4.2.11}$$

$$U_{joint_{i_{\max}}}(q_i) = \frac{1}{1 + e^{-\alpha - \beta \cdot \left(q_{i_{\max} - q_i}\right)}} \tag{4.2.12}$$

The two parameters α and β help scale the sigmoid function. The values for α and β are not the same for equations (4.2.11) and (4.2.12). Their values can be calculated as follows.

Let $U_{\text{Threshold}}$ represent a threshold value that corresponds to zones in the workspace that cannot be reached by the robot, because it would violate the joint limit condition. Since the sigmoid function returns values in the interval [0 1) let the threshold value be:

$$U_{\text{Threshold}} = 0{,}9 \tag{4.2.13}$$

When combining the usage of different types of potential function and not all return values from the interval [0 1), this value can easily be scaled up with a weighing factor. Let ϵ be a value of the potential function that is close to 0, and it is considered that it does not influence the path planning.

$$\epsilon = 0{,}01 \tag{4.2.14}$$

Let q_d be the angle at which the joint starts to apply a virtual force, to guide the robot away from the workspace limit. (Actually it is a virtual torque for rotational joints, but the generalized concept of force has been considered.) In order to use as much space of the workspace as possible this should be close to the joint limits $q_{i_{\min}}$ or $q_{i_{\max}}$. Because the calculations are similar for case of the inferior limit and the case of the superior limit, only one case is presented. Based on equations (4.2.11) and (4.2.14):

$$q_d = q_{i_{max}} - \delta \tag{4.2.15}$$

Starting from the limit values, equations (4.2.16) and (4.2.17) can be written.

$$U_{joint_{i_{min}}}(q_d) = \epsilon \tag{4.2.16}$$

$$U_{joint_{i_{min}}}(q_d) = U_{\text{Threshold}} \tag{4.2.17}$$

The parameters α and β can be determined from the equation system (4.2.18).

$$\begin{cases} \frac{1}{1+e^{-\alpha-\beta\cdot\left(q_d-q_{i_{min}}\right)}} = \epsilon \\ \frac{1}{1+e^{-\alpha-\beta\cdot\left(q_{i_{min}}-q_{i_{min}}\right)}} = U_{\text{Threshold}} \end{cases} \tag{4.2.18}$$

Thus results:

$$\begin{cases} \alpha = -\ln\left(U_{\text{Threshold}}^{-1} - 1\right) \\ \beta = \ln\left(\epsilon^{-1} - 1\right) \cdot \frac{1}{\left(q_d - q_{i_{min}}\right)} - \alpha \end{cases} \tag{4.2.19}$$

This way the parameters of the potential function (4.2.7) can be assigned with less empirical considerations. The effects of this parameter assignment on the potential function are represented in figure 4.9. The graphical representation of the joint limit potential function (4.2.7) can be done in 2 dimensions, where the two axes of the coordinate systems are $U_{joint_i}(q_i)$ and q_i. For comparison a polynomial joint limiting potential is also presented. The effect of the parameter α can be observed on the difference between the $sigm1$ and $sigm3$ curve. The curve $sigm1$ has a larger α parameter then the curve $sigm2$. The effect of the parameter β can be observed on the difference between the $sigm1$ and $sigm2$ curves. The curve $sigm1$ has a larger β parameter as the curve $sigm2$. The *poly* curve represents a polynomial potential function, shown for qualitative comparison with the sigmoid functions. The polynomial potential function represents the most used potential function in the sate of the art.

Formulating the potential function of the joint limit as a sigmoid function, not as a polynomial function, has two advantages:

- On advantage is that the values returned by the function are between in the interval [0 1]. This is more advantageous then [0 ∞] in the case of polynomial potential field, since large numbers (and the concept of infinity) can be hard to represent on some computational systems. It is also possible to scale up this interval, with a simple weighing factor, to allow combined use with other types of potential functions.

- Another advantage of the function lays in the values it returns for input of values that are outside the interval [$q_{i_{\min}}$ $q_{i_{\max}}$]. Based on the return

value alone it can be decided, that the currently analyzed position should not be occupied by the robot, and no other check is necessary. This is helpful when the defined $[q_{i_{\min}}\ q_{i_{\max}}]$ interval is smaller than the range of motions actually executable by the robot. In the most cases of robot usage software limits permit less motion that actual hardware limits.

The approach also presents two disadvantages:

- One disadvantage of the method is that, although the potential function is better suited for path planning then polynomial functions, its gradient can take large values based on the parameters α and β. To overcome this disadvantage the virtual force, $\vec{F}_{\text{joint}_i}$, has not been defined as the negative

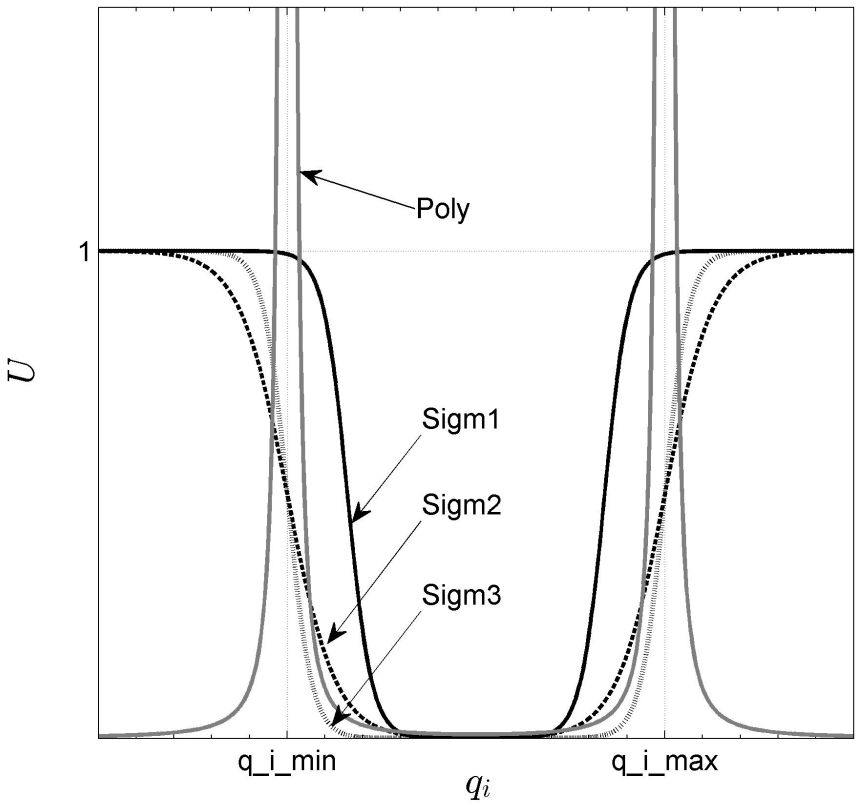


Figure 4.9: Comparison of sigmoid and polynomial potential functions

gradient of the potential function, as presented in equation (3.1.3). It has been defined as:

$$\vec{F}_{\text{joint}_i} = U_{joint_i} \cdot \hat{r} \tag{4.2.20}$$

Where $\hat{r}$ is the unit vector of the relative distance vector between the joint limit and the current joint coordinate. In this case, since this is a one dimensional case, considering only one joint, $\hat{r} = 1$, but this definition of the force is used also in higher dimensional cases. Defining the virtual force this way, brings the definition of the virtual force closer to the formulation presented in equation (3.1.2), which is dimensionally and physically correct.

- The second disadvantage is that computation of the sigmoid potential function can take more time than the computation of a polynomial potential function. The difference is considered to be too little to influence the overall cycle time.

Considering a serial robot with n DoF, there are n independent potentials (each being the superposition of two potential fields for minimum and maximum, as shown in figure 4.9) limiting the n joints. Considering the limits for the whole robot a vector can be written with n components.

$$\vec{F}_{\text{joint}} = \begin{bmatrix} F_{\text{joint}_1} & \dots & F_{\text{joint}_n} \end{bmatrix}^{\text{T}} \tag{4.2.21}$$

The virtual forces, the case of rotational joints, are in fact virtual torques applied virtually to the robot joints, but the generalized concept of force is used to preserve consistency. Care must be taken when superpositioning potential fields of different kinds. This virtual force has been defined in the joint space of the robot and expresses the joint limit avoidance component of the path planning. In this form it can be used for path planning in joint space, but in order to use it for planning in operation space it has to be converted to

Cartesian space. Conversion between the joint space and the Cartesian space can be done using the Jacobi matrix of the robot.

$$^{\mathcal{W}}\vec{F}_{\text{joint}} = J \times {}^{\mathcal{C}}\vec{F}_{\text{joint}} \tag{4.2.22}$$

Where the superscript $\mathcal{W}$ refers to a Cartesian space value, the superscript $\mathcal{C}$ refers to a joint space value, and J represents the Jacobi matrix of the robot.

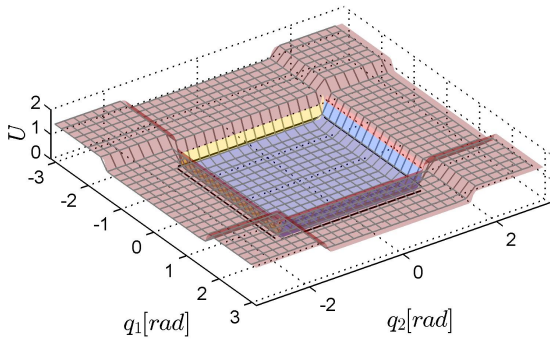
Applied to the 2 DoF serial manipulator the joint limit potential function can be represented in three dimensions (figure 4.10). In two dimensions (X and Y axis) the workspace of the robot is represented, the third dimension (the Z axis) is used to represent the value of the potential field throughout the workspace. Figure 4.10 shows the potential field inside the robot workspace due to joint limits. An important aspect can be observed: joint limit potential, in Cartesian space, limits a large part of the workspace limits, but not all.

In figure 4.10c it can be observed that, in joint space, the joint limit potential surrounds the workspace of the robot, as presented. However, this is not the case in Cartesian space, figures 4.10b and 4.10c. This may seem contradictory, but it is explained by the fact that points that are outside the geometrical workspace of the manipulator (the outer circle in Cartesian space) cannot be mapped in the joint space. Thus an additional limiting potential must be defined.

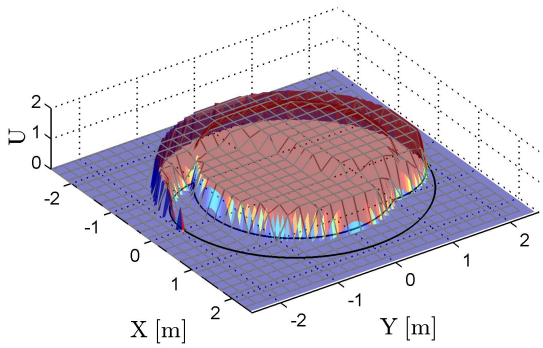
4.2.2 Geometrical Workspace Limit

Besides joint limits another important workspace limit is a geometrical limit. In this work the geometrical workspace limit potential is introduced and it is formulated as a sigmoid function.

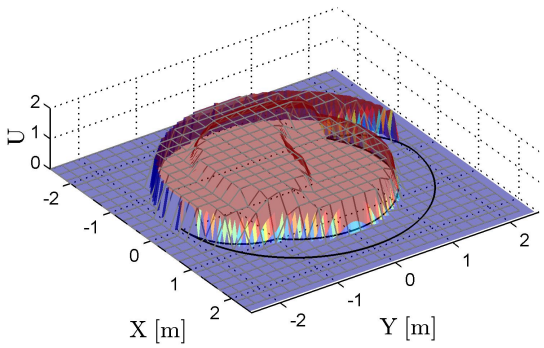
This type of limit can be formulated in the form of an equation, and it is formulated in Cartesian space. This limit is explained by the fact that the maximum distance between the base coordinate system of the robot and the TCP is equal to the sum of the length of the linkages of the robot (when considering only rotational joints). The geometrical workspace of the robot has to be known in order to attach a potential function to it. The potential field which models it, should depend on the shape of the workspace. For most 6



(a)



(b)



(c)

Figure 4.10: Joint limit potential in the workspace of the robot (a) in joint space, (b) in Cartesian space, lower solution, (c) in Caretsian space, upper solution

DoF manipulators the geometrical limit of the workspace is the surface of a sphere. For the 2 DoF serial manipulator it is a circle. Other shapes, such as a cylinder for the SCARA type manipulator can also be considered. The shape of the geometrical workspace depends on the type of joints in the structure of the robot (rotational or translational), the size of the workspace depends on the length of the robot linkages.

Similarly to the case of join limit potential functions, constraints for the potential function can be formulated. Since it is also a limit that should be

avoided it should be modeled as a potential source, similarly as an obstacle, and it is dependent on the current pose of the robot.

$$U_{\text{Geom}} = U_{\text{Geom}}(X) \tag{4.2.23}$$

Where X denotes the coordinates of the TCP. The size of X is $1 \times n$ where n is the number of DoF of the robot. The potential value has to be zero, or close to zero, when the robot is far from the limit of the workspace.

$$U_{\text{Geom}}(X) < \epsilon \quad \text{where} \quad \epsilon \cong 0 \quad \text{and} \quad X < X_{\text{lim}} + \delta \tag{4.2.24}$$

As in the case of the joint limit, it is advantageous to define this potential function as a sigmoid function. Advantages of such a formulation are presented in subsection 4.2.1.

$$U_{\text{Geom}} = \frac{1}{1 + e^{-\alpha - \beta \cdot D(X_{TCP}, \zeta)}} \tag{4.2.25}$$

Where the constants α and β help scale the sigmoid function, $D\left(X_{TCP}, \zeta\right)$ represents the distance between the current TCP position, X_{TCP} and the curve or surface ζ, representing the geometrical workspace limit.

In order to benefit from the effects of this potential a virtual force is defined, which can be used in the path planning algorithm to repel the TCP of the robot from the geometrical limit of the workspace.

$$\vec{F}_{\text{Geom}_i} = U_{\text{geom}} \cdot \hat{r} \tag{4.2.26}$$

Where $\hat{r}$ is the unit vector of the relative distance vector between the geometrical workspace limit and the current TCP position.

In order to visually represent the above equations, these have been applied for the 2 DoF serial manipulator. In figure 4.11 the visual representation of the function (4.2.26) is shown. This workspace limit potential completes the joint limit potential and this way, the two potential functions delimit completely the workspace of the robot, and it is not anymore possible for the path planner to plan paths outside the workspace. Figure 4.12 shows the workspace of the

manipulator in upper configuration (figure 4.12a) and in lower configuration (figure 4.12b). The two potential functions delimit the workspace completely.

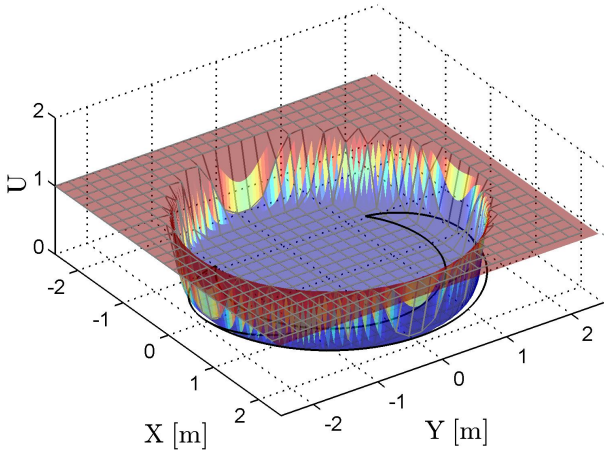
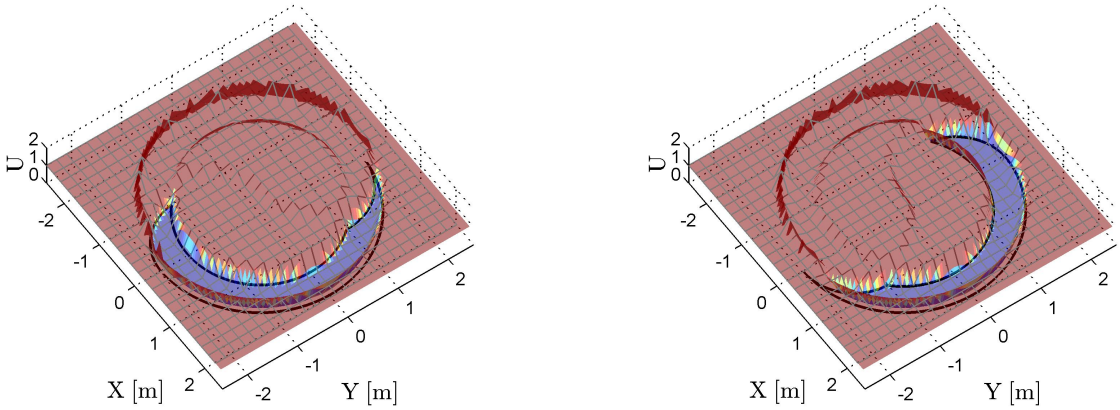


Figure 4.11: Geometrical workspace limit potential



(a) (b)

Figure 4.12: The sum of the geometrical workspace limit potential and the joint limit potential in Cartesian space for (a) lower solution and (b) upper solution

4.2.3 Singularities

In this thesis the singularity avoidance potential is introduced and it has been formulated as a potential function. In singular positions the robot can become uncontrollable.

The path planner should avoid planning paths that lead through singular positions. Singular positions are not dependent on the environment of the robot, just on the structure of the robot. This way they can be computed offline, but they must be avoided online.

Since singularities can also be considered poses of the TCP which should be avoided, these poses must be modeled as potential sources. Avoiding singularities is similar to avoiding obstacles, but it is important that the potential functions of obstacles and singularities are treated separately, because singularities have to be avoided only by the TCP of the robot, while obstacles have to be avoided by the entire structure of the robot, including the linkages. Since the kinematic structure of the robot is known, all singular points and regions are considered to be known. A way to find singular poses of a robot has been described in Brisan et al. 2011 and Merlet 2002. Singular poses, in most cases, can be well identified for the robots, and so, a list of singular points (or regions) in the workspace can be defined. Let Q_s be a point in the joint space of the robot for which

$$|J| = |J(Q_s)| = 0 \tag{4.2.27}$$

Where J represents the Jacobi matrix of the robot and $|J|$ represents its determinant. For the most widely used manipulator structures the size of J is $n \times n$, where n represents the number of DoF of the robot. Note that the pseudoinverse of the Jacobi matrix might be needed to calculate the determinant in some cases, when the the Jacobi matrix is not quadratic. The Jacobi matrix of a robot is dependent from the robots current pose. Since this pose should be avoided it is treated similarly to an obstacle. A potential field, described by a potential function, is attached to this point. The potential function, similarly

to joint limits and geometrical workspace limit, is defined as a sigmoid function, and it is a potential source (which repels the TCP).

$$U_{\text{Sing}_i} = U_{\text{Sing}_i}(Q_{\text{TCP}}) \tag{4.2.28}$$

$$U_{\text{Sing}_i}(Q_{\text{TCP}}) = \frac{1}{1 + e^{-\alpha - \beta \cdot D(Q_{\text{TCP}} - Q_s)}} \tag{4.2.29}$$

Where $D(Q_{\text{TCP}}, Q_s)$ represents the distance between the current TCP pose, Q_{TCP} and the singular pose Q_s. The size of the Q matrices is $n \times 1$, where n represents the number of DoF of the robot. α and β are scaling parameters. These can be used to set the precise range in the joint space which should be avoided. This is useful, because in close vicinity of the singular pose, characteristics of the robot (e.g. dexterity) degrade Merlet 2002. The value of the parameters can be calculated similarly as in equations (4.2.19).

A virtual force, $\vec{F}_{\text{Sing}_i}$ is be defined, which repels the TCP of the robot from the singular poses.

$$\vec{F}_{\text{Sing}_i} = U_{\text{Sing}_i} \cdot \hat{r} \tag{4.2.30}$$

Where $\hat{r}$ represents the relative distance vector between the singularity pose Q_s and the current TCP pose, Q_{TCP}. When $m \in \mathbb{N}$ singular regions exist, the total virtual force repelling from singularities, F_{Sing}, can be defined as:

$$\vec{F}_{\text{Sing}} = \sum_{i=1}^{m} \vec{F}_{\text{Sing}_i} \tag{4.2.31}$$

Since $\vec{F}_{\text{Sing}}$ has been defined in joint space, and it has to be converted to Cartesian space, so it can be summed up with other virtual forces.

$${}^{\mathcal{W}}\vec{F}_{\text{joint}} = J \times {}^{\mathcal{C}}\vec{F}_{\text{Sing}} \tag{4.2.32}$$

Note that in this case, the Jacobi matrix J is dependent on the TCP position, not on the singularity position as in equation (4.2.28).

$$J = J\left(Q_{\mathrm{TCP}}\right) \tag{4.2.33}$$

In order to demonstrate the behavior and the advantages of introducing this potential function the joint limits presented in table 3.1 have been disregarded. The reason for this is, that the presented 2 DoF manipulator has singular poses either at its base, either at the outer contour of the workspace, which has already an assigned potential, the geometrical workspace limit U_{Geom}. This is not the case for a 6 DoF manipulator or for other structures. The 2 DoF manipulator structure is in a singular position when

$$q_2 = k \cdot \pi \quad \text{where} \quad k \in \mathbb{Z}^* \tag{4.2.34}$$

For a specific case, $k = 1$ has been chosen.

$$q_2 = \pi \tag{4.2.35}$$

The singular pose in Cartesian space can be expressed as:

$$P_s = f\left(Q_s\right) = [0\ 0] \tag{4.2.36}$$

Where P_s represents a singular point in Cartesian space, and f represents the direct kinematic function.

Equation (4.2.29) has been applied to obtain the value of the potential field.

The integration of the presented potential functions in the path planning algorithm is presented in section 4.5, how these these perform is presented in chapter 5.

The three presented potential functions delimit areas of the space which cannot or should not be reached (based on the characteristics of the robot). These assure that only valid paths are planned by the algorithm. If humans are also present in the workspace of the robot, forbidden zones should be defined,

which have the role of protecting the human from an accidental collision with the robot.

4.3 Fault-Tolerant Distance Criterion

The safety criteria presented in chapter 2, section 2.3.1, used in the path planning algorithms for human-robot interaction presented in chapter 2, section 2.4 use a "best-case" approach. None of the presented methods consider that the functionality of the robot can be compromised by a fault or malfunction of the robot. The nature of the possible faults is various (e.g. software faults, cable faults, etc.). The key to assuring the safety of the human participant in the interaction is to consider the worst case scenario, where a fault causes robot malfunction. Such an approach has been used to formulate a novel safety criterion that assesses "worst case" situation instead of "best case" or "common case" situation.

The emergency brakes are the only part of the robot which can be trusted in the case of a malfunction. This is due to the reduction of the risk of brake malfunction, sometimes even by using redundant braking systems (even in this case it can be considered that only one brake functionality can be trusted). This fault tolerant property of the braking system, and the fact that the brakes are (or can be) activated when a malfunction occurs has been the starting point of assuring human safety in a "worst case" scenario, in the case robot malfunctions.

4.3.1 Definition of the Fault-Tolerant Distance

A novel safety criterion has been formulated, based on the fact that for the current robot state the emergency brakes have to able to stop the robot before colliding with the human. The key difference between the state of the art and the proposed approach is the ability to guarantee human safety also in the case of a malfunction, a requirement set by safety standards. In order to implement such a criterion is has to be formulated mathematically.

In Brock et al. 2002 the elastic tunnel framework is presented. In this framework it is suggested that the path executed by the robot, should not be described simply as a curve, but as a volume of the operational space which is occupied by the robot during the execution of a path. In the mentioned publication only the task related path of the robot was considered, not the braking path.

In the case of braking the braking path of the robot is of interest. A workspace volume can be attached to this path which corresponds to the volume in Cartesian space occupied by the robot during the execution of the braking maneuver. Although the execution of this braking trajectory is unlikely, it represents the worst-case situation.

Let the volume occupied by the robot during the execution of a braking maneuver be called braking volume, V_{Braking}. If the braking volume does not intersect any obstacles it can be assured, that in a case of an emergency braking maneuver no collisions would take place. This has been translated to a safety criterion.

The distance (or safety clearance) criteria consider the minimum distance between the robot and the human. A novel, fault-tolerant distance criterion is proposed. The fault-tolerant distance criterion considers the distance between the braking volume $V_{Braking}$, and the workspace volume occupied by the human, V_{Human}. This concept, compared to a traditional distance criterion, is presented in figure 4.13. The volume occupied by the robot is presented as the robot convex hull (H_{robot}), $\wp_{task}$ represents the path executed by the robot. $\wp_{\text{Stop}}$ represents the braking trajectory which would be described by the TCP, if the brakes were engaged at X_i. During the execution of the braking maneuver the robot would occupy the volume described by the braking convex hull H_{braking}. The volume occupied by the human is described by the convex hull H_{human}. The convex hulls presented in the figure represent a simple case for illustration, the volume description can be further refined.

The safety criterion is a binary condition, a collision detection between the braking volume and the workspace volume occupied by the human. A safety observer also can be implemented based on this criterion. This is the virtual replacement of the safety fence which is currently used to separate the human

workers from the robot in factories. Separating the path planning from the safety observer has the advantage, that special safety certified hardware and software would be only required by the safety observer. However, the separated components have to use the same safety criterion.

The advantages of outlined approach are:

- The guarantee of human safety even in scenarios when the robot malfunctions.
- The criterion characterizes the current state of the robot, relative to the current state of the obstacle. This way it is well suited both for local and for global planning methods
- The characterization of a state as safe or unsafe, without empirical thresholds.
- Cost reduction due to decentralization (separation of safety observer and path planner).
- It characterizes a state of the robot, not an entire path, this way, it is well suited also for local planning algorithms.

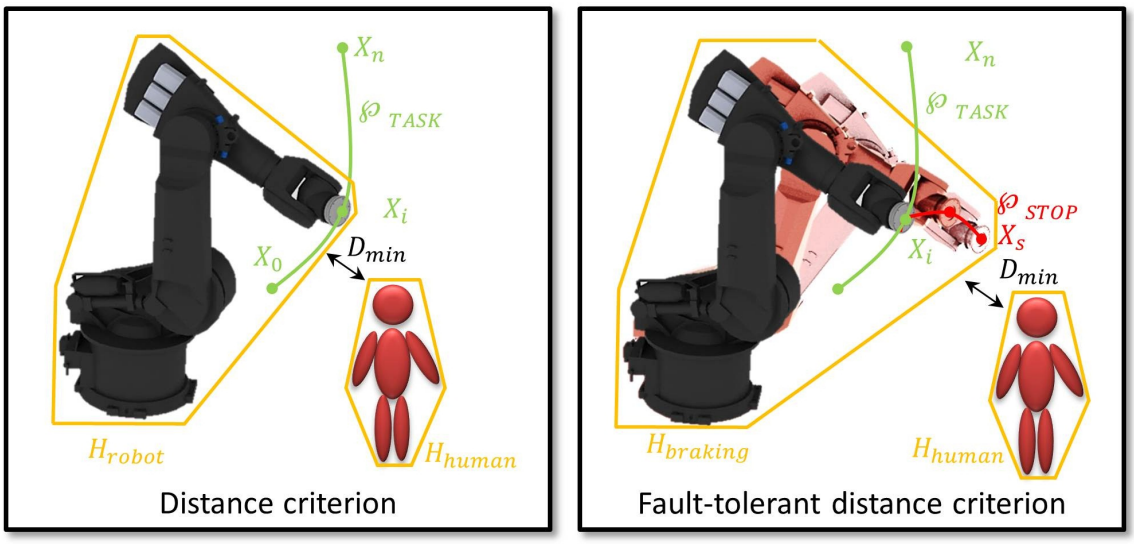


Figure 4.13: Difference between distance criterion and fault-tolerant distance criterion

The disadvantage lies in the difficulty of real-time operation. It it not yet possible to generate braking trajectories in real-time.

A workaround for this disadvantage is based on the fact that the braking trajectories are not effected by the environment, so they can be pre-computed offline. As this would be a very large lookup table its size can be reduced by considering that small changes in the robot state will cause small changes in the braking volume. In such cases one slightly larger braking volume can be used to all braking volumes that are contained in that slightly larger volume. It is believed that this way real-time performance can be achieved.

Another workaround is to use controlled braking (in which case the braking trajectory is controlled) instead of emergency braking, this way the real-time computation of emergency braking trajectories is not necessary. The safety standards permit controlled braking, with the condition, that if it is not working, the emergency brakes must be engaged.

Convex hulls are one way of mathematically describing volumes. Other methods include: Spheres, Bounding Boxes (normal, oriented, or axis aligned), and other methods. For different volume description types different collision detection and distance calculations are recommended. Many of these are real-time capable.

The integration of the Fault-Tolerant Distance in a path planning algorithm is of interest. This functionality avoids the collisions between the two mentioned volumes. This way the path planner can plan paths, which are considered safe. The criterion is evaluated based on the braking volume.

4.3.2 Computation of the Braking Volume

In order to obtain the braking volume of the robot, the braking path has to be known. Calculating the braking path is not the topic of this research. This is also the reason why it has not been presented in chapter 2 State of the Art. The theoretical aspects behind obtaining the braking path have been described in Dietz et al. 2010; Dietz et al. 2011. The computation of the braking path cannot be done in real-time. Since it is not affected by the obstacles in the

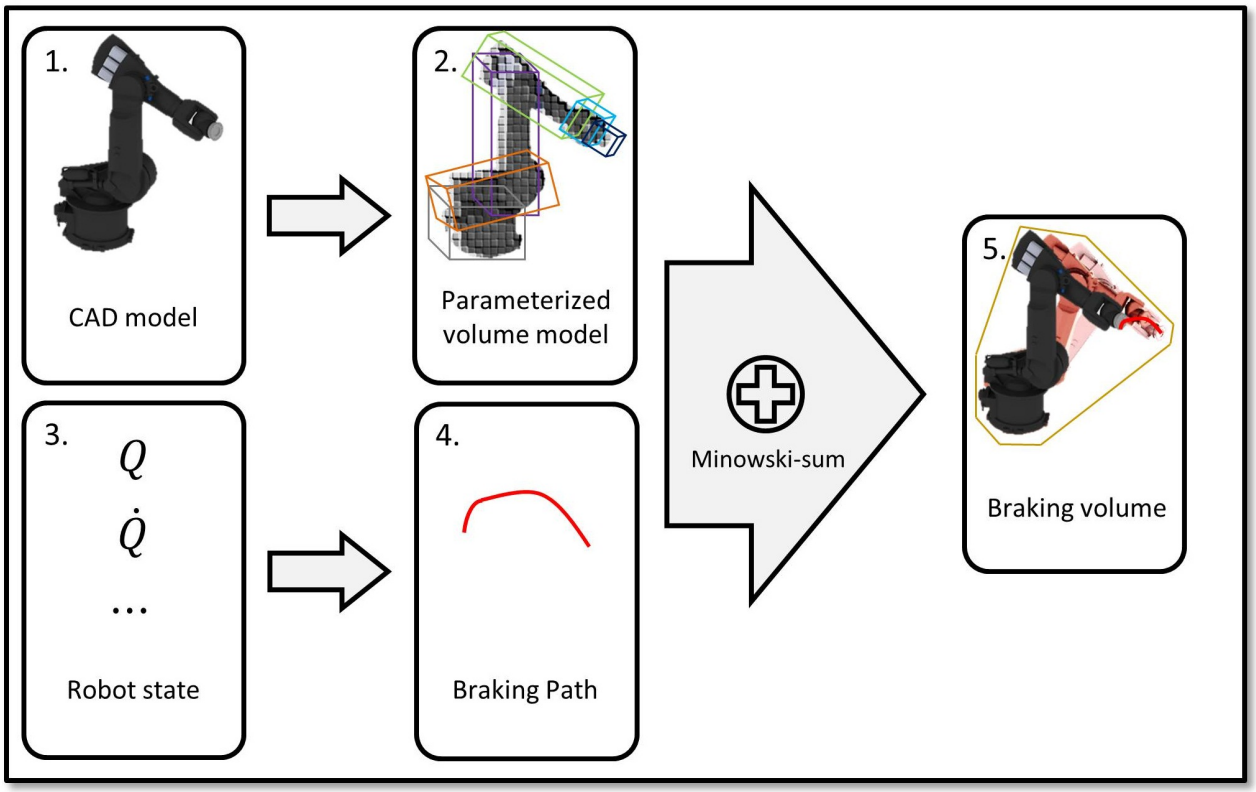


Figure 4.14: Obtaining the Braking Volume

workspace, it can be simulated a priori, defining this way a braking volume database, which can act similarly as a lookup table. Obviously size reduction of this database must be considered.

Figure 4.14 presents visually how to obtain the braking volume.

The following steps have been defined in order to compute the braking volume:

1. The CAD model of the robot is required as input to the process. Most commercially available robots have their CAD model available free of charge. Custom made robots are usually built after their CAD design. This requirement does not limit the applicability of the method. Although there are many CAD model formats, they all have the same aim, describing the parts of the robot accurately (up to micrometers). By describing the parts, the volume occupied by the parts is also defined. Most CAD formats can be converted to almost all other formats, using the correct software tools.

2. CAD models are well suited for manufacturing processes, but they are less advantageous for mathematical models. In order to include CAD models in mathematical models it is a common practice to convert them, loosing this way the precision of CAD models, but gaining simplified mathematical description, and so lowering the needed computational power. There are more ways in which volumes can be described mathematically, all having their advantages and disadvantages. Their principle is to approximate the volume described with a mathematical entity, that is less precise than the CAD model, but has the advantages, that it requires less computing power when included in a mathematical model. The simplified volume, having a regular shape must always fully incorporate the precisely described (CAD) volume. Hence the name of such volumes is bounding volume. Examples for such bounding volumes with regular shapes are bounding boxes, bounding spheres, convex hulls. It is important to parameterize this volume model. Every linkage of the robot is described separately. The parametrization of this volume model is based on the kinematic constraints of the robot, and these constraints link together the bounding volumes, surrounding the linkages.

3. The current state of the robot is an input to the process. The state of the robot has to include all parameters which are relevant from the point of view of braking simulation. As mentioned before this aspect is not the topic of this research.

4. The braking path is generated based on the state of the robot. The braking path describes the coordinates of the TCP while the robot executes a braking maneuver from the state described earlier.

5. The braking volume is computed based on the braking path and the parameterized volume model. It represents the volume in Cartesian space occupied by the robot during the execution of the braking maneuver. It is computed as sum of all the instances in discreet time of the parameterized volume model while executing the braking path. This, in fact, is

the Minkowski sum of the parametrized volume model and the braking path. The obtained volume is the envelope in which the robot moves while executing the braking trajectory. The braking path is not likely to be executed, it represents the worst case scenario which is unlikely to happen. The braking volume quantifies this worst case scenario and it is the basis of the fault-tolerant distance criterion.

4.3.3 Fault-Tolerant Safety Criterion

The Fault-Tolerant Safety Criterion is defined as a binary condition. This eliminates the empirical factor of defining different safety thresholds like in the case of other published safety criteria. It expresses the overlapping of the volume occupied by the human inside the workspace of the robot, and the braking volume. As such, it can be formulated as a collision detection problem (also called intersection check). If the two volumes do not overlap, the criterion is satisfied.

$$V_{Braking} \bigcap V_{Human} = \emptyset \tag{4.3.1}$$

Where $V_{Braking}$ represents the braking volume, V_{Human} represents the volume occupied by the human, $\cap$ represents the intersection of the two volumes, and $\emptyset$ represent an empty set.

The violation of this condition, where the two volumes overlap, is presented in figure 4.15.

Collision detection problems are in many cases reformulated and solved as distance calculation problems. If the minimum distance between two volumes is greater than zero they cannot overlap. The minimum distance between the braking volume and the volume occupied by the human should be greater than zero, this way the condition 4.3.1 is satisfied.

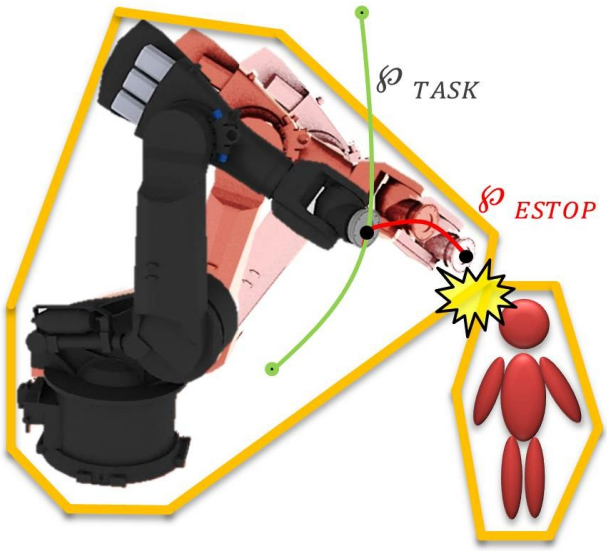


Figure 4.15: The violation of the Fault Tolerant Safety Criterion. $\wp_{task}$ represents the path executed by the robot, while $\wp_{brake}$ represents the path that would be executed if a fault would occur.

Converting the Fault-Tolerant Safety Criterion from a collision detection problem to a distance calculation problem leads to the definition of the Fault-Tolerant Distance.

$$d\left(V_{Braking}, V_{Human}\right) \geq 0 \tag{4.3.2}$$

Where d represents the Cartesian distance between the braking volume, $V_{Braking}$, and the volume occupied by the human, V_{Human}.

The method of calculation of the distance between two volumes is strongly influenced by how the volumes are described mathematically. The Fault-Tolerant Distance, as a concept, is not dependent on which volume description method is chosen. The important aspect in this matter is that this distance calculation, unlike the braking volume, is dependent on the state of the environment of the robot, not just the state of the robot. As such it has to be carried out in real-time. Different algorithms for most volume description methodologies are published, which can compute this distance in real-time, having as input the

two volumes. Furthermore, trading of the precision of the volume description speeds up calculations significantly, this way real-time performance is assured.

A fault in the hardware or software structure of the robot provokes the activation of the brakes. A Fault-Tolerant Distance greater than zero assures, that even if a fault occurs, the robot will not collide with the human. This assures the physical, bodily integrity of the human, psychological factors (e.g. fear, panic, stress) have not been considered.

4.4 Fault-Tolerant Distance in the APF Method

In order to plan paths based on the Fault-Tolerant Distance it has been integrated in a path planning algorithm.

Global planning algorithms can incorporate such a criterion easily, by incorporating equation 4.3.2 as a constraint in the optimization procedure. This shows the universal applicability of this newly defined criterion.

On the other hand, in order to include the Fault-Tolerant Distance criterion in a local path planning algorithm, its mathematical definition, as presented in 4.3.2, is not enough. This is due to the nature of the APF path planning algorithm, not due to the nature of the criterion. In order to include a constraint in the APF method, it has been formulated as a potential function.

Obstacles in the APF method have a potential function attached to them, which is defined in function of the distance between the obstacle and the TCP (or the obstacle and different characteristic points of the robot). This is presented in equation 3.1.6. A similar approach has been considered also in the case of human obstacles. In the case of ordinary obstacles the potential function is in function of the distance.

$$U_{obst} = f\left(d_{cart}\right) \tag{4.4.1}$$

Where U_{obst} represents the potential function attached to the obstacle and d_{cart} represents the Cartesian distance.

When considering a human obstacle, as presented above, the Cartesian distance does not guarantee safety in all scenarios. For the humans inside the workspace, instead of a repellent potential function based on Cartesian distance, a repellent potential function based on the Fault-Tolerant Distance is proposed.

$$U_{human} = f\left(d_{FT}\right) \tag{4.4.2}$$

Where U_{human} represents the potential function attached to the human-obstacle and d_{FT} represents the fault tolerant distance attached to the human-obstacle.

This way, formulated as a sigmoid function, the potential function of the human-obstacle becomes:

$$U_{human} = \frac{1}{1 + e^{-\alpha - \beta \cdot (d_{FT})}} \tag{4.4.3}$$

Where U_{human} represents the potential function attached to the human-obstacle, e represents Euler's constant α and β represent scaling constants and d_{FT} represents the Fault-Tolerant Distance.

The quantitative difference between the the potential function attached to the human-obstacle and an ordinary obstacle is obvious. It relies in the difference between the calculation of the Cartesian distance and the Fault-Tolerant Distance.

In the state of the art APF algorithm the obstacles do not repel only the TCP of the robot, but also the linkages. This effect has been achieved by defining characteristic points in the structure of the robot, and considering that these characteristic points, and this way the linkages they are defined on, are also repelled. This is presented in figure 4.16.

A similar approach has been considered to integrate the Fault-Tolerant Distance based linkage repel functionality. In the above presented form, the Fault-Tolerant Distance criterion characterizes the state of the robot.

A method has been developed, in order to reflect the linkage repel characteristic of the APF algorithm also in the case of the Fault-Tolerant Distance.

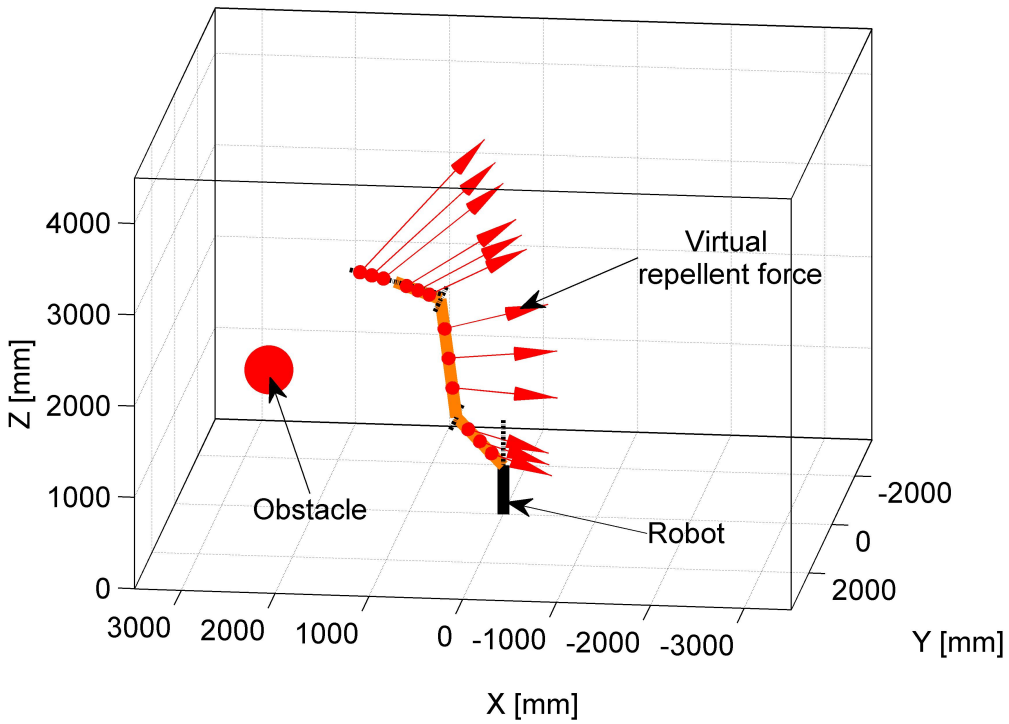
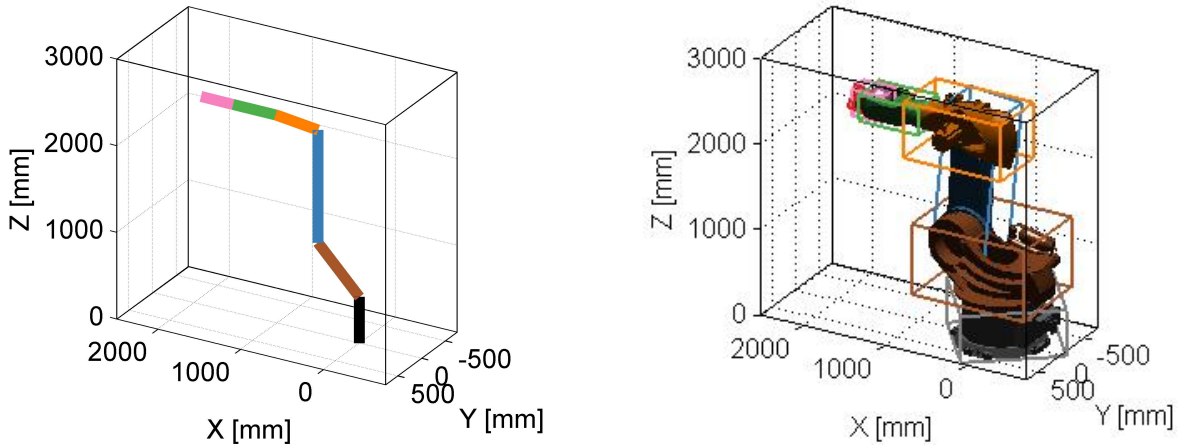


Figure 4.16: Obstacle repelling the linkages of the robot

Figure 4.17a shows the theoretical, wire-frame model of the robot. The differently colored segments represent the different linkages of the robot. Figre 4.17b shows the CAD model of the robot. The linkages are colored similarly as in figure 4.17a. Also the bounding boxes attached to each linkage are presented. Instead of using one large bounding box for the whole robot, each bounding box surrounds just one linkage (parametrized volume model). This leads to a more detailed volume description, but has a small trade off when considering the number of parameters required to describe the volume. It is important to observe the analogy between theoretical model and bounding box model. The integration of the Fault-Tolerant Distance in the APF algorithm has been carried out similarly to this analogy.

Figure 4.18a shows the same theoretical model, but the characteristic points on the linkages are also represented. The red dot marks the characteristic points. Each linkage has 3 such points. Please note that the number 3 is just for representation, more or less characteristic points can be used. In the APF



(a) Kinematic model

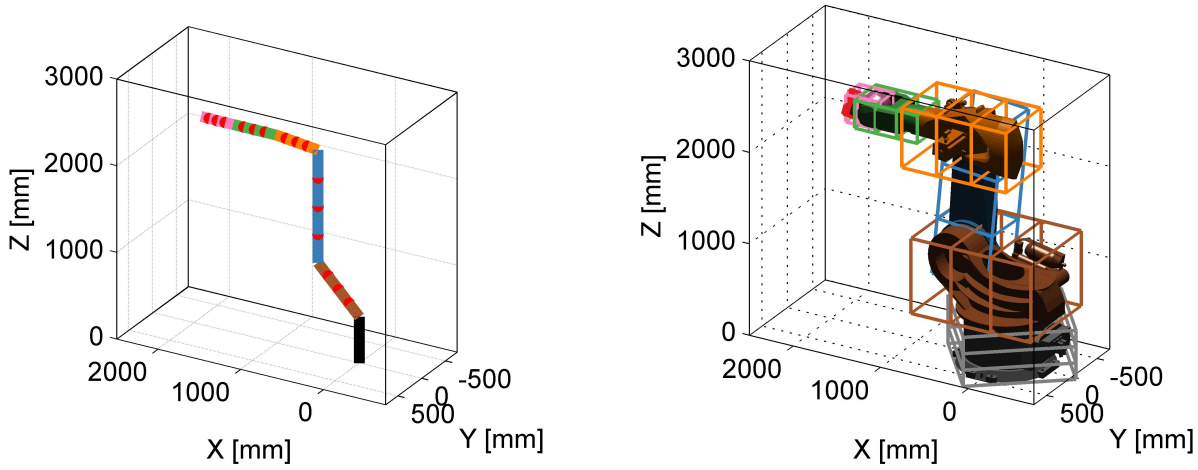
(b) Volume model

Figure 4.17: Analogy of paramterized kinematical model and paramterized volume model

method these points are repelled by obstacles, this way collisions between the obstacles and the linkages is not possible.

From the above presented analogy it has been concluded, that for each characteristic point a subvolume of the bounding box should be attributed. In this manner, the bounding boxes of the linkages have to be divided. Each bounding box from figure 4.17b has been divided into 3 subvolumes. Since the characteristic points are distanced equivalently, the subvolumes have been also divided equally. Figure 4.18b shows the division of the bounding boxes into subvolumes.

Dividing the linkage bounding boxes into subvolumes permits the division of the braking volume into subvolumes. By having braking subvolumes, the Fault Tolerant Distance can be calculated for every subvolume. This way a Fault Tolerant Distance parameter has been attached to each characteristic point, not just one for the entire robot. Although this is not needed for global planning algorithms it is essential for the APF method. This way the Fault Tolerant Distance has also been reflected in the linkage repel functionality of the APF method.



(a) Characteristic point for linkage repel

(b) Subvolumes for fault-tolerant linkage repel

Figure 4.18: Analogy of characteristic points and subvolumes

In order to illustrate the presented concept, a linearized braking model has been considered. A linear deceleration has been assumed as law of motion that for each axis, q_i.

$$\ddot{q}_i = \alpha_i \cdot t + \beta_i \tag{4.4.4}$$

Where α_i and β_i represent the parameters of the deceleration and t represents time.

The initial condition of the braking maneuver is the current pose of the TCP

$$Q_{\text{initial}} = [q_{1_{\text{initial}}} \quad \dots \quad q_{6_{\text{initial}}}]^T \tag{4.4.5}$$

and the current velocity

$$\dot{Q}_{\text{initial}} = [\dot{q}_{1_{\text{initial}}} \quad \dots \quad \dot{q}_{6_{\text{initial}}}]^T \tag{4.4.6}$$

The braking trajectory in this case has been computed by integrating the law of motion. The velocity component is given by

$$\dot{Q} = \int_{t_{\text{init}}}^{t_{\text{stop}}} \ddot{Q} \mathrm{d}t \tag{4.4.7}$$

The position (or in this case angular) component is given by

$$Q = \int_{t_{\text{init}}}^{t_{\text{stop}}} \dot{Q} \mathrm{d}t \tag{4.4.8}$$

Descretizing the curve obtained in equation 4.4.8, and applying the direct kinematic function to each point makes it possible to represent the curve in Cartesian space.

Given the initial conditions

$$Q = [-0.3646\ 0.4717\ 1.5555\ 1.2603\ 0.5761\ -2.5135]\,[rad] \tag{4.4.9}$$

and

$$\dot{Q} = [-0.1476\ 0.0034\ -0.2960\ -0.6693\ -0.1952\ 0.4075]\,[rad/s] \tag{4.4.10}$$

having as parameter

$$\mathrm{B} = [0.174\ -0.174\ 0.174\ -0.349\ 0.349\ -0.349]\,[rad/s^2] \tag{4.4.11}$$

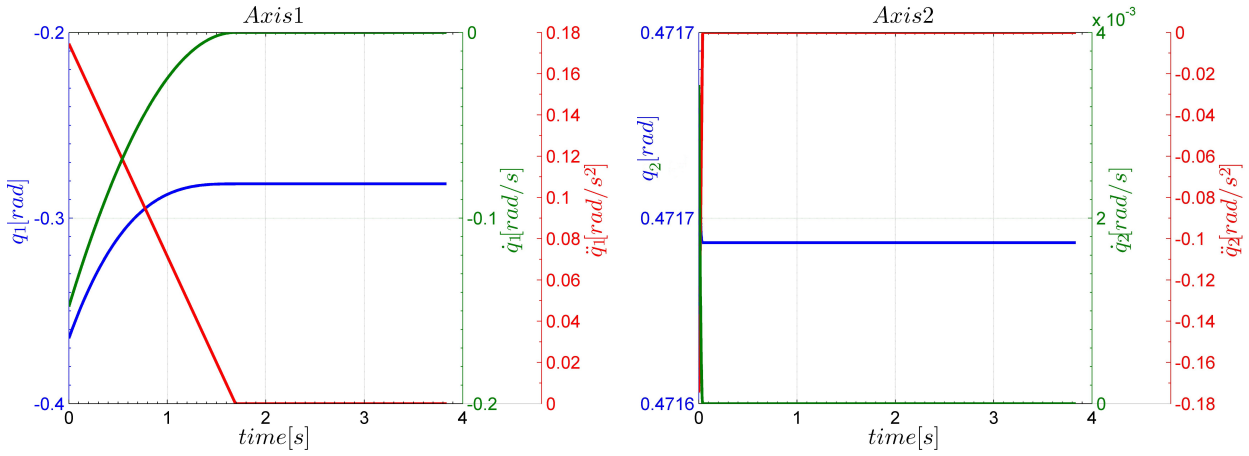
the obtained braking trajectories, using assumption 4.4.4 on each axes are presented in figure 4.19.

The equivalent of the braking trajectory in Cartesian space for this motion is presented in figure 4.20.

The subvolume attached to a linkage, and the braking volume corresponding to this subvolume is also presented in figure 4.21a. Figure 4.21b shows the braking volume for the whole robot structure. The whole robot structure remains inside this space (given the initial conditions) while executing the braking maneuver.

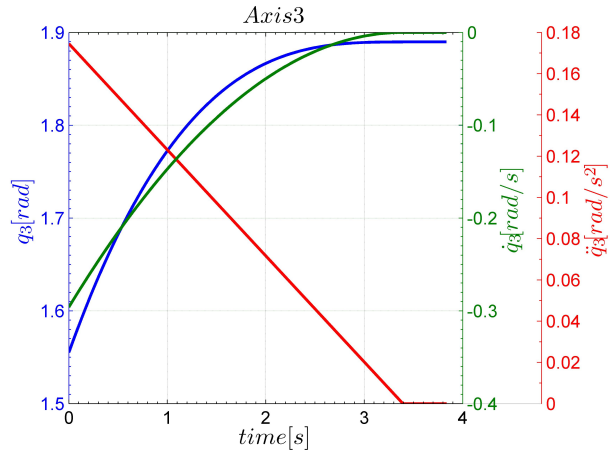
This way, taking into consideration the linkage repel functionality, equation 4.4.2 becomes:

$$U_{human} = f\left(d_{FT}\left(V_{\text{sub}}, V_{\text{human}}\right)\right) \tag{4.4.12}$$

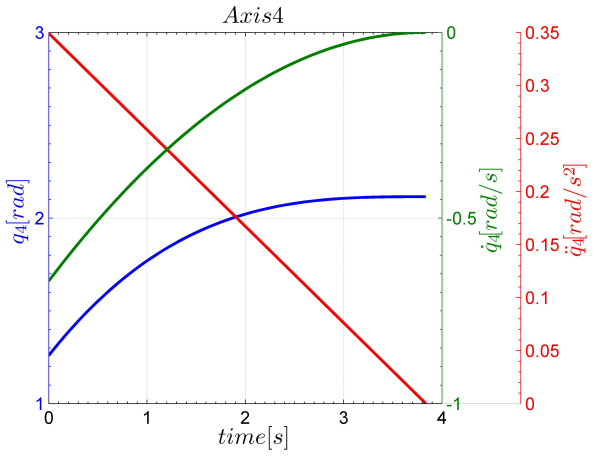


(a) Axis 1

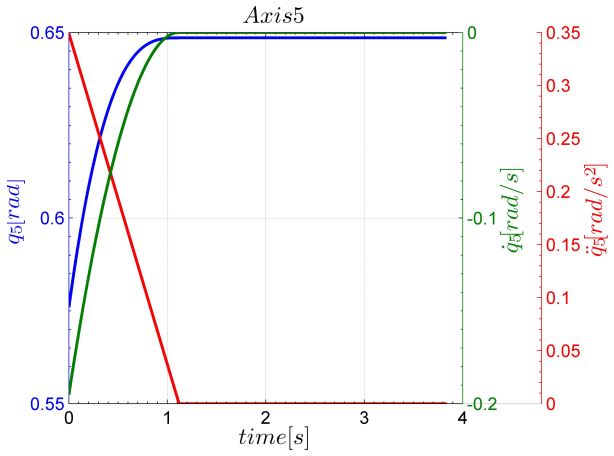
(b) Axis 2



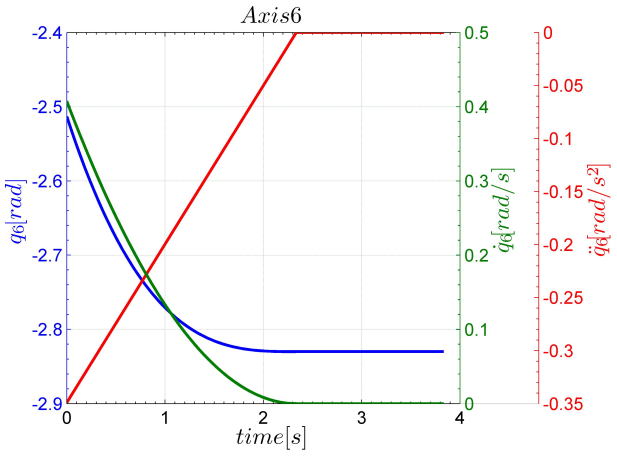
(c) Axis 3



(d) Axis 4



(e) Axis 5



(f) Axis 6

Figure 4.19: Braking trajectory in joint space

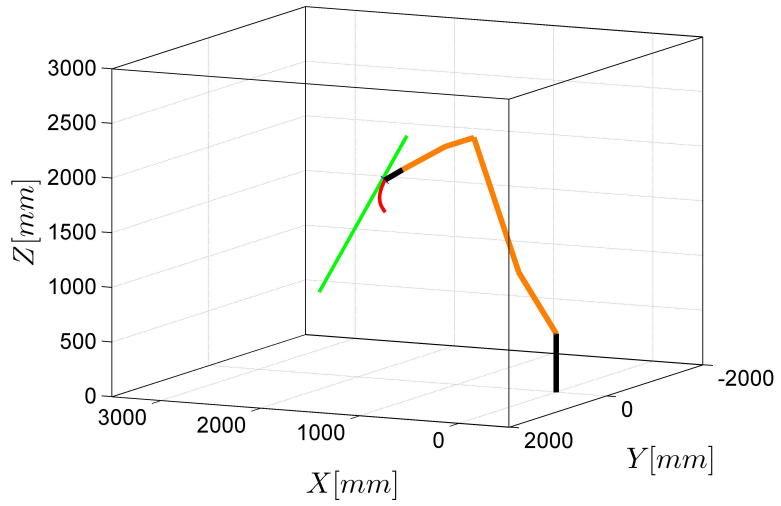
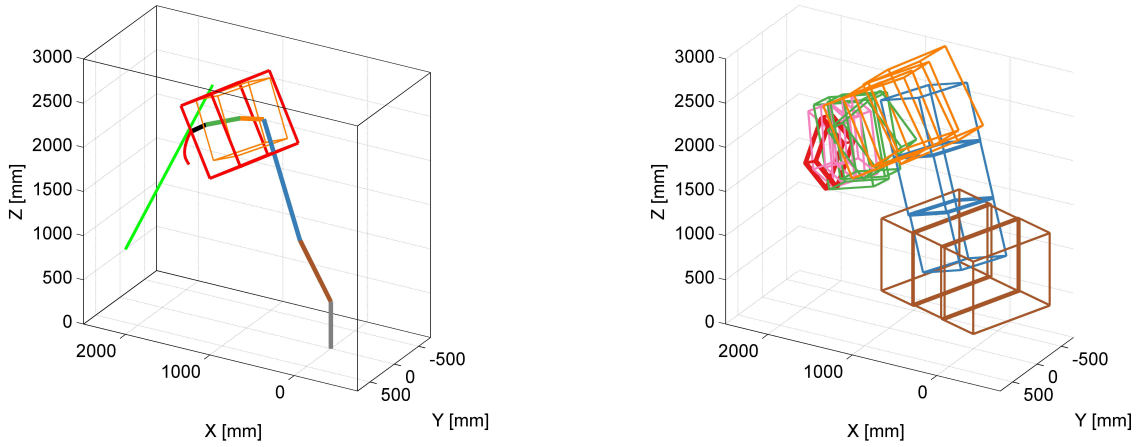


Figure 4.20: Braking trajectory in Cartesian space. The orange structure represents the kinematic structure of the robot, the green line represents the trajectory executed by the robot, the red line represents the trajectory that would be executed in case of braking



(a) One braking subvolume

(b) Entire braking volume

Figure 4.21: Braking Volume

Similarly, equation 4.4.3 becomes:

$$U_{human} = \frac{1}{1 + e^{-\alpha - \beta \cdot (d_{FT}(V_{\text{sub}}, V_{\text{human}}))}} \tag{4.4.13}$$

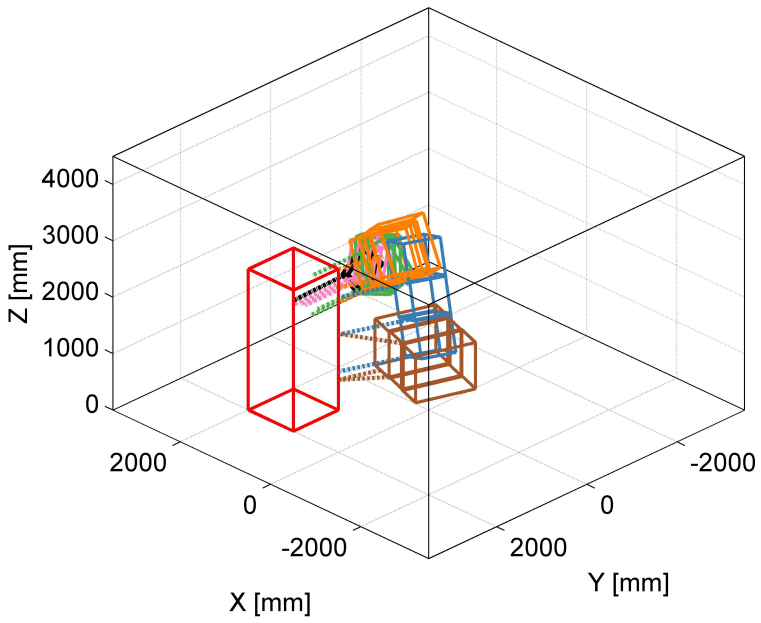


Figure 4.22: Linkage repel functionality using Fault Tolerant Distance

Where $d_{FT}\left(V_{\text{sub}}, V_{\text{human}}\right)$ represents the distance between one subvolume of the braking volume and the volume occupied by the human. This potential can calculated for every braking subvolume, attached to every characteristic point.

Based on this the linkage repel functionality is presented in figure 4.22. The dotted lines represent the distance from the volume occupied by the human (red bounding box) to the braking subvolumes. This way, for each characteristic point the correct potential field strength can be calculated.

In order to further facilitate the integration of the Fault-Tolerant Distance in the APF method, changes are proposed also to the modeling of the human body as a potential field.

4.4.1 Human Body Potential Model

The human body in the workspace of the robot is essentially an obstacle. The fact that the risk of collision between the robot and the human has to be reduced makes this obstacle special. The Fault-Tolerant distance deals with this risk from the robot side. Another risk can be mitigated by a modification to how the human body is modeled.

The detail of representation of the human body is of interest from the point of view of the human robot interaction. The precise modeling of hand and arms makes complex interaction scenarios possible. However not every dimension of the human body participates in the interaction.

Modeling the human body with an infinite height (or at list as having a height slightly larger then the workspace of the robot) makes the interaction safer. The infinite height prevents a scenario where the payload of the robot is above the human. This can be dangerous, and if not prevented, it can lead to injury.

A common scenario is a robot handling palettes. Palettes are usually worn out, maybe damaged. When lifted these palettes support a high payload and may break. Modeling a human as an obstacle with infinite length can prevent this situation, thus reducing risk of injury. It acts as a built-in safety criterion.

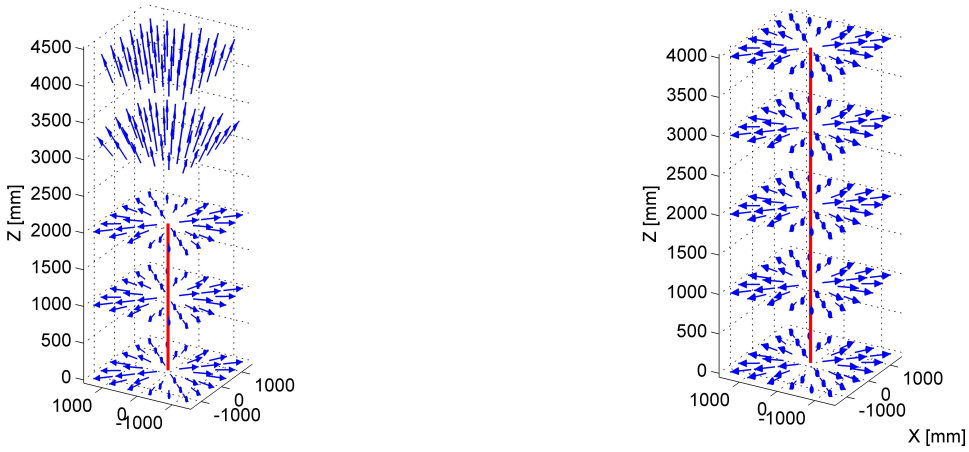
The human body, when using the APF method, has to virtually repel the robot. As such, it has to be modeled as a potential source. The potential function attached to the human body has to reflect its form and size. The detail it has to be modeled with is dependent on the application. For sophisticated collaborating operations the hands and palms have to be modeled independently.

The human body is modeled as a line type obstacle with infinite height. The line type obstacle defines a zone in which the robot cannot enter. The forbidden zone has the shape of a cylinder Csiszar et al. 2012. The radius of the cylinder is given by the parameters of the potential function. Based on the sigmoid potential function, presented in equation 4.2.10, a potential function can be defined:

$$U_{\text{human}} = U \frac{1}{1 + e^{-\alpha - \beta \cdot d(X, H_{\text{line}})}} \tag{4.4.14}$$

Where α and β represent scaling constants, X represents the coordinates where the potential function is evaluated, $d\left(X, H_{\text{line}}\right)$ represents the Cartesian distance from the point X to the line representing the human obstacle.

Figure 4.23 presents equation (4.4.14) visually.



(a) Potential Field with finite length

(b) Potential field with infinite length

Figure 4.23: Potential field created by the human body model

The aim of this approach is to complete the Fault-Tolerant Criterion, not to substitute it. In order to integrate the Fault-tolerant distance with this safety functionality the Z dimension of the volume occupied by the human has to be considered infinitely long.

$$U_{\text{human}} = U \frac{1}{1 + e^{-\alpha - \beta \cdot d\left(V_{Braking}, V_{Human_{ZInf}}\right)}} \tag{4.4.15}$$

Where α and β represent scaling constants, e represents Euler's constant, $V_{Braking}$ represents the braking volume, $V_{Human_{ZInf}}$ represents the volume occupied by the human, with the Z dimension considered infinitely long and d_{FT} represents the distance between $V_{Braking}$ and $V_{Human_{ZInf}}$.

Equation 4.4.15 incorporates the two safety measures. The dependency from the Fault-Tolerant Distance reflects the safety guarantee in case of a malfunction. The infinite length of the human obstacle guarantees, that the payload of the robot cannot pass above the head of the human.

4.5 Planning Algorithm

4.5.1 Virtual Force to Reference Value

Many industrial robot controllers accept position offset as control references. In order to obtain the reference position offset the virtual force has to be "converted" to position. This "conversion" is needed, since the dynamics of the robot are not modeled, and applying the virtual force to the real robot would result in unpredictable effect. The potential field can be correlated with the dynamic model of the robot, and so the virtual force can be used as reference for the TCP force. This requires the precise dynamical model of the robot, which is not openly available for commercial robots. Depending on the application, it can be considered, that the total virtual force, $\vec{F}_{\text{Total}}$, is acting on a massless solid in viscous environment Csiszar et al. 2012. In this case:

$$\vec{v}_s = \vec{F}_{\text{Total}} \cdot \frac{1}{b} \tag{4.5.1}$$

$$\Delta\vec{p} = \vec{v}_s \cdot \Delta t \tag{4.5.2}$$

Where b is the viscosity of the virtual environment, $\vec{v}_s$ is the velocity of the massless solid, Δt represents the time change and $\vec{p}$ is the position of the massless solid.

By giving $\Delta\vec{p}$ as relative position reference to the robot, it will copy the motions of the considered virtual massless solid.

In order to reflect the limited bandwidths of the actuators of the robot and to create motion profiles with reduced jerk, it can be considered, that the total virtual force is acting upon a solid with non-zero mass. In this case, the velocity of the solid $\vec{v}_s$ will be an additional state, in order to compute the position offset, $\Delta\vec{p}$.

$$\vec{v}_{s_i} = \vec{v}_{s_{i-1}} + \vec{F}_{total} \cdot \frac{1}{m} \cdot \Delta t \tag{4.5.3}$$

$$\Delta\vec{p} = \vec{v}_{s_i} \cdot \Delta t \tag{4.5.4}$$

Where m is the mass of the solid.

4.5.2 Path Planning Algorithm

Path planning in joint space has the advantage, that the planned path, in the absence of obstacles, lies inside the workspace limits. However, obstacles can "push" the path to cross the workspace limits. Also mapping the obstacles to joint space is a computationally intensive task.

Path planning in Cartesian space has the advantage of requiring few calculations. It has the disadvantage, that in many cases, it produces paths that would cross workspace limits.

In this thesis a modification to the APF method is proposed that combines the advantages of the two path planning strategies (the joint space planning and the Cartesian space planning), and minimize their disadvantages. For a global planning approach, the definition of some aspects in joint space and some aspects in Cartesian space may not be efficient, the reduced number of calculations, implied by local path planning, make such an approach viable. This combination is efficient because of the characteristics of the local path planning. The potential field approach to path planning permits the combined use of the joint space and Cartesian space for path planning. The potential field attached to the target point is evaluated in joint space, while the potential field attached to the obstacles is evaluated in Cartesian space. This preserves the advantage of joint space planning, the planned paths in the absence of obstacles will not cross the workspace limits. When obstacles are present, the workspace limit potentials presented in section 4.2 will prevent planning outside the workspace. It also preserves the advantage of Cartesian space planning. The obstacles are not mapped into joint space, this way the algorithm is not computationally demanding.

This combined planning is possible because of the characteristics of potential field methods. In this method information about the environment is condensed

down to a vector value (a virtual force), and the path planning is based only on this force. This way, by defining the potential functions in different spaces only a vector value has to be converted between different planning spaces, not the whole environment.

The reason that permits such an approach in the case of the APF method is the difference between defining a point in joint space, evaluating its potential function in joint space and converting the obtained potential in operation space, as described in equation (4.5.5), and between defining a point in joint space converting the point to Cartesian space and evaluating its potential function in Cartesian space, as described in equation (4.5.6).

$$^{\mathcal{W}}\nabla U_1 = {}^{\mathcal{W}}\nabla U_1 \left(f \left(Q \right) \right) \tag{4.5.5}$$

$$^{\mathcal{W}}\nabla U_2 = J \times {}^{\mathcal{C}}\nabla U_1 \left(Q \right) \tag{4.5.6}$$

$$^{\mathcal{W}}\nabla U_1 \neq {}^{\mathcal{W}}\nabla U_2 \tag{4.5.7}$$

Where the superscript $^{\mathcal{W}}$ refers to a parameter in Cartesian space, the superscript $\mathcal{C}$ refers to a parameter in joint space, J represent the Jacobi matrix if the robot and f represents the direct kinematic function.

The path planning algorithm is described below, and also represented visually in figure 4.24.

1. Initializations
   - In this phase, all operations are executed which need to be executed only once, outside the real-time loop (e.g. initialization of the communication with the robot controller, initialization of the communication with the sensor system, etc.).

2. Read data
   - Read data from robot (e.g. current TCP coordinates)
   - Read data from sensors (e.g. current obstacle coordinates)

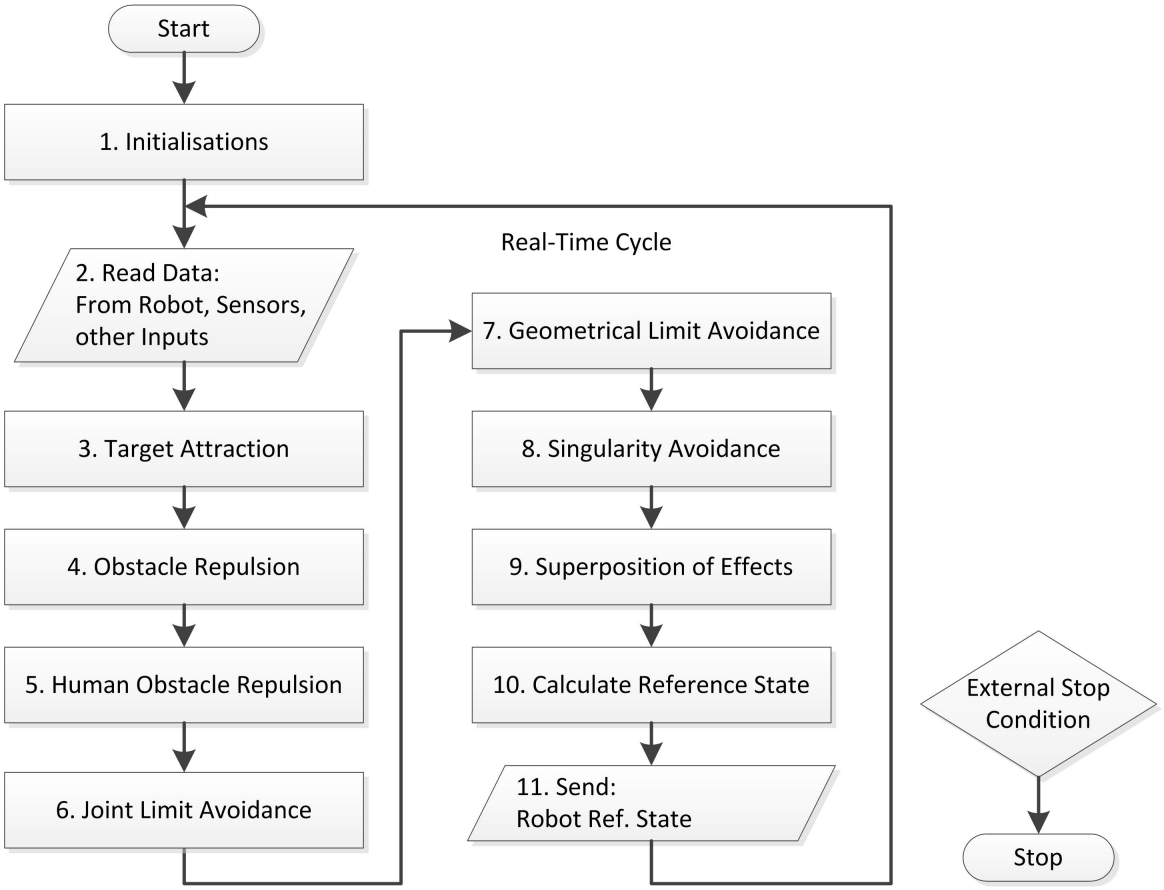


Figure 4.24: Path planning algorithm flowchart

3. Target Attraction
   - Convert the target position to joint space using equation (3.1.19).
   - Evaluate the target potential function, based on equation (3.1.20).
   - Calculate the target virtual force in joint space based in equation (3.1.3).
   - Convert the target virtual force from joint space to Cartesian space, using the Jacobi matrix of the robot.

$$^{\mathcal{W}}F_{\text{Target}} = J \times {}^{\mathcal{C}}F_{\text{Target}} \tag{4.5.8}$$

4. Obstacle Repulsion
   - Evaluate the potential functions attached to obstacles in Cartesian space. Polynomial formulation of the potential function is shown in equation (3.1.6), sigmoid function formulation is shown in Ren et al. 2007. The potential must be evaluated at the TCP, and at

different characteristic points of the robot, as presented in figure 3.4 and described in Khatib 1986. This way, collisions will be avoided both by the TCP and the linkages.

- Calculate the virtual force repelling the the robot from the obstacles based on equation (3.1.3). When using sigmoid functions a similar approach as presented in equation (4.2.20) can be used.

5. Human Obstacle Repulsion
   - The humans in the workspace of the robot are handled differently then ordinary obstacles in order to guarantee their safety. The potential function attached to the humans is evaluated, as presented in equation (4.4.3). To include also linkage repulsion the potential function has to be evaluated for the subvolumes attributed to the characteristic points on the robot. Similarly as in the case of ordinary obstacles, at every characteristic point the potential function is evaluated, as shown in equation (4.4.13)
   - Based on the value of the potential function, the virtual force repelling the robot from the humans is calculated, as described in equation (4.5.8)

6. Joint Limit avoidance
   - Evaluate the joint limit potential function in joint space, using equation (4.2.7).
   - Calculate virtual force repelling from joint limits in joint space, using equation (4.2.21).
   - Convert virtual force to Cartesian space, using equation (4.2.22).

7. Geometrical Limit Avoidance
   - Evaluate geometrical limit potential function in operational space, using equation (4.2.23).
   - Calculate virtual force repelling from geometrical workspace limit, using equation (4.2.26).

8. Singularity Avoidance
   - Evaluate potentials attached to singular poses in joint space, using equation (4.2.28).
   - Calculate virtual force repelling from singular regions in joint space, using equation (4.2.30).
   - Convert virtual force to Cartesian space, using equation (4.2.32).
9. Superposition of Effects
   - Summarize all calculated virtual forces. A weighing factor for the different forces can also be considered, using the following equation:

$$\begin{aligned}\vec{F}_{\text{Total}} = \; & \mathsf{w}_{\text{Target}} \cdot \vec{F}_{\text{Target}} + \mathsf{w}_{\text{Obst}} \cdot \vec{F}_{\text{Obst}} + \mathsf{w}_{\text{Joint}} \cdot \vec{F}_{\text{Joint}} + \\ & + \; \mathsf{w}_{\text{Geom}} \cdot \vec{F}_{\text{Geom}} + \mathsf{w}_{\text{Sing}} \cdot \vec{F}_{\text{Sing}}\end{aligned} \tag{4.5.9}$$

   Where w represents a scaling factor for every term of the sum.
10. Calculate Reference State
    - Calculate the reference coordinates, which will be sent to the robot controller, as reference values, based on equation (4.5.2).
11. Send Data
    - Send reference values to the robot.
12. The real-time cycle restarts at step 2. Since it is a real-time control algorithm is is ran in an infinite loop, with an external stop condition.

5 Numerical and Experimental Results

5.1 Workspace determination

5.1.1 Experimental Setup

Experimental validation of the computation of the Partner Robot's workspace is a challenging task, because it would require an external sensor system, A simple and low cost method has been developed to compare the theoretical and practical aspects of this problem presented in figure 5.1. A large sheet of paper (a) is placed on a hard surface (b). The TCP in this case is in fact a marker (c), capable of leaving a continuous trail on the paper, having passive compliance along the z-axis. The sheet with the hard surface below is fixed at different heights below the TCP. The pose of the TCP is controlled using dSpace development board (d), along the contour of the generated theoretical workspace, for that given height. The tip of the marker leaves a continuous trail on the paper. This trail in this specific case is in fact the contour of the workspace. The sheet of paper is then digitized, and compared to the theoretical contour, generated numerically. The described method is used to validate the mathematical model for the 6 DoF configuration of the PARTNER parallel robot.

The experimental setup presented in 5.1 consists of:

- Mechanical structure (e) of the robot including actuators (f)
- PC for algorithm development (Matlab) and Human Machine Interface (ControlDesk) (g)
- dSpace DS1103 Real-time Development Board (d)
- power supply (h)
- CAN communication network (i)

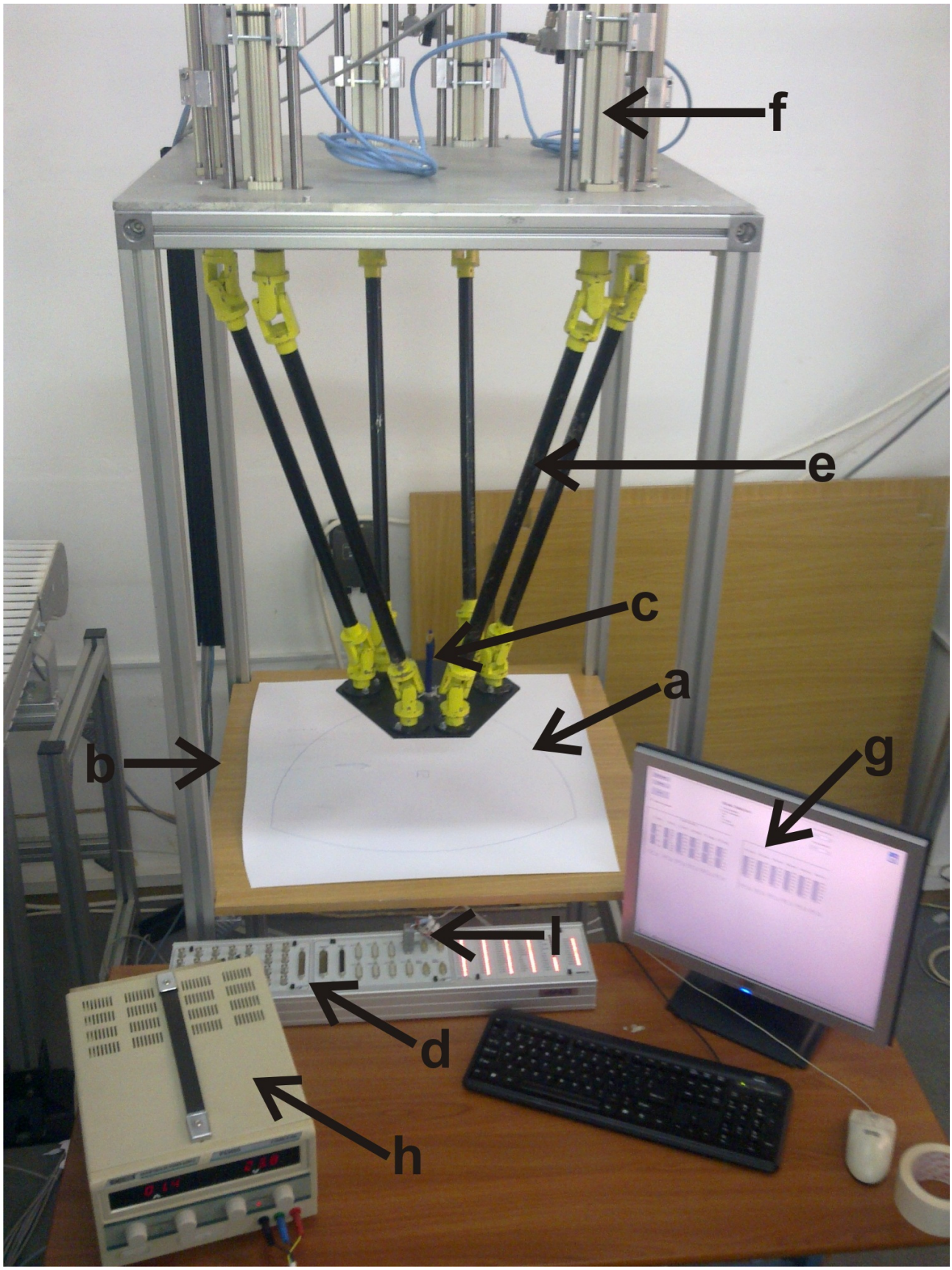


Figure 5.1: The experimental setup for workspace contour determination

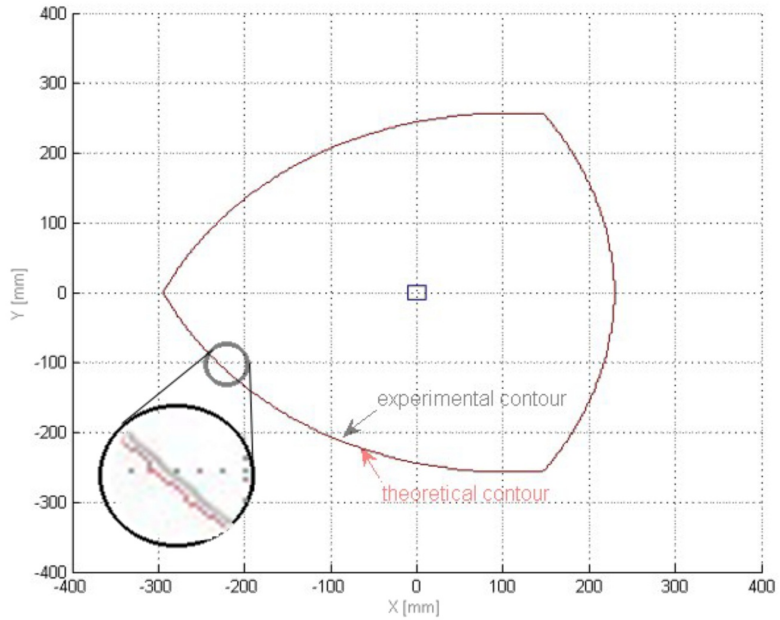
5.1.2 Experimental Results

The workspace of the 6 degrees of freedom configuration of the robot is generated as described above. An algorithm selects the contour of the workspace for the given height. The points from the contour of the workspace are given as target points for the robots control algorithm. This way the TCP will describe a (closed) curve. Since the TCP in this case is a felt tip marker, in contact with a sheet of paper, the contour can be visualized. Also a rectangle is plotted in the middle of the drawing. The scaling of the image after digitization is done using this rectangle, with known size. Figure 5.2a shows the contour of the workspace drawn by the robot the Z coordinate of $800mm$. The numerically generated contour of a workspace section parallel to the OXY plane, at the given Z coordinate is almost identical to the contour obtained experimentally. Small differences are caused by mechanical tolerances at manufacturing and assembly of the robot and/or, digitization errors. The experiment has been repeated for the Z coordinates $850mm$ and $900mm$ and very similar results have been obtained. Figure 5.2a proves that the contour generated numerically lies inside the workspace of the robot. Figure 5.2b presents the coordinates of each motor in function of time while plotting the contour. One can observe that at any given time at least one motor is at the maximum stroke (approximately). This proves that the plotted contour is the limit of the workspace. Note that the oscillations on the graphs are caused by the point to point motion of the TCP (no synchronous motion control algorithm has been used).

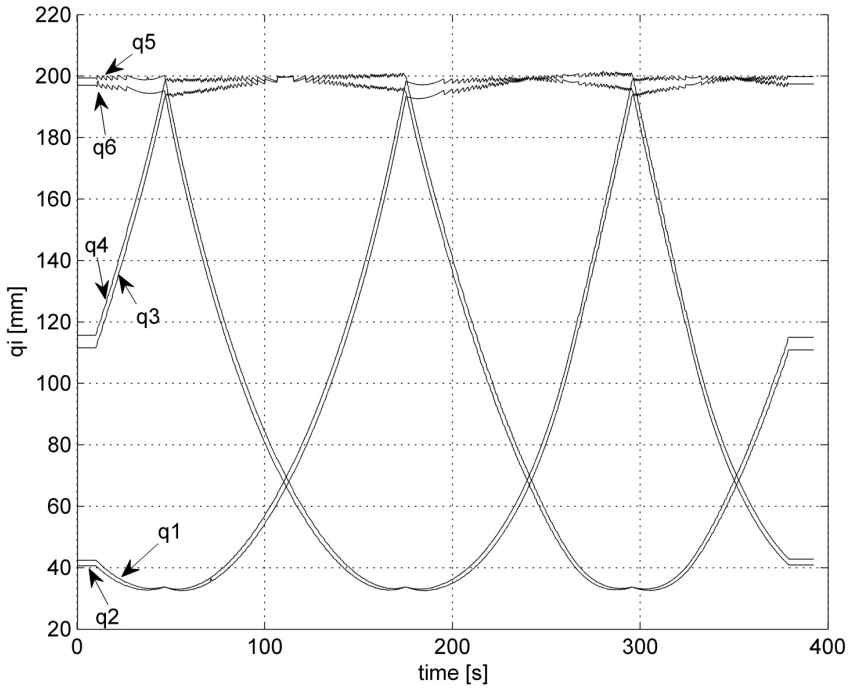
5.2 Path Planning

5.2.1 Numerical Results

The presented theoretical considerations have been verified and validated both in simulation and in practice. Simulation can offer insights to path planning algorithms, which experimental validation cannot, because in simulation phys-



(a) Workspace contour at $Z = 800mm$



(b) Motor strokes while drawing workspace contour at $Z = 800mm$

Figure 5.2: Partner Robot workspace section

ically impossible scenarios can easily be tested (e.g. TCP position outside the workspace).

The aim of the presented simulation results is to demonstrate the improvements on the path planning of the contributions of this thesis.

Figure 5.3 shows the utility of the joint limiting potential. The difference between the paths planned with and without this potential function can be observed. The path planned using the joint limiting potential function stays inside the workspace, while the path planned without this potential function violates the joint limits of the robot.

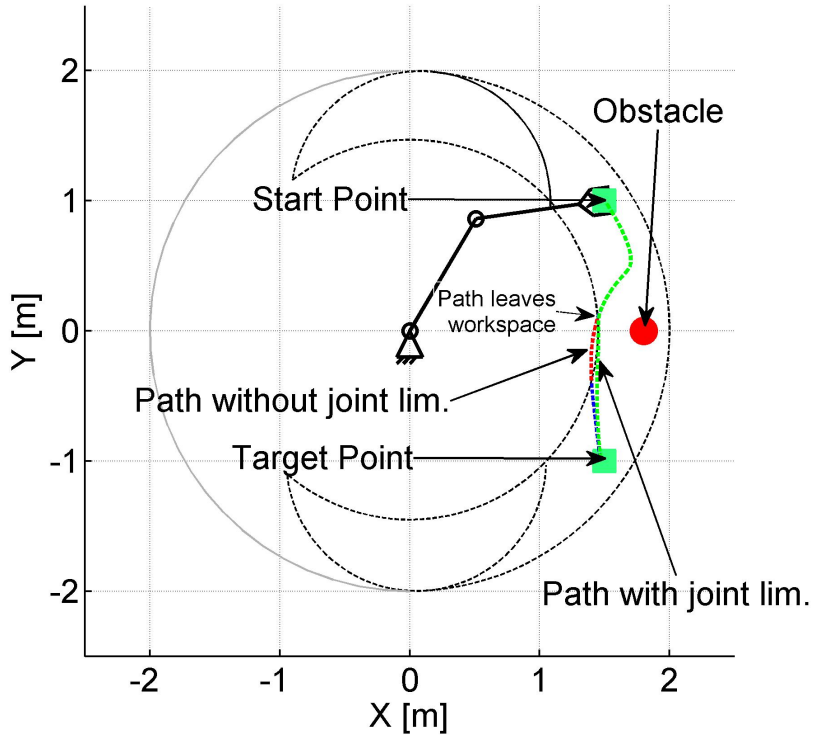


Figure 5.3: Path planning with and without joint limiting potential

Since this generated in simulation, the simulated TCP can "leave" the workspace. This is only possible mathematically, by violating the mathematical joint limit conditions. On a real robot this is not possible, an attempt to leave to workspace would lead to either damage to the joints, a stop command from the RC (robot controller), which may require further human assistance to continue work, or other undesired behavior, depending on the type of the RC and how it is con-

figured. The role of the joint limiting potential is to prevent an attempt for the TCP to leave the workspace. The introduction of the joint limiting potential can lead to further local minima points. In this case it is preferred that robot stops in a local minimum point, which can be handled internally in the path planning algorithm. Otherwise the RC or the joint limits will stop the robot, and the path planning algorithm cannot handle the error, without external human assistance.

Figure 5.4 shows the effects of the introduction of the geometrical workspace limit potential. The difference between the paths planned is convincing. The paths planned using the geometrical workspace limiting potential does not violate the workspace limits. The path planned without the geometrical workspace limiting potential leaves the workspace. Also in this case, the TCP can only leave the workspace in simulation, the mathematical relations which represent the mechanical integrity of the robot are violated (equation (3.1.18) has no real solutions), and the scenario where the TCP leaves the workspace has no equivalent in reality.

This is the reason why the path does not converge to the target point. The inverse kinematic function, f^{-1} has only complex solutions outside this limit. This way equation (3.1.18) has no real solutions. The role of the geometrical workspace limiting potential is to prevent an attempt for the TCP to leave the workspace.

The value of the potential field created by the target point, defined in joint space, as presented in (3.1.20) cannot be calculated in $\mathbb{R}$, and so, based on (3.1.3) the virtual force attracting the TCP to the target point has a complex value. This way the target point fails to attract the TCP in the case when the TCP leaves the geometrical workspace.

The aspects of local minimum points are mentioned above are valid also in this case. The presented scenario is somewhat unrealistic for the 2 DoF serial robot, since the linkages of the robot would collide with the obstacle while executing the path. When considering robots with movements in at least 3 dimensions (e.g. SCARA type) this scenario becomes valid, but hard to represent visually in an intuitive way.

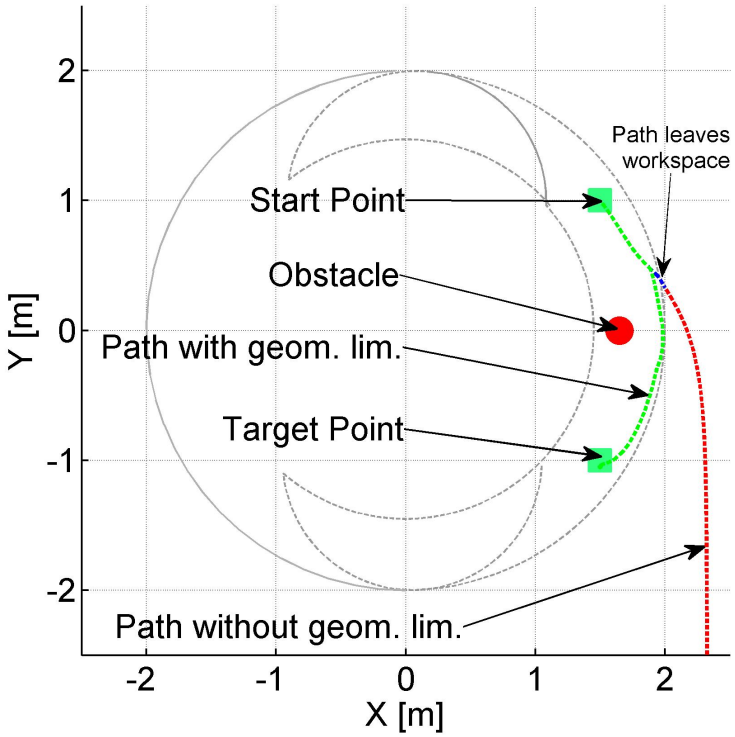


Figure 5.4: Path planning with and without geometrical workspace limiting potential

Figure 5.5 shows the effects of the singularity limiting potential. Since for the 2 DoF serial manipulator the joint limits considered in table 3.1 delimit a workspace without singularities, these limits are disregarded in this case. For other robot structures singularities can be found inside their workspace. The two planned paths show that singularity limiting potential has the effect of avoiding singular poses.

5.2.2 Experimental Setup

The experimental validation of the above presented theoretical considerations was done using industrial equipment. A Kuka KR-500 robot, with 6 degrees of freedom, with a KC-4 Robot controller has been used. The path planning algorithm has been implemented in Matlab Simulink. C code has been generated by Real-Time Workshop. The generated code of the path planning algorithm has been run in RTX, a real time extension of the Windows operating system.

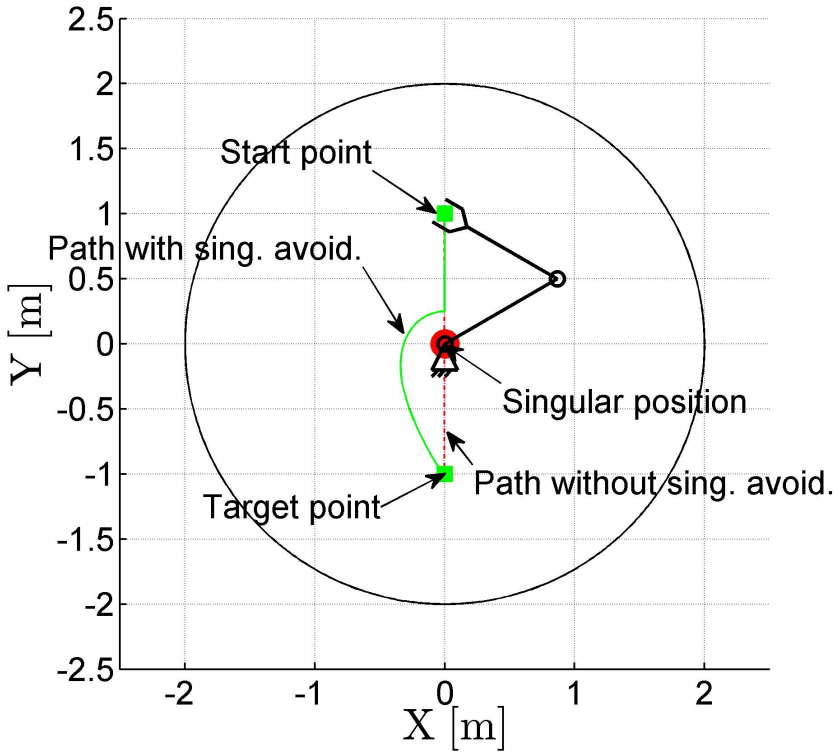


Figure 5.5: Path planning with and without singularity avoidance potential

The Kuka RSI, an ethernet based (quasi-) real-time protocol, has been used for bidirectional communication between the robot controller and the path planner.

In order to correctly apply the path planning, information about the size and the position of the obstacles in the workspace have to be available in real time. Since the aim of the experiment was to validate the functionality of the path planning algorithm, not industrial deployment, some simplifications have been made regarding the sensor system. The obstacles in the workspace have been detected by a Sick laser scanner, placed near the base of the robot with known coordinates. The scanner can only measure in two dimensions (in this case X and Y coordinates where measured), the third dimension of the obstacle has been considered to be infinitely long. This has an effect also on the path planner, no paths can be planned, that cross above the obstacle. Another simplification is also due to the used sensor. It has been considered that all obstacles are sensed by the scanner. The possibility of shadowing of the obstacles, a known problem for laser scanners, has been disregarded.

An industrial joystick, connected to the path planner algorithm using an EtherCAT IO connector, has been used to move the target point freely, or to move a virtual obstacle freely, to test the algorithm easily for various scenarios. The overall structure of the application is shown in figure 5.6.

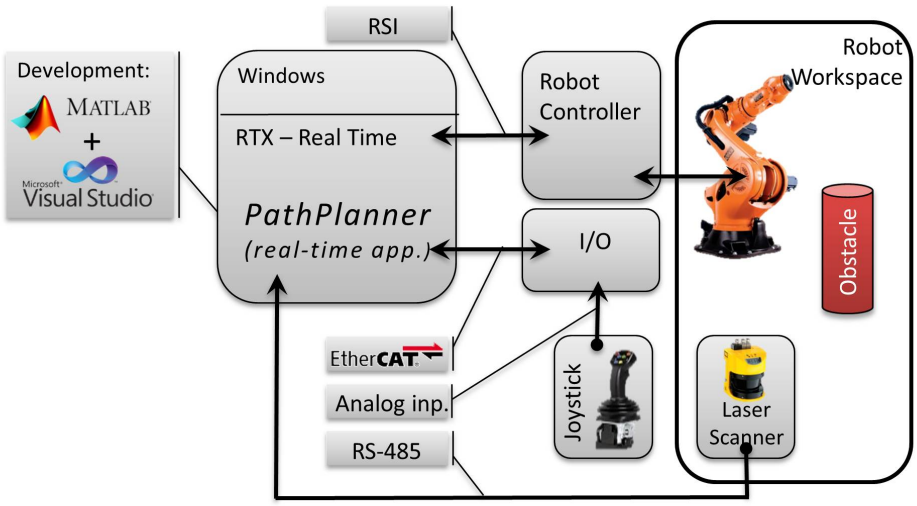


Figure 5.6: Overall structure of the experimental setup

5.2.3 Experimental Results

Figure 5.7 shows the path executed by the Kuka robot, planned by the artificial potential field planner. The point coordinates where logged by the Kuka RC, and imported to Matlab for visualization, and representation of the obstacles. The figure shows that the Kuka robot successfully avoids obstacles in the workspace, using the potential field planner.

The experiment was repeated successfully under similar conditions with a slightly different setup. Figure 5.8 shows a Kuka KR-360 Robot avoiding collision with another Kuka KR-360 Robot, considered a mobile obstacle.

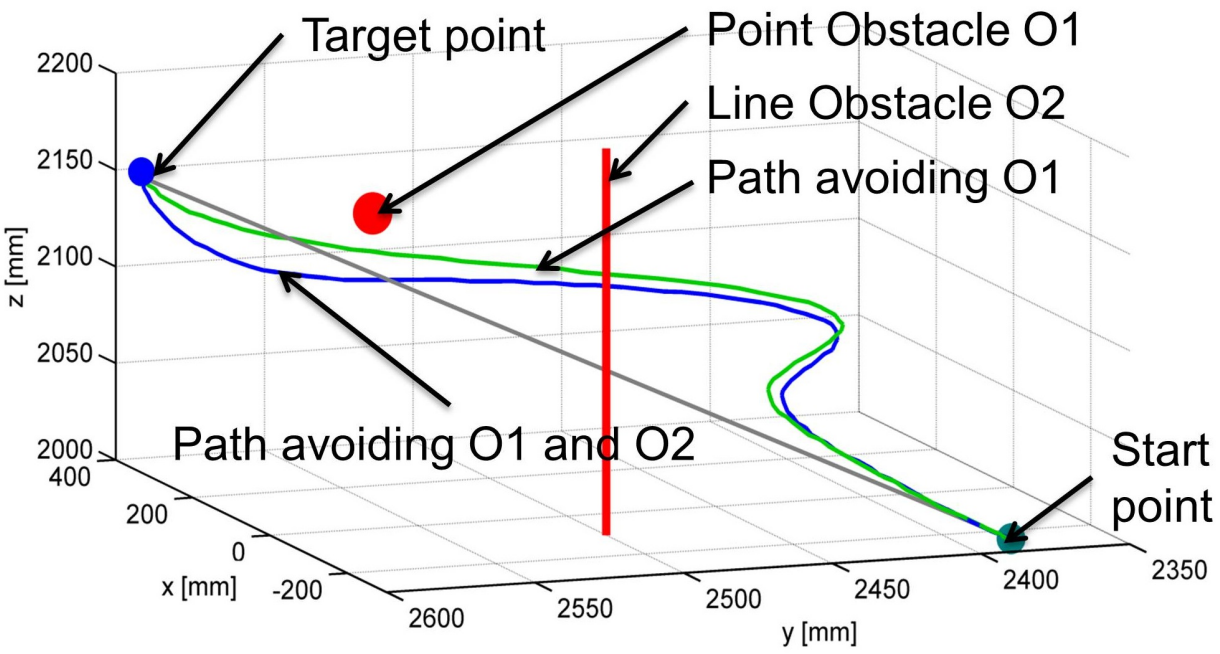


Figure 5.7: Path executed by the Kuka robot, planned by the potential field planner. Data is logged by the robot controller

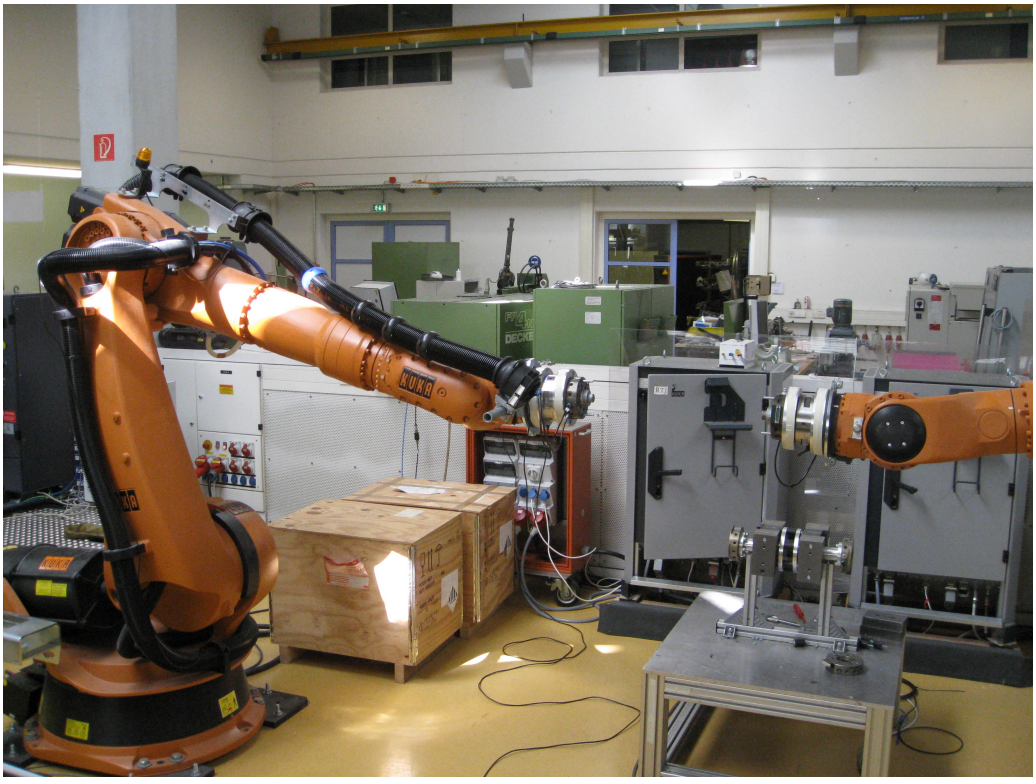


Figure 5.8: Kuka KR-360 Robot avoiding collision with another KUKA KR-360

5.3 Safe Path Planning

5.3.1 Numerical Results

The path planning algorithm was validated experimentally. Since the safety criterion is based on the assumption that the braking characteristics of the robot are known, it was validated in simulation.

A point to point motion of the robot and a human obstacle was considered for the validation scenario. The path planning algorithm ran two times, with and without the fault tolerant safety criterion. The results are shown in figure 5.9. The red trajectory represents the path planned without using the safety criterion. This trajectory violates the Fault-Tolerant Safety criterion. It a fault occurs during the execution of the portions of the path which are close to the human obstacle, the robot executing a braking maneuver would collide with the human obstacle.

The blue trajectory was planned by the APF method with integrated Fault-Tolerant Safety criterion. No other parameter of the path planning was changed. The qualitative difference can be seen on the figure. The generated path respects the Fault-Tolerant Safety criterion. This way, if a fault should occur at any time during the execution of the path, the robot executing a braking maneuver would not collide with the human obstacle.

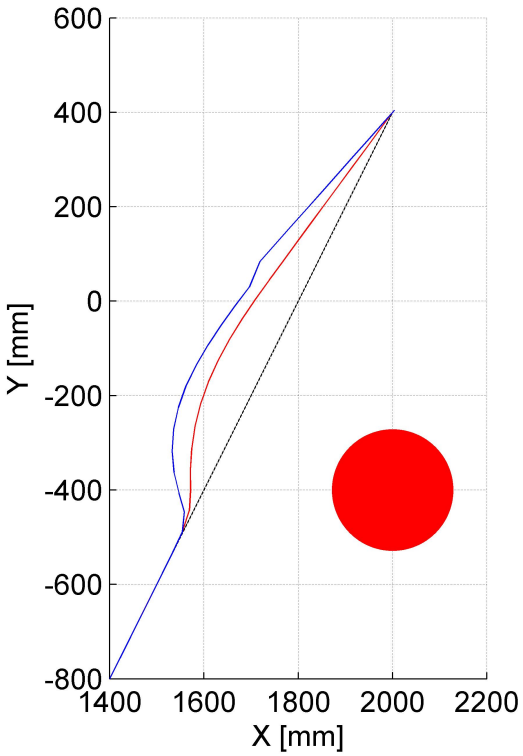


Figure 5.9: Path planning with APF method with (blue trajectory) and without (red trajectory) the fault-tolerant safety criterion

6 Conclusions and Future Outlook

The benefits of physical Human Robot Interactions come from the its possible applications. Such a technology can increase the productivity of factory worker by aiding them in their job, it can increase quality of life by aiding elderly or persons with disabilities in their everyday life, etc. There is one important condition that will enable the usage of such a technology. The safety of the persons in the vicinity of the robot has to be assured.

First of all the persons that can potentially be harmed by the robot have to be identified. It is physically impossible for a robot to collide with someone who is out of its reach, who is not inside its workspace. This is why the analysis of the workspace is important. In this thesis a modular approach to the computation of the workspace was presented. It is well suited for modular and reconfigurable structures, but not only. The workspace limits of the robots have been identified together with the structural elements responsible for these limits.

The workspace limits are important from the path planning point of view. The path planner should not plan paths that cannot be executed by the robot. In this sense all workspace limits were integrated in the APF method.

For pHRI the safety of the human participant in the interaction has to be guaranteed. For guaranteeing safety in every case, usually the worst case has to be identified and based on the worst case a safety assurance strategy has to be developed. The worst case in robotic path planning has been identified as the case where a software or hardware fault causes the robot to malfunction. This leads to an emergency braking maneuver of the robot. Based on this situation a safety criterion was developed. This safety criterion was integrated in the APF path planing algorithm, this way the APF algorithm plans paths that respect this criterion. This way, it is assured that the robot will never be in a state,

from which if an emergency braking maneuver is carried out, it would collide with the human obstacle.

The goal of the thesis was achieved, an online path planing algorithm with integrated safety criterion was developed, considering the predefined simplifications. The online path planning algorithm is an extension of an already published method. The safety criterion is based on research concerning the braking characteristic of the manipulators. It is a novel approach, that can assure safety also in a case of a malfunction of the robot.

The goal was achieved by the contributions to the state of the art that resulted from this research. These contributions are:

- development of a modular approach to workspace computation
- linking workspace limits to motion limits of the joints in the structure of a robot
- integrating the geometrical workspace limit in the APF method (formerly only joint limits were integrated)
- reformulating the potential function of workspace limits as sigmoid functions
- developing a novel safety criterion based on worst-case scenario
- integrating the safety criterion in the APF method
- integrating a safety functionality in the potential field of the human obstacle
- validating the contributions in simulation
- defining an experimental setups to validate most of the contributions experimentally
- validating most of the contributions experimentally

It is was not the goal of the thesis to develop industrially deployable solution to pHRI, but to offer new insights to the research regarding pHRI. The presented research can be considered a starting point for research regarding industrially deployable pHRI solutions. Such an application is presented in figure 6.1.

Figure 6.1: Illustration of industrial application of pHRI. The robot senses and reacts to the movements of the worker. © Fraunhofer IPA 2008

Prior to industrial deployment the braking trajectory of the robot has to be generated precisely. This either has to be done in real-time, either it has to be generated offline and saved as a lookup table or database. The fact that is only dependent on the structure of the robot, not on the obstacles, makes offline generation possible, however the load of the robot has to be considered. For offline generation the maximum allowable payload can be considered.

The described path planning solution can only handle point to point motions. For more complex motions, where the trajectory itself, or different aspects of the trajectory is of interest a different path planning methodology may be better suited, or the APF method has to be further developed to allow such applications.

Bibliography

Ahmed et al. 2010 — Ahmed, Muhammad Rehan and Kalaykov, Ivan. 2010. Static and dynamic collision safety for human robot interaction using magneto-rheological fluid based compliant robot manipulator. *Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on*, pp. 370–375. DOI: 10.1109/ROBIO.2010.5723355.

Alonso et al. 1992 — Alonso, Marcelo and Finn, Edward J. 1992. *Physics*. Workingham: Addison-Wesley. ISBN: 0201565188.

Angeles 2002 — Angeles, Jorge. 2002. *Fundamentals of Robotic Mechanical Systems*. 2nd ed. Secaucus, NJ, USA: Springer. ISBN: 038795368X.

ANSI/RIA R15.06-2012 — American, National Standards Institute. 2012. *Industrial Robots and Robot Systems - Safety Requirements*. Washington D.C.: American National Standards Institute.

Bell et al. 2005 — Bell, Graeme and Livesey, Mike. 2005. The Existence of Local Minima in Local-Minimum-Free Potenitial Surfaces. *Proceedings of TARDOS2005 (Towards Autonomic Robotics Systems)*, pp. 15–20.

Bicchi et al. 2004 — Bicchi, Antonio and Tonietti, Giovanni. 2004. Fast and "soft-arm" tactics [robot arm design]. *Robotics Automation Magazine, IEEE* **11** (2), pp. 22–33. DOI: 10.1109/MRA.2004.1310939.

Bischoff et al. 2010 Bischoff, Rainer, Kurth, Johannes, Schreiber, Guenter, Koeppe, Ralf, Albu-Schaeffer, Alin, Beyer, Alexander, Eiberger, Oliver, Haddadin, Sami, Stemmer, Andreas, Grunwald, Gerhard and Hirzinger, Gerhard. 2010. The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing. *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1–8.

Brisan et al. 2011 Brisan, Cornel and Csiszar, Akos. 2011. Computation and analysis of the workspace of a reconfigurable parallel robotic system. *Mechanism and Machine Theory* **46** (11), pp. 1647–1668. DOI: 10.1016/j.mechmachtheory.2011.06.014.

Brock et al. 2000 Brock, Oliver and Khatib, Oussama. 2000. Real-time replanning in high-dimensional configuration spaces using sets of homotopic paths. *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on.* vol. 1, 550–555 vol.1. DOI: 10.1109/ROBOT.2000.844111.

Brock et al. 2002 Brock, Oliver and Khatib, Oussama. 2002. Elastic Strips: A Framework for Motion Generation in Human Environments. *The International Journal of Robotics Research* **21** (12), pp. 1031–1052.

Burelli et al. 2009 Burelli, Paolo and Jhala, Arnav. 2009. Dynamic Artificial Potential Fields for Autonomous Camera Control. *Artificial Intelligence In Interactive Digital Entertainment Conference*. Palo Alto, California, USA: AAAI.

Chatterjee 2011 — Chatterjee, Anirban. 2011. Motion planning approach that produces critical point-free configuration space. *Electronics Letters* **47** (19), pp. 1073–1075. DOI: 10.1049/el.2011.2094.

Chawla et al. 2000 — Chawla, Anoop, Mohan, Dinesh, Sharma, Vivek and Kajzer, Janusz. 2000. Safer Truck Front Design for Pedestrian Impacts. *Journal of Crash Prevention and Injury Control* **2** (1), pp. 33–43. DOI: 10.1109/MRA.2004.1310939.

Chen et al. 2009 — Chen, Fei, Di, Pei, Huang, Jian, Sasaki, H. and Fukuda, T. 2009. Evolutionary artificial potential field method based manipulator path planning for safe robotic assembly. *Micro-NanoMechatronics and Human Science, 2009. MHS 2009. International Symposium on*, pp. 92–97. DOI: 10.1109/MHS.2009.5352075.

Corke et al. 2000 — Corke, Peter, Trevelyan, James, Brock, Oliver and Khatib, Oussama. 2000. Elastic strips: A framework for integrated planning and execution. *Experimental Robotics VI*. vol. 250. Lecture Notes in Control and Information Sciences. Berlin: Springer, pp. 329–338. ISBN: 978-1-85233-210-5. DOI: 10.1007/BFb0119411.

Csiszar et al. 2012 — Csiszar, Akos, Drust, Manuel, Dietz, Thomas, Verl, Alexander and Brisan, Cornel. 2012. Dynamic and Interactive Path Planning and Collision Avoidance for an Industrial Robot Using Artificial Potential Field Based Method. *Mechatronics*. ed. by Ryszard Jabloblski and Tomas Bezina. Berlin/Heidelberg: Springer, pp. 413–421. ISBN: 978-3-642-23244-2. DOI: 10.1007/978-3-642-23244-2-50.

De Santis et al. 2007

De Santis, Agostino, Lippiello, Vincenzo, Siciliano, Bruno and Villani, Luigi. 2007. Human-Robot Interaction Control Using Force and Vision. *Advances in Control Theory and Applications*. ed. by Claudio Bonivento, Lorenzo Marconi, Carlo Rossi and Alberto Isidori. vol. 353. Lecture Notes in Control and Information Sciences. Berlin / Heidelberg: Springer, pp. 51–70. ISBN: 978-3-540-70700-4. DOI: 10.1007/978-3-540-70701-1-3.

De Santis et al. 2008

De Santis, Augustina, Siciliano, Bruno, De Luca, Alessandro and Bicchi, Antonio. 2008. An atlas of physical human robot interaction. *Mechanism and Machine Theory* **43** (3), pp. 253–270. DOI: 10.1016/j.mechmachtheory.2007.03.003.

Dietz et al. 2010

Dietz, Thomas, Pott, Andreas and Verl, Alexander. 2010. Simulation of the Stopping Behavior of Industrial Robots. *New Trends in Mechanism Science*. ed. by Doina Pisla, Marco Ceccarelli, Manfred Husty and Burkhard Corves. vol. 5. Mechanisms and Machine Science. Heidelberg/London/New-York: Springer, pp. 369–376. ISBN: 978-90-481-9689-0. DOI: 10.1007/978-90-481-9689-043.

Dietz et al. 2011

Dietz, Thomas and Verl, Alexander. 2011. Simulation of the stopping behavior of industrial robots using a complementarity-based approach. *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on*, pp. 428–433. DOI: 10.1109/AIM.2011.6027053.

DIN EN 1175-1:06-2011

Normung, Deutsches Institut für. 2011. *Sicherheit von Flurförderzeugen - Elektrische Anforderungen - Teil 1: Allgemeine Anforderungen für Flurförderzeuge mit batterieelektrischem Antrieb*. Berlin: Beuth.

Duchaine et al. 2009

Duchaine, Vincent, Lauzier, Nicolas, Baril, Mathieu, Lacasse, Marc-Antoine and Gosselin, Clement. 2009. A flexible robot skin for safe physical human robot interaction. *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 3676–3681. DOI: 10.1109/ROBOT.2009.5152595.

Ebert et al. 2002 Ebert, Dirk M and Henrich, Dominik D. 2002. Safe human-robot-cooperation: image-based collision detection for industrial robots. *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. vol. 2, 1826–1831 vol.2. DOI: 10.1109/IRDS.2002.1044021.

Fei et al. 2004 Fei, Yanqiong, Fuqiang, Ding and Xifang, Zhao. 2004. Collision-free motion planning of dual-arm reconfigurable robots. *Robotics and Computer-Integrated Manufacturing* **20** (4), pp. 351–357. DOI: 10.1016/j.rcim.2004.01.002.

Gao et al. 2008 Gao, Meijuan, Xu, Jin, Tian, Jingwen and Wu, Hao. 2008. Path Planning for Mobile Robot Based on Chaos Genetic Algorithm. *Natural Computation, 2008. ICNC '08. Fourth International Conference on*. vol. 4, pp. 409–413. DOI: 10.1109/ICNC.2008.627.

Garcia et al. 2007 — Garcia, Elena, Jimenez, Maria Antonia, De Santos, Pablo Gonzalez and Armada, Manuel. 2007. The evolution of robotics research. *Robotics Automation Magazine, IEEE* **14** (1), pp. 90–103. DOI: 10.1109/MRA.2007.339608.

Ge et al. 2002 — Ge, Shuzhi Sam and Cui, Yun J. 2002. Dynamic Motion Planning for Mobile Robots Using Potential Field Method. *Autonomous Robots* **13** (3), pp. 207–222. DOI: 10.1023/A:1020564024509.

Gecks et al. 2005 — Gecks, Thorsten and Henrich, Dominik. 2005. Human-robot cooperation: safe pick-and-place operations. *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pp. 549–554. DOI: 10.1109/ROMAN.2005.1513837.

Graf et al. 2009 — Graf, Jürgenand, Czapiewski, Piotr and Wörn, Heinz. 2009. Incorporating Novel Path Planning Method into Cognitive Vision System for Safe Human-Robot Interaction. *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, 2009. COMPUTATIONWORLD '09. Computation World:* pp. 443–447. DOI: 10.1109/ComputationWorld.2009.33.

Graf et al. 2010 — Graf, Jürgenand, Czapiewski, Piotr and Wörn, Heinz. 2010. Evaluating Risk Estimation Methods and Path Planning for Safe Human- Robot Cooperation. *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp. 1–7.

Haddadin et al. 2008

Haddadin, Sami, Albu-Schaffer, Alin, De Luca, Alessandro and Hirzinger, Gerd. 2008. Collision detection and reaction: A contribution to safe physical Human-Robot Interaction. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 3356–3363. DOI: 10.1109/IROS.2008.4650764.

Hagelback et al. 2008

Hagelback, Johan and Johansson, Stefan J. 2008. The rise of potential fields in real time strategy bots. *Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference*. Menlo Park, California, USA: AAAI Press. ISBN: 978-1-57735-391-1.

Heinzmann et al. 2003

Heinzmann, Jochen and Zelinsky, Alexander. 2003. Quantitative Safety Guarantees for Physical Human-Robot Interaction. *The International Journal of Robotics Research* **22** (7-8), pp. 479–504. DOI: 10.1177/02783649030227004.

Helble et al. 2007

Helble, Heiko and Cameron, Stephen. 2007. 3-D Path Planning and Target Trajectory Prediction for the Oxford Aerial Tracking System. *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1042–1048. DOI: 10.1109/ROBOT.2007.363122.

Hui et al. 2009

Hui, Nirmal Baran and Pratihar, Dilip Kumar. 2009. A comparative study on some navigation schemes of a real robot tackling moving obstacles. *Robotics and Computer-Integrated Manufacturing* **25** (4–5), pp. 810–828. DOI: 10.1016/j.rcim.2008.12.003.

Hwang et al. 1992 — Hwang, Yong K. and Ahuja, Narendra. 1992. A potential field approach to path planning. *Robotics and Automation, IEEE Transactions on* **8** (1), pp. 23–32. DOI: 10.1109/70.127236.

Ikuta et al. 2003 — Ikuta, Koji, Ishii, Hideki and Nokata, Makoto. 2003. Safety Evaluation Method of Design and Control for Human-Care Robots. *The International Journal of Robotics Research* **22** (5), pp. 281–297. DOI: 10.1177/0278364903022005001.

ISO 10218:2011 — ISO, International Organization for Standardization. 2011. *Robots and robotic devices – Safety requirements for industrial robots*. Geneve: International Organization for Standardization.

ISO 12100:2011 — International, Organization for Standardization. 2011. *Safety of machinery – General principles for design – Risk assessment and risk reduction*. Geneva: International Organization for Standardization.

ISO 13849-1:2006 — International, Organization for Standardization. 2006. *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*. Geneve: International Organization for Standardization.

ISO 13854:1996 — International, Organization for Standardization. 1996. *Safety of machinery – Minimum gaps to avoid crushing of parts of the human body*. Geneve: International Organization for Standardization.

ISO 13855:2010 International, Organization for Standardization. 2010. *Safety of machinery – Positioning of protective equipment with respect to the approach speeds of parts of the human body*. Geneve: International Organization for Standardization.

ISO 14118:2000 International, Organization for Standardization. 2000. *Safety of machinery – Prevention of unexpected start-up*. Geneve: International Organization for Standardization.

ISO 14119:2013 International, Organization for Standardization. 2013. *Safety of machinery – Interlocking devices associated with guards – Principles for design and selection*. Geneve: International Organization for Standardization.

ISO 14120:2002 International, Organization for Standardization. 2002. *Safety of machinery – Guards – General requirements for the design and construction of fixed and movable guards*. Geneve: International Organization for Standardization.

ISO 14121-1:2007 International, Organization for Standardization. 2007. *Safety of machinery – Risk assessment – Part 1: Principles*. Geneve: International Organization for Standardization.

ISO/TR 14121-2:2012 International, Organization for Standardization. 2012. *Safety of machinery – Risk assessment – Part 2: Practical guidance and examples of methods*. Geneve: International Organization for Standardization.

Kanaan et al. 2006 — Kanaan, Daniel, Wenger, Philippe and Chablat, Damien. 2006. Workspace Analysis of the Parallel Module of the VERNE Machine. *Problems of Mechanics* **25** (4), pp. 26–42.

Khatib 1986 — Khatib, Oussama. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotic Research* **5** (1), pp. 90–98.

Kim et al. 2004 — Kim, Dong Hun, Wang, Hua and Shin, Seiichi. 2004. Decentralized control of autonomous swarm systems using artificial potential functions: analytical design guidelines. *Decision and Control, 2004. CDC. 43rd IEEE Conference on.* vol. 1, 159–164 Vol.1. DOI: 10.1109/CDC.2004.1428623.

Kim et al. 1992 — Kim, J-O and Khosla, Pradeep K. 1992. Real-time obstacle avoidance using harmonic potential functions. *Robotics and Automation, IEEE Transactions on* **8** (3), pp. 338–349. DOI: 10.1109/70.143352.

Kim et al. 2008 — Kim, Seong-Tae and Chang, Pyung. 2008. Safety-Ensuring Systematic Design for Service Robots. *Smart Homes and Health Telematics.* ed. by Sumi Helal, Simanta Mitra, Johnny Wong, Carl Chang and Mounir Mokhtari. vol. 5120. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 208–217. ISBN: 978-3-540-69914-9. DOI: 10.1007/978-3-540-69916-3_24.

Krogh 1984 — Krogh, Bruce H. 1984. A Generalized Potential Field Approach to Obstacle Avoidance Control. *Proceedings of ASME Conference of Robotic Research: The Next Five Years and Beyond*. 14-16, August, 1984, Bethlehem, PA, USA: RI/SME.

Kuffner et al. 2000 — Kuffner, James J. and LaValle, Steven M. 2000. RRT-connect: An efficient approach to single-query path planning. *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*. vol. 2, 995–1001 vol.2. DOI: 10.1109/ROBOT.2000.844730.

Kulic et al. 2004 — Kulic, Dana and Croft, Elizabeth. 2004. Safe planning for human-robot interaction. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*. vol. 2, 1882–1887 Vol.2. DOI: 10.1109/ROBOT.2004.1308098.

Kulic et al. 2005 — Kulic, Dana and Croft, Elizabeth. 2005. Safe planning for human-robot interaction. *Journal of Robotic Systems* **22** (7), pp. 383–396.

Kulic et al. 2006 — Kulic, Dana and Croft, Elizabeth. 2006. Real-time safety for human robot interaction. *Robotics and Autonomous Systems* **54** (1), pp. 1–12. DOI: 10.1016/j.robot.2005.10.005.

Lacevic et al. 2010a — Lacevic, Bakir and Rocco, Paolo. 2010a. Kinetostatic danger field - a novel safety assessment for human-robot interaction. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 2169–2174. DOI: 10.1109/IROS.2010.5649124.

Lacevic et al. 2010b Lacevic, Bakir and Rocco, Paolo. 2010b. Towards a complete safe path planning for robotic manipulators. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 5366–5371. DOI: 10.1109/IROS.2010.5650945.

Laffranchi et al. 2009 Laffranchi, Matteo, Tsagarakis, Nikolaos G and Caldwell, Darwin G. 2009. Safe human robot interaction via energy regulation control. *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, pp. 35–41. DOI: 10.1109/IROS.2009.5354803.

Latombe 1991 Latombe, Jean-Claude. 1991. *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers. ISBN: 079239206X.

LaValle 2006 LaValle Stephen, M. 2006. *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press. ISBN: 9780521862059.

Lee et al. 2006 Lee, Leng-Feng and Krovi, Venkat. 2006. A Standardized Testing-Ground for Artificial Potential-Field based Motion Planning for Robot Collectives. *Proceedings of the 2006 Performance Metrics for Intelligent Systems Workshop, Gaithersburg, MD*.

Liu et al. 2005 Liu, Hong, Deng, Xuezhi and Zha, Hongbin. 2005. A planning method for safe interaction between human arms and robot manipulators. *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2724–2730. DOI: 10.1109/IROS.2005.1545241.

Lozano-Perez 1983 — Lozano-Perez, Tomas. 1983. Spatial Planning: A Configuration Space Approach. *Computers, IEEE Transactions on* **C-32** (2), pp. 108–120. DOI: 10.1109/TC.1983.1676196.

Lu et al. 2008a — Lu, Yi, Hu, Bo and Xu, Jia-Yin. 2008a. Kinematics analysis and solution of the active/passive forces of a 4SPS+SPR parallel machine tool. *The International Journal of Advanced Manufacturing Technology* **36** (1), pp. 178–187. DOI: 10.1007/s00170-006-0833-7.

Lu et al. 2008b — Lu, Yi, Shi, Yan and Hu, Bo. 2008b. Solving reachable workspace of some parallel manipulators by computer-aided design variation geometry. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. vol. 9, pp. 1773–1781. DOI: 10.1243/09544062JMES1069.

Macho et al. 2009 — Macho, Erik, Altuzarra, Oscar, Amezua, Enrique and Hernandez, Alfonso. 2009. Obtaining configuration space and singularity maps for parallel manipulators. *Mechanism and Machine Theory* **44** (11), pp. 2110–2125. DOI: 10.1016/j.mechmachtheory.2009.06.003.

Merlet 2002 — Merlet, Jean-Pierre. 2002. *Parallel Robots*. Dordrecht: Kluwer Academic Publishers. ISBN: 978-1-4020-0385-1.

Morita et al. 1999 — Morita, Toshio, Iwata, Hiroyasu and Sugano, Shigeki. 1999. Development of human symbiotic robot: WENDY. *Proceedings of the IEEE International Conference on Robotics and Automation*. vol. 4, 3183–3188 vol.4. DOI: 10.1109/ROBOT.1999.774083.

Najmaei et al. 2009 Najmaei, Nima and Kermani, Mehrdad R. 2009. Superquadric obstacle modeling and a danger evaluation method with applications in safe planning for human-safe industrial robots. *Technologies for Practical Robot Applications, 2009. TePRA 2009. IEEE International Conference on*, pp. 129–134. DOI: 10.1109/TEPRA.2009.5339633.

Najmaei et al. 2011 Najmaei, Nima and Kermani, Mehrdad R. 2011. Applications of artificial intelligence in safe human–robot interactions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **41** (2), pp. 448–459.

Nokata et al. 2002 Nokata, Makoto, Ikuta, Koji and Ishii, Hideki. 2002. Safety-optimizing method of human-care robot design and control. *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*. vol. 2. IEEE, pp. 1991–1996.

Ogorodnikova 2009 Ogorodnikova, Olesya. 2009. How Safe the Human-Robot Coexistence Is? Theoretical Presentation. *Acta Polytechnica Hungarica* **6** (4), pp. 51–74.

Pan et al. 2012 Pan, Zengxi, Polden, Joseph, Larkin, Nathan, Duin, Stephen Van and Norrish, John. 2012. Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing* **28** (2), pp. 87–94. DOI: 10.1016/j.rcim.2011.08.004.

Pedrocchi et al. 2009

Pedrocchi, Nicola, Malosio, Matteo and Tosatti, L Molinari. 2009. Safe obstacle avoidance for industrial robot working without fences. *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 3435–3440. DOI: 10.1109/IROS.2009.5353980.

Petti et al. 2005 Petti, Stephane and Fraichard, Thierry. 2005. Safe motion planning in dynamic environments. *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2210–2215. DOI: 10.1109/IROS.2005.1545549.

Pike 2001 Pike, Ralph W. 2001. *Optimization for Engineering Systems*. Luisiana, USA: Louisiana State University. ISBN: 9780442275815.

Raja et al. 2012 Raja, Purushothaman and Pugazhenthi, Sivagurunathan. 2012. Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences* **7** (9), pp. 1314–1320. DOI: 10.5897/IJPS11.1745.

Ur-Rehman et al. 2009

Ur-Rehman, Raza, Caro, Stephane, Chablat, Damien and Wenger, Philippe. 2009. Path placement optimization of manipulators based on energy consumption: Application to the orthoglide 3-axis. *Transactions of the Canadian Society for Mechanical Engineering* **33**, pp. 1–19.

Ur-Rehman et al. 2010

Ur-Rehman, Raza, Caro, Stephane, Chablat, Damien and Wenger, Philippe. 2010. Multi-objective path placement optimization of parallel kinematics machines based on energy consumption, shaking forces and maximum actuator torques: Application to the Orthoglide. *Mechanism and Machine Theory* **45** (8), pp. 1125–1141. DOI: 10.1016/j.mechmachtheory.2010.03.008.

Reimann et al. 2010

Reimann, Hendrik, Iossifidis, Ioannis and Schoner, Gregor. 2010. Generating collision free reaching movements for redundant manipulators using dynamical systems. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 5372–5379. DOI: 10.1109/IROS.2010.5650603.

Ren et al. 2007

Ren, Jing, McIsaac, Kenneth A., Patel, Rajni V. and Peters, Terry M. 2007. A Potential Field Model Using Generalized Sigmoid Functions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **37** (2), pp. 477–484. DOI: 10.1109/TSMCB.2006.883866.

Sfeir et al. 2011

Sfeir, Joe, Saad, Maarouf and Saliah-Hassane, Hamadou. 2011. An improved Artificial Potential Field approach to real-time mobile robot path planning in an unknown environment. *Robotic and Sensors Environments (ROSE), 2011 IEEE International Symposium on*, pp. 208–213. DOI: 10.1109/ROSE.2011.6058518.

Shiller et al. 1989 Shiller, Zvi and Dubowsky, Steven. 1989. Robot Path Planning with Obstacles, Actuator, Gripper, and Payload Constraints. *The International Journal of Robotics Research* **8** (6), pp. 3–18. DOI: 10.1177/027836498900800601.

Al-Sultan et al. 1996 Al-Sultan, Khalid S. and Aliyu, Mohammad Dikko. 1996. A new potential field-based algorithm for path planning. *Journal of Intelligent & Robotic Systems* **17** (3), pp. 265–282. DOI: 10.1007/BF00339664.

Svenstrup et al. 2010 Svenstrup, Mikael, Bak, Thomas and Andersen, Hans Jørgen. 2010. Trajectory planning for robots in dynamic human environments. *Intelligent robots and systems (IROS), 2010 IEEE/RSJ international conference on*. IEEE, pp. 4293–4298.

Svenstrup et al. 2009 Svenstrup, Mikael, Tranberg, Soren, Andersen, Hans Jorgen and Bak, Thomas. 2009. Pose estimation and adaptive robot behaviour for human-robot interaction. *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, pp. 3571–3576.

Tan et al. 2009 Tan, Jeffrey Too Chuan, Duan, Feng, Zhang, Ye, Kato, Ryu and Arai, Tamio. 2009. Safety design and development of human-robot collaboration in cellular manufacturing. *Automation Science and Engineering, 2009. CASE 2009. IEEE International Conference on*, pp. 537–542. DOI: 10.1109/COASE.2009.5234120.

Tao et al. 2011 — Tao, Pengfei, Jin, Sheng and Wang, Dianhai. 2011. Car-following model based on artificial potential field. *Journal of Southeast University. Natural Science Edition* **41** (4), pp. 854–858.

Tonietti et al. 2005 — Tonietti, Giovanni, Schiavi, Riccardo and Bicchi, Antonio. 2005. Design and control of a variable stiffness actuator for safe and fast physical human/robot interaction. *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, pp. 526–531.

Vadakkepat et al. 2000 — Vadakkepat, Prahlad, Tan, Kay Chen and Ming-Liang, Wang. 2000. Evolutionary artificial potential fields and their application in real time robot path planning. *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*. vol. 1, 256–263 vol.1. DOI: 10.1109/CEC.2000.870304.

Valavanis et al. 2000 — Valavanis, Kimon P, Hebert, Timothy, Kolluru, Ramesh and Tsourveloudis, Nikos. 2000. Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* **30** (2), pp. 187–196. DOI: 10.1109/3468.833100.

Volpe et al. 1990 — Volpe, Richard and Khosla, Pradeep. 1990. Manipulator control with superquadric artificial potential functions: theory and experiments. *Systems, Man and Cybernetics, IEEE Transactions on* **20** (6), pp. 1423–1436. DOI: 10.1109/21.61211.

Wagner et al. 2012 — Wagner, Glenn, Kang, Minsu and Choset, Howie. 2012. Probabilistic path planning for multiple robots with subdimensional expansion. *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2886–2892. DOI: 10.1109/ICRA.2012.6225297.

Wahid et al. 2008 — Wahid, Abdel, Mabrouk, Mohamed Hussien and McInnes, CR. 2008. Solving the potential field local minimum problem using internal agent states. *Robots and Autonomous Systems* **56** (12), pp. 1050–1060.

Xu et al. 2010 — Xu, Lu and Chow, Tommy Wai Shing. 2010. Self-Organizing Potential Field Network: A New Optimization Algorithm. *Neural Networks, IEEE Transactions on* **21** (9), pp. 1482–1495. DOI: 10.1109/TNN.2010.2047264.

Yamada et al. 2005 — Yamada, Yoji, Morizono, Tetsuya, Umetani, Yoji and Takahashi, Hitoshi. 2005. Highly soft viscoelastic robot skin with a contact object-location-sensing capability. *IEEE Transactions on Industrial electronics* **52** (4), pp. 960–968.

Yanai et al. 2011 — Yanai, Haruo, Takeuchi, Kei and Takane, Yoshio. 2011. *Projection Matrices, Generalized Inverse Matrices, and Singular Value Decomposition*. New York, NY: Springer. ISBN: 978-1-4419-9887-3. DOI: 10.1007/978-1-4419-9887-3_5.

Yang et al. 2005a — Yang, Jingzhou and Abdel-Malek, Karim. 2005a. Singularities of manipulators with non-unilateral constraints. *Robotica* **23** (5), pp. 543–553. DOI: 10.1017/S0263574704001262.

Yang et al. 2005b — Yang, Jingzhou and Abdel-Malek, Karim. 2005b. Singularities of manipulators with non-unilateral constraints. *Robotica* **23** (05), pp. 543–553. DOI: 10.1017/S0263574704001262.

Yannakakis et al. 2004 — Yannakakis, Georgios N., Levine, John and Hallam, John. 2004. An evolutionary approach for interactive computer games. *Evolutionary Computation, 2004. CEC2004. Congress on.* vol. 1, 986–993 Vol.1. DOI: 10.1109/CEC.2004.1330969.

Yao et al. 2010 — Yao, Junfeng, Lin, Chao, Xie, Xiaobiao, Wang, Andy JuAn and Hung, Chih-Cheng. 2010. Path planning for virtual human motion using improved A* star algorithm. *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on.* IEEE, pp. 1154–1158.

Yun et al. 1997 — Yun, Xiaoping and Tan, Ko-Cheng. 1997. A wall-following method for escaping local minima in potential field based motion planning. *Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference on*, pp. 421–426. DOI: 10.1109/ICAR.1997.620216.

Zhang et al. 2006 — Zhang, Bing, Chen, Wanmi and Fei, Minrui. 2006. An Optimized Method for Path Planning Based on Artificial Potential Field. *Intelligent Systems Design and Applications, 2006. ISDA '06. Sixth International Conference on.* vol. 3, pp. 35–39. DOI: 10.1109/ISDA.2006.11.

Zou et al. 2003 — Zou, Xi-Yong and Zhu, Jing. 2003. Virtual local target method for avoiding local minimum in potential field based robot navigation. *Journal of Zhejiang University-Science A* **4** (3), pp. 264–269. DOI: jzus.2003.0264.