POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Obstacle Avoidance for an Holonomic Robotic Manipulator with Constraint-Based Model Predictive Control

Laurea Magistrale in utomation and Control Engineering - Ingegneria dell'Automazione

Author: Gonzalo Jesús, Meza Pérez

Advisor: Prof. Lorenzo Fagiano

Co-advisor: Kristoffer Løwenstein

Academic year: 2023-2024

---

1. Introduction

In recent years, the integration of robotic arms into real-world applications has opened a new era of automation and efficiency in different industries. Nevertheless, new challenges have arisen with the ever evolving technologies and applications. Robotic manipulators have been widely included into manufacturing industries, automation and lastly into mobile robotic manipulators. Mobile manipulation systems (MMSs) are robotic systems consisting of one or more robot arms mounted on a mobile base, the coupling of both enable the manipulator to navigate and expand its operational workspace, all while preserving its manipulative capabilities. Nevertheless, with the increasing number of applications it seems inevitable that more dexterity(than 3-DOF) of the robotic manipulators will be required independently from the mobile part of the robot (e.g. for confined spaces or coexistence with humans). This is the reason why this work is focused on a novel solution for obstacle avoidance and inverse kinematics control tailored for a 6-degrees-of-freedom robotic arm operating in unknown environments, without loss of generality of the proposed approach that can be extended to the mobile part.

The core mission is to guide the robotic arm safely toward the successful execution of desired tasks in an environment that is characterized to be unknown, with fixed or dynamic obstacles, relying only on partial environmental information provided by an exteroceptive sensor. The complete architecture of the proposed system embraces a hierarchical control system, comprising a high-level strategy for generating obstacle-free trajectories, a medium-level Model Predictive Controller (MPC) as Inverse Kinematics and reactive obstacle avoidance, and assumed low-level controllers for joint set-point tracking. More in detail, the overall framework will address the tasks of Local Path Planning and End-Effector positioning in the workspace.

2. Problem Statement

Local Path Planning emerged due to new robotics applications within dynamical environments where the robot must be able to plan its motion online and without an entire knowledge of its environment. These local methods rely solely on local environmental information(provided by sensors), suppressing a previous knowledge of a complete map. In this regard, a new methodology for Local Path Planners based on the generation of Convex Regions of Obstacle-Free Space has been acquiring popularity during the last years and specifically in the field of robotic manipulators [9][7][4]. This method allows the generation of regions that are convex and collision free that surpass efficiently the computational burden of collision checking from standard motion planning techniques. In detail, the trajectory planning under an approximated obstacle free region can be approached in Configuration space or Cartesian space, and can be generated based on convex optimization techniques(were constraints can be enforced) or efficient algorithms. For example in [4] the generation of convex polytopes is based Cartesian space while in [9] [7] in configuration spaces, both approaches by means of convex optimization techniques. Nevertheless, the computational time presented for a 7DOF robot in a cluttered environment is in order of the 10^3 seconds which can compromise an online iterative algorithm. On the other hand, new computationally efficient methods based on Cartesian space were developed in [3][8] showcasing its potential for an online real implementation which is the aim of this work.

On the other hand, the task of positioning the end-effector in the workspace should involve an Inverse Kinematics algorithm to map the trajectory generated in task space into joint space while reactively avoiding obstacles of the environment and respecting other enforced constraints. During the last years with the increasing computational power, online-solvable instantaneous constrained optimization has become increasingly popular due to its capacity to easily incorporate kinematic functions, position, velocity, and acceleration constraints into the problem formulation. For example, recent works [1][2] propose Model Predictive Control(MPC) to account for the system evolution over a prediction horizon. This method provides a state feedback controller which is optimal over the N future time steps and complies with kinematic constraints. Nevertheless, in this architecture the Inverse Kinematics solution is given as a reference to the MPC which is formulated only in joint space.

Therefore, in [5] the operational space task as a reference of the MPC is included. This formulation allows to impose not only kinematic constraints of the joints but also secondary tasks in the operational space such collision avoidance. Even though the latter formulation shows significant advantages, the prediction of the future state of the system over an horizon has been a topic of discussion. Thus, with the aim to maintain the linear nature of the optimization problem over an horizon, as proposed in [6][1][2] a linearization of the task at each iteration k is formulated by using the solution obtained at time $k-1$ to compute the future trajectory vector of joint positions q_{k+i} so that the overall linear nature of the QP problem is preserved. While the ultimate quadratic program (QP) is an approximation of the initial non-linear problem, it allows the computational time to be of the same order of magnitude of local methods(in the order of milliseconds), but still conferring the advantages of a predictive strategy.

On top of the mentioned approaches, this work proposes a structural division. From one side, the generation of obstacle-free regions[3][8] in Cartesian space for the purpose of trajectory generation by including the computational efficient algorithms in the current field of robotic manipulators. While for the positioning of the end effector, the present work proposes a QP formulation of the IK-MPC which includes the operational space task as a reference, a predictive strategy, direct acceleration constraints (not only joint and velocity constraints), jerk penalization in the cost function, self-collision and reactive obstacle avoidance as inequality constraints. The proposed partition guarantees a double layer of safety(obstacle free trajectory generation and reactive obstacle avoidance) for obstacle avoidance which allows a safe motion of the manipulator towards a target position in a obstacle characterized environment .

3. Proposed Method

As mentioned, the proposed approach can be decomposed in two main structures detailed in the following sections. In detail, the overall control architecture of the proposed method is shown in Figure 1.

As it can be observed, the Local Path Planner anticipates a generated trajectory (position X_D and velocity $\dot{X}_D$) to the Inverse Kinematics block, and the latter provides the joint references($q(k), \dot{q}(k)$) to the low level controllers

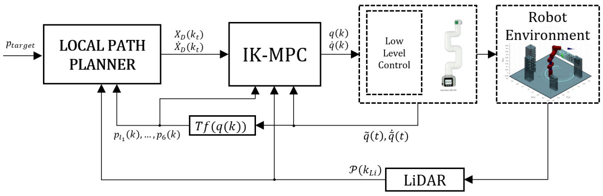

Figure 1: Block diagram scheme of the architecture

to perform the required motion. The feedback of the actual position of the robot is computed by its forward kinematic function $Tf(q(k))$ while the point-cloud $\mathcal{P}$ of the position of the obstacles in the environment is provided by the an exteroceptive LiDAR sensor mounted in the end effector of the robot.

The methodology adopted focuses in the development of modular simpler algorithms which together conforms the complete algorithm of the proposed solution. Therefore, this section is meant to present the formulation and algorithms belonging to the Local Path Planning stage and the Inverse Kinematics, both will serve as ground for the understanding of the complete structure presented in the last section.

3.1. Local Path Planning

3.1.1 Convex Approximation of the Free Space

As mentioned, the manipulator has an exteroceptive sensor used to gather information of its environment. The position and configuration of this LiDAR sensor mounted in the head of the robot recovers all the measurements of the closest obstacles to the robot within the range of measurement. Lets denote $\mathcal{P}(k_t) = [\rho_{Li,k_t}(0), ..., \rho_{Li,k_t}(N_{Li}-1)]$ the N_{Li} readings of the LiDAR sensor at time k_t, p_{EE} as the position of the end effector of the manipulator and N_{obs} the number of obstacles in the environment. The convex hull $\mathcal{W}$ represents the workspace of the robot and d_{step}, r_{Li} are fixed parameters. Therefore, for the purpose of generating obstacle free convex polytope $\mathcal{S}(k_t)$ an Algorithm (see Thesis Section 3.3.2) was developed:

$$\mathcal{S}(k_t) \leftarrow polytope\_gen(\mathcal{P}(k_t), \mathcal{W}, p_{EE}(k_t), d_{step}, r_{Li}).$$

Where $\mathcal{S}(k_t)$ is the intersection of the convex polytope and the feasible workspace $\mathcal{W}$ of the robot.

3.1.2 Obstacle Free Trajectory Generation

This section focuses on the proposed strategy for the local path planning of the 6-degree-of-freedom robotic manipulator operating in an unknown environment. The key challenge is to avoid obstacles in the environment that might impede the robot's movement toward its target location. The method involves a multi-step strategy, starting with a Temporary Target Shifting Strategy to the Trajectory Generation of the obstacle-free reference path to be fed to the Inverse Kinematics block.

The Temporary Target Shifting Strategy tackles the problem of finding an obstacle free path when an obstacle obstructs the robot's direct path to its target position. The proposed strategy selects a temporary target from the sensor readings within a specific threshold range in order to avoid trajectories towards an imminent obstacle. The choice of this temporary target involves vector-based strategies and it is detailed within the Algorithm presented in Thesis Section 3.3.3.

Following this strategy, the Trajectory Generation phase is deployed. Taking the convex under-approximation of free space, this strategy generates trajectories for the robot's movement from its current location to the temporary target inside the region. This process involves defining minimum jerk polynomial trajectories to ensure smooth, controlled, and vibration-free motion. The orientation of the robot's end effector is also considered, aligning it towards the target position.

Finally, exploiting the algorithms detailed above, the complete Obstacle Free Trajectory Generation algorithm is deployed (see Thesis Algorithm 3.5). It synthesizes and combine these steps to generate the robot's path, handling obstacle avoidance and orienting the robot towards its target location. The trajectory is developed to guarantee an obstacle-free path characterized by smooth polynomial motions.

A 3D representation of the position and orientation trajectories generated(result of the complete Algorithm) inside the obstacle free convex polytope is reported in Figure 2. Graphically, it can be observed that the orientation trajectory is iteratively moving towards the direction of the temporary target which is the desired behaviour.

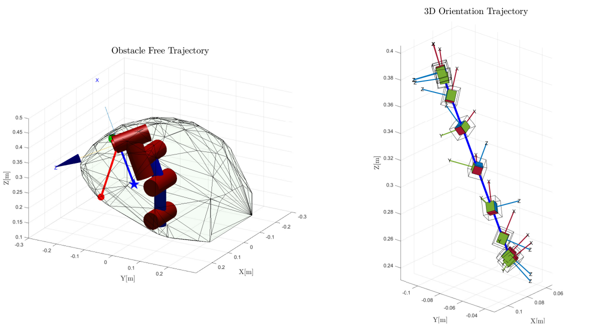

Figure 2: The convex free region is represented by the light-green geometry, the direct trajectory towards the target is represented by the solid red line, the trajectory towards the shifted target in solid blue and the orientation axis(x, y, z) are represented by the red, green and blue lines centered at the initial and final positions.

3.2. Inverse Kinematics as an Optimization Problem

In this section, the main formulation of the proposed approach for the Inverse Kinematics as an optimization problem is presented. First, the system model is defined, based on the discrete-time Linear Time-Invariant (LTI) system governing robotic kinematic chain. Second, the fundamental objective function is defined, denoted as $\mathcal{L}$, which the optimization routine seeks to minimize.

3.2.1 System Model

Considering the discrete time LTI system describing the linear kinematics as a dual-integrator system, whose dynamic equation can be written as :

$$\underbrace{\begin{bmatrix} q(k+1) \\ \dot{q}(k+1) \end{bmatrix}}_{z(k+1)} = \underbrace{\begin{bmatrix} I_n & T_s I_n \\ 0_{n\times n} & I_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} q(k) \\ \dot{q}(k) \end{bmatrix}}_{z(k)} + \underbrace{\begin{bmatrix} 0.5{T_s}^2 I_n \\ T_s I_n \end{bmatrix}}_{B} u(k) \tag{1}$$

Where $u(k) = \ddot{q}(k) \in \mathbb{R}^{n_u}$ and $z(0) = [q(0)^T, \dot{q}(0)^T]^T \in \mathbb{R}^{2n_q}$. The matrix $I_n \in \mathbb{R}^{n_q \times n_q}$ refers to an identity matrix of the dimension of the joints (in this case $n_q = 6$), T_s is the discrete integration time and k as the discrete time variable.

3.2.2 Cost Function

As discussed, the Inverse Kinematics problem considers a specific coordinate task $x_d(k)$ that the robot needs to reach at every time instant k while complying with motion constraints. First, a reference trajectory that can drive the robot given its current configuration $x_e(k) \leftarrow T_e^b(q(k))_{(4,1:3)}$ to a desired position $x_d(k)$ in M steps can be computed beforehand (e.g. through polynomial trajectories) obtaining a set of references $x_d(k+i), i = 1, ..., M$. Notice that the construction of this trajectory does not enforce any motion constraint in joint space. Then, a MPC structure of the Inverse Kinematics can be used to compute the sequence of control inputs to achieve the desired position in the operational space. The nature of the MPC formulation allows the computation of the control inputs at each control interaction, allowing it to deal with unforeseen events or dynamic scenarios. Moreover, as mentioned in Section **??**, constraints linearly dependent on the sequence of control inputs can be easily included. At each time instant k, the MPC problem is defined as finding a M sequence of future control inputs $U_k = [u(0|k)^T, u(1|k)^T, ..., u(M-1|k)^T]^T$ that minimizes an specific cost function. The proposed cost function for the purpose of this work is shown in Equation (2) where the operational error and operational error velocity were included as well the joint velocity, joint acceleration and jerk for penalization purposes.

$$\mathcal{L} = \sum_{i=1}^{4} J_i \tag{2}$$

Where $J_1, ..., J_4$ are the desired tasks to minimize defined as:

$$J_1 = \sum_{i=1}^{M} \| J_A(q_{k+i})\dot{q}_{k+i} - \dot{x}_{d_{k+i}} + K(x_{d_{k+i}} - x_e(q_{k+i})\|_W^2$$

$$J_2 = \sum_{i=1}^{M} \|\dot{q}_{k+i}\|_S^2$$

$$J_3 = \sum_{i=1}^{M} \|\ddot{q}_{k+i}\|_R^2$$

$$J_4 = \sum_{i=1}^{M-1} \|\Delta\ddot{q}_{k+i}\|^2_{R_{\Delta\ddot{q}}}$$

The jerk is defined as $\Delta\ddot{q}(k+i) = \ddot{q}(k+i) - \ddot{q}(k)$, M is the prediction horizon and K is a positive definite weighting matrix related to the convergence of the error. W, S, R and $R_{\Delta\ddot{q}}$ are symmetric and positive definite weighting matrices, and the following shorthand notation was employed:

$$\|x\|^2_Q = x^T Q x \in \mathbb{R} \tag{3}$$

Therefore, the formalization of the optimization problem at hand can be written as:

$$\underset{U_k}{min}\, \mathcal{L}(U_k, \hat{Q}_k, \dot{\hat{Q}}_k, X_d, \dot{X}_d, k) \tag{4}$$

Subject to :

$$z(i+1|k) = Az(i|k) + Bu(i|k) \tag{5a}$$
$$q(i|k) = C_1 z(i|k) \tag{5b}$$
$$\dot{q}(i|k) = C_2 z(i|k) \tag{5c}$$
$$A_{ineq} U(k) \leq b_{ineq} \tag{5d}$$

Notation $z(i|k)$ denotes the value of z at time $k+i$, predicted at time k. Where $\hat{Q}_k$ and $\dot{\hat{Q}}_k$ are the vectors related to the prediction through the horizon of the future joint position and velocity respectively. X_d and $\dot{X}_d$ are the vectors which refer to the position and velocity of a partial trajectory composed by M steps ahead. A_{ineq} and b_{ineq} are know inequality matrices defining the boundaries of the joint position, velocity and acceleration vectors. Additionally, the reactive Obstacle Avoidance function as well as the self-collision avoidance will be both also enforced as inequality constraints.
Finally, the optimization problem in (4) can be derived and solved as a Quadratic Programming(QP) problem (see Thesis Section 3.4.5). Thereby, in order to integrate the receding horizon strategy, a numerical simulation of the system model is required as well as a recursive implementation of the overall Inverse Kinematics-Model Predictive Control. To this end, a sequential iterative Algorithm was developed (refer to Thesis Algorithm 3.7).

In the current section, modular algorithms were developed and presented in order to build the proposed hierarchical architecture. As stated, both main Algorithms are subjected to the 3D LiDAR sensor readings that are also simulated by means of a developed Algorithm (refer to Thesis Algorithm 3.1). A graphical scheme representing the interaction of these Algorithms is deployed in Figure 3. As it can be observed, the proposed structure allows the inclusion of other components such an offline or sequential high level planner which can compute the referenced target p_{target} based on the specific application, or a customized low level control system which can run at a different discrete time. This characteristic makes the presented architecture adaptable and capable to be extended with other motion or control techniques.

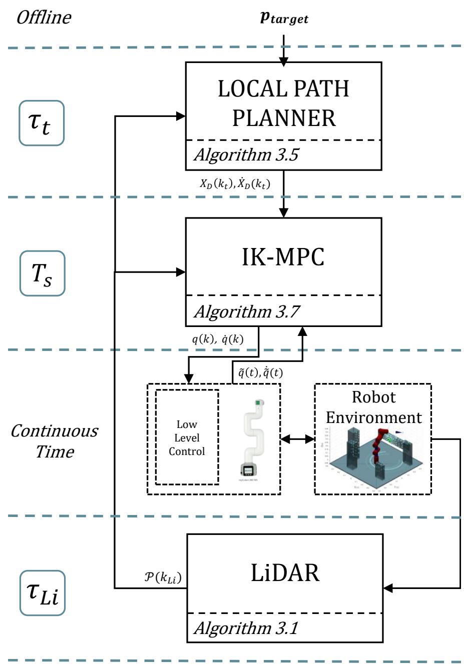

Figure 3: Block diagram scheme of the architecture

4. Simulation Results

The evaluation of the proposed method for a 6 degrees of freedom robotic arm in unknown environments provides insights into the effectiveness and versatility of the developed framework. The results will be summarized in an independent test for the IK-MPC and then an example of the overall framework in a challenging environment (for detailed results refer Thesis Chapter 4).
The tests were performed on a developed Digital

Twin of a MyCobot280 Robotic Arm model. For the purpose of a graphical representation of the results, several toolboxes will be used such as *Simulink-Simscape*(for dynamical motion simulation) and *PeterCorke Robotic Toolbox*(for a kinematic representation of the robot). Regarding the technical setup, the reported tests have been executed on a laptop equipped with an Intel core i7-11800H CPU, 16 GB of RAM , and MATLABR2023a version. For the optimization algorithm the **Active-Set** method was chosen accompanied by a **5-steps** ahead prediction horizon.

First, the IK-MPC will be tested with obstacles obstructing the generated trajectory. To this end, a linear polynomial trajectory was generated to a target position(returning to the same position) with a sphere shape obstacle in the middle of the path. The resultant performed trajectory is presented in 3D-space in Figure 4. It can be observed that the performed trajectory avoids the obstacle even though the trajectory was generated crossing the obstacle.

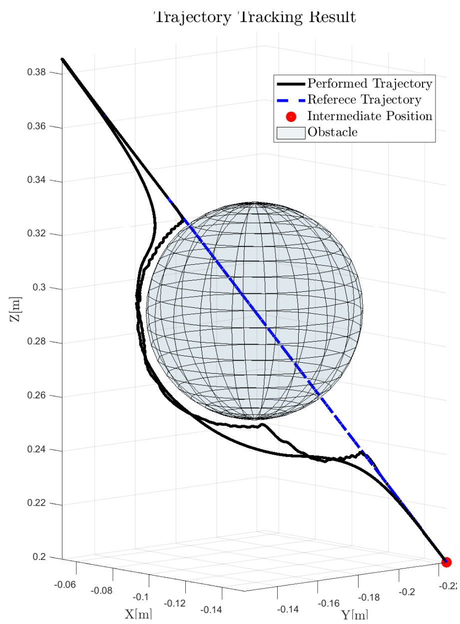

Figure 4: 3D Trajectory Tracking Results

Both algorithms, Trajectory Generation and IK-MPC, are shown independently a satisfactory performance according to what was required in the problem formulation. Now it is possible to test the coupled performance of the entire architecture in a challenging environment simulating a real-world robotic application. This scenario aims to challenge the robotic manipulator with a common task such as "picking an object under a table", where the environment will be also characterized with two additional aleatory obstacles generated within the workspace apart from the table. The manipulator will try to reach a target position $p_{target} = (0.0944, -0.0956, 0.3855)[m]$ (with no direct visibility of the target) and then return to its initial position $p_{init} = (-0.0056, -0.2156, 0.1500)[m]$ given an initial configuration of $q(0) = [\pi/2, 0, \pi/8, 0, -2.443, 0][rad]$. Graphically this initial state is represented in Figure 5.

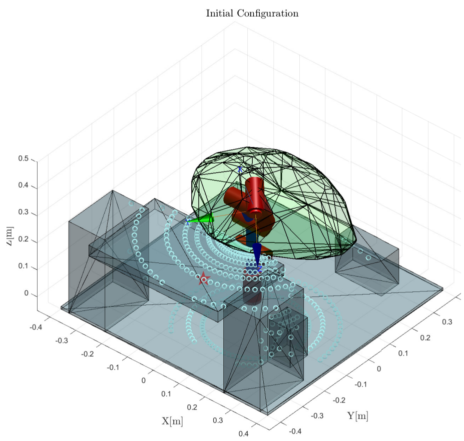

Figure 5: First Scenario: Initial Configuration

The complete 3D trajectory performed by the manipulator is shown in Figure 6 while a detailed task space trajectory tracking is shown in Figure 7. It can be observed that the reference orientation is not tracked as much as the position, this was expected since the cost function weights were focused most on the positioning part(for the results in joint positions, velocities and acceleration refer to Thesis Section 4.2.3.).

Finally, the distances related to the obstacle avoidance constraint are reported in Figure 8 while the ones related to the self-collision constraint in Figure 9. It should be noticed that the new position of the obstacles is updated every $\tau_{Li} = 0.5$, which leads in some cases to a sudden violation of the obstacle constraints, therefore, the control input(acceleration) saturates trying to return to the feasible region.

Additionally, for each trajectory, there was a time allocated to generate the obstacle free polytope and generate the trajectory, and a time to solve the IK. In this case, the robot reached the target and returned in 9 iterations, those val-

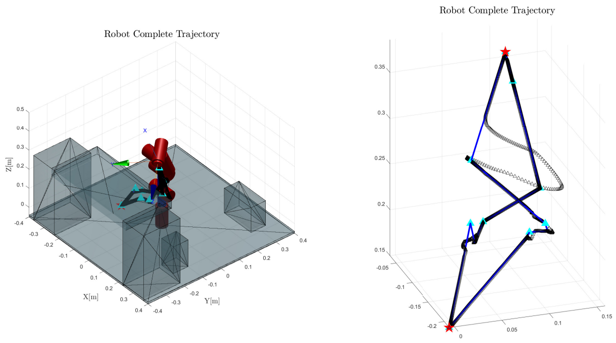

Figure 6: First Scenario: Complete Performed Trajectory.Red-Stars: Initial and target positions. Black triangle-dashed line: Trajectory performed. Blue-line: intermediate trajectories. Cyan triangles: Intermediate Shifted targets.

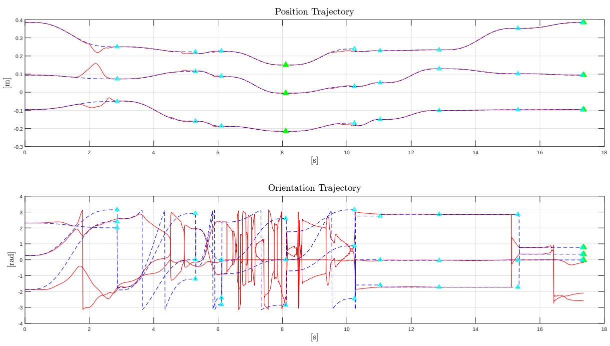

Figure 7: Performed Trajectory(solid red) and Generated Trajectory(blue dashed).

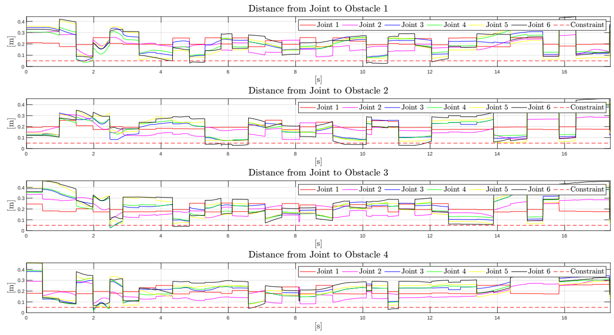

Figure 8: Distance from each link to each obstacle

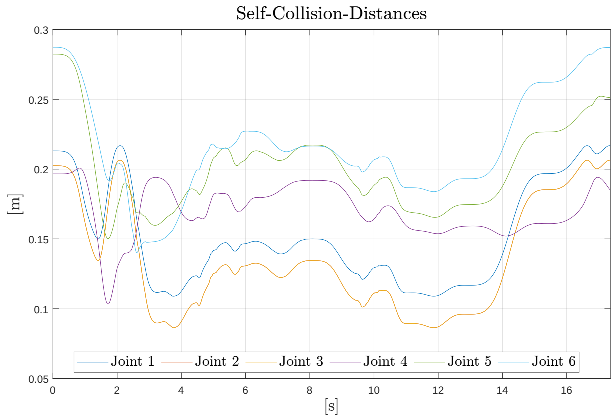

Figure 9: Self-Collision Distances. Constraint at 0.045 m

ues are reported in Table 1 respectively. From the table can be concluded that the collected time results show the efficiency of the method with a lower than $2ms$ for the IK time period. This shows the capability of the method to be used for an online implementation since the IK is solved much faster than the discrete time T_s. While a maximum of $1s$ for the trajectory generation, trajectory that can be computed in parallel while the robot is still reaching its target.
Moreover, it is essential to mention the possibilities of generating an obstacle free local map while the robot is reaching the target. This feature can be exploited by means of the the union of the intermediate convex polytopes generated, for this example, graphically this is represented in Figure 10.

5. Conclusions

This work presented a novel hierarchical control framework for a 6-DOF robotic manipulator operating in unknown and dynamic environments. Novel strategies for Local Path Planning and a constrained IK-MPC were proposed, this partition guaranteed a double layer of safety (obstacle free trajectory generation and reactive obstacle avoidance) for obstacle avoidance.
In the Local Path Planning block, a Convex Approximation of the Free Space was developed with the aim to generate obstacle free trajectories towards a target, and an efficient strategy for target shifting was presented and tested.
The Inverse Kinematics used a MPC approach, which exhibited improved performance compared to traditional one-step-ahead formulations. The IK-MPC achieved accurate trajectory following while satisfying the joint motion, obstacle and self-collision avoidance constraints. Despite the fact the robot is not redundant, the self-collision avoidance constraint allowed different reconfigurations of the robot while operating. The average time per iteration collected showed its potential for an online real implementation.

Iter.	Traj. Gen. Time $[s]$	Iter. Avg. Time IK-MPC $[s]$
1	1.0055	1.5326 $\times 10^{-3}$
2	0.4278	0.7157 $\times 10^{-3}$
3	0.1409	0.9011 $\times 10^{-3}$
4	0.1742	0.5413 $\times 10^{-3}$
5	0.3018	0.6899 $\times 10^{-3}$
6	0.1303	0.9900 $\times 10^{-3}$
7	0.1863	0.4813 $\times 10^{-3}$
8	0.8454	0.4748 $\times 10^{-3}$
9	0.1931	0.4990 $\times 10^{-3}$

Table 1: First Scenario: Time Execution Results.

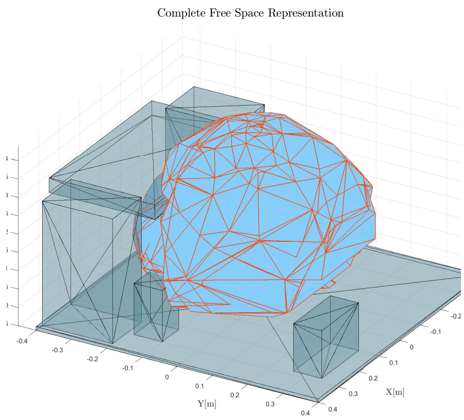

Figure 10: First Scenario: Complete Obstacle Free Region.

In conclusion, the developed hierarchical control framework exhibited promising capabilities for robust and adaptable robotic manipulator operation in complex environments. In fact, the proposed method was developed as general as possible to be able to include redundant, humanoids or hexapods robots in future works. In addition, the proposed structure allows the inclusion of other components that can compute the referenced target p_{target} based on the specific application, or a customized low level control system. The results obtained elucidates a path for continued advancements in the field of robotic control systems, with potential applications in diverse industrial and research domains.

References

[1] Giovanni Buizza Avanzini, Andrea Maria Zanchettin, and Paolo Rocco. Constraint-based model predictive control for holonomic mobile manipulators. pages 1473–1479, 2015.

[2] Giovanni Buizza Avanzini, Andrea Maria Zanchettin, and Paolo Rocco. Constrained model predictive control for mobile robotic manipulators. *Robotica*, 36(1):19–38, 2018.

[3] Leonardo Cecchin, Danilo Saccani, and Lorenzo Fagiano. G-beam: Graph-based exploration and mapping for autonomous vehicles. pages 1011–1016, 2021.

[4] Robin Deits and Russ Tedrake. *Computing Large Convex Regions of Obstacle-Free Space Through Semidefinite Programming*, pages 109–124. Springer International Publishing, Cham, Switzerland, 2015.

[5] Marco Faroni, Manuel Beschi, and Nicola Pedrocchi. An mpc framework for online motion planning in human-robot collaborative tasks. pages 1555–1558, 2019.

[6] Marco Faroni, Manuel Beschi, and Antonio Visioli. Predictive inverse kinematics for redundant manipulators: Evaluation in re-planning scenarios. *IFAC-PapersOnLine*, 51(22):238–243, 2018. 12th IFAC Symposium on Robot Control SYROCO 2018.

[7] Mark Petersen and Russ Tedrake. Growing convex collision-free regions in configuration space using nonlinear programming. 2023.

[8] Danilo Saccani, Leonardo Cecchin, and Lorenzo Fagiano. Multitrajectory model predictive control for safe uav navigation in an unknown environment. *IEEE Transactions on Control Systems Technology*, 31(5):1982–1997, 2023.

[9] Peter Werner, Alexandre Amice, Tobia Marcucci, Daniela Rus, and Russ Tedrake. Approximating robot configuration spaces with few convex sets using clique covers of visibility graphs. 2023.