Fusion Engineering and Design 202 (2024) 114334

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

# An obstacle avoidance path planning algorithm to simulate hyper redundant manipulators for tokamaks maintenance

Sara Buonocore [a,b,*], Andrea Zoppoli [a,b,d], Giuseppe Di Gironimo [a,b,c]

[a] Consorzio CREATE, Via Claudio 21, 80125 Napoli (NA), Italy
[b] Università degli Studi di Napoli Federico II - DII, Piazzale Vincenzo Tecchio, 80, 80125 Napoli (NA), Italy
[c] DTT S.C. a r.l., Via E. Fermi 45, I-00044, Frascati, Italy
[d] Università degli Studi di Padova, Via Gradenigo 6/A, 35131, Padova, Italy

ARTICLE INFO

*Keywords:*
Path planning
Collision avoidance
Hyper-redundant manipulator
Remote Maintenance
Tokamak
DTT

ABSTRACT

The present work proposes an obstacle avoidance path planning algorithm for virtual simulations of hyper-redundant manipulators, with the possibility to customize the optimization criteria to select the best trajectory given in output. For test purposes, the effectiveness of the proposed Inverse Kinematic algorithm has been tested by simulating the Remote Maintenance (RM) tasks conducted by the HyRMan: the Hyper Redundant Manipulator developed for the Divertor Tokamak Test (DTT) project. The algorithm has been employed to simulate some critical handling tasks of the First Wall (FW) modules, with a specific optimization criterion as an example of its potentialities.

## 1. Introduction

In 1999, an overall classification of robots was given by Robinson and Davies [1] into three categories according to their structural types and motion characteristics: discrete (or conventional) robots, Hyper-redundant manipulators and continuous robots (the last two categories are often merged due to their similarities [27]). The discrete robot is composed of single DOF rigid joints (no more than 7 DOF) and rigid links. For these robots, the pose change in the operation space can be realized by controlling the single DOF motion of each joint, that is generally actuated [26].

Despite they are the first appeared in literature and the most mature technology, discrete robots are suitable for working only in structured environments. Whether the inspection or maintenance operations must be performed in a narrow or confined space, conventional robots may often be inadequate to avoid obstacles, joint singularity, and joint overrun or to ensure a proper dexterous operating space. For such applications, Hyper-redundant manipulators (usually meant as robots with more than 10 DOF [26,27]), result to be more suitable. The main advantages that Hyper-redundant manipulators offer over conventional manipulators are: (i) slender arms, (ii) flexibility in terms of bending, (iii) strong adaptability to confined spaces such as pipes, trusses, and equipment gaps, (iiii) any tools for different purposes can be installed at the end-effector, and (iiiii) the entire manipulator can be used as a tool to complete the winding action [1,26,27]. Furthermore, such manipulators are characterized by low weight, low inertia, and large workspace. In particular, they can achieve positioning in 3-D space by pulling cables through centralized motors in a control box; since the latter is fixed on the base, the weight and inertia of the moving parts are reduced [27]. Finally, Hyper-redundant manipulators allow to increase operations' safety in unstructured environments, due to the flexibility of the driving cable itself [27].

In light of all the cited advantages, Hyper-redundant manipulators are recognized as the best solution for remote inspection and maintenance tasks in narrow spaces, as tokamaks [4–6]. In such special applications, it is necessary to design specifically and appropriately these one-of-a-kind redundant manipulators.

During the complex iterative design process (Fig. 1), virtual simulations are conducted to verify if the concept design is coherent with the ideated strategy, and can be integrated with the rest of Remote Handling (RH) System. At this stage, simulations are necessarily programmed offline, aiming to find (if it exists) at least one feasible path for the manipulator, in compliance with all the specific constraints. Both Forward Kinematic (FK) [23,30,31] or Inverse Kinematic (IK) [13,16,32] approaches may be employed, according to the complexity of the tasks to be simulated. The FK approach may seem more immediate and

* Corresponding author.
*E-mail address:* sara-buonocore@outlook.it (S. Buonocore).

https://doi.org/10.1016/j.fusengdes.2024.114334
Received 30 October 2023; Received in revised form 17 February 2024; Accepted 8 March 2024
Available online 11 March 2024

simpler to be programmed for virtual simulations, as it mainly consists in inserting manually a sequence of joints configurations. Despite this, the greatest limitation is that it cannot guarantee the existence (or not) of at least one feasible trajectory for the manipulator, in compliance with the given constraints [7]. Since it is a "trial and error" process, the obtained results are strongly dependent on the number and type of attempts made, taking the risk of giving “fake alarms” in the Virtual Verification step (Fig. 1). Such “fake alarms” may lead to take significant wrong design decisions within the manipulator’s iterative design, resulting in waste of resources such as time, money, etc. For these reasons, Inverse Kinematic simulations are generally preferrable for manipulators’ offline programming activities. Currently, path planning of hyper redundant manipulators is an important area of research in fusion field, and a significant effort is put into the improvement of the efficiency of the IK algorithm’s optimization criterions [1].

In light of this, we propose a novel IK algorithm for obstacle avoidance path planning that is particularly suitable for hyper-redundant manipulators used for remote maintenance within toroidal machines as nuclear tokamaks. The IK algorithm, if compared to others adopted in this specific application domain, exhibits the following advantages: (i) has the potential to become a design tool to support engineers and researchers during the iterative design process of such complex robotic machines, (ii) offers the possibility to customize cost function and parameters to find the optimal trajectory, (iii) allows to save joint configurations sequence of the optimal trajectory in a log file, as they may be exploited for future offline and online programming of the manipulator’s activities.

The remainder of the paper is structured as follows. Section 2 offers a brief overview of the main IK path planners, while Section 3 addresses the main characteristics of our IK algorithm. Section 4 shows the application of the proposed IK algorithm in the framework of the DTT project [2,3], to find at least one feasible trajectory for the DTT HyRMan, responsible for First Wall modules handling. Finally, conclusions are drawn in Section 5.

## 2. Background

The objective of a trajectory planning algorithm for a redundant manipulator is to find a solution to the IK problem in order to satisfy multiple tasks, taking advantage of the high degrees of redundancy of the robot.

Despite current literature offers plenty of both Cartesian and joint space trajectory planning algorithms, the latter are often preferred for both offline and online programming, since they typically require lower computational burden [1].

In literature, trajectory planning algorithm are mainly distinguished into control-based (whether the selected system is subject to differential constraints) [17,28,29], multilevel-based (whether the selected system is characterized by high-dimensional state space and must be simplified into lower dimensional state spaces) [33–34] and geometric planners. Specifically, planners in the latter category are the best suited to the design process of a robotic manipulator (and its motion strategy) as they account for the geometric and kinematic constraints of the system (e.g., in-vessel components encumbrance for tokamaks), assuming that any feasible path can be turned into a dynamically feasible trajectory. One of the most known geometric planners is the Rapidly exploring Random Trees (RRT) [18]. The algorithm in the basic version, primarily aimed at single-query applications, incrementally builds a tree of feasible paths rooted at the initial condition [12]. Despite its significative advantages (efficient in high-dimensional spaces, adaptable to complex environments, with strong global search capability and probabilistic completeness [11–12]), a consistent limit in its application was found. Specifically, the paths found by sampling algorithms as RRT and Probabilistic Roadmap Method (PRM) classes are not trajectory optimized, as they only have spatial information and cannot be used as input for motion control. In order to address the limitations of the only sampling-based path planning algorithms originally available, geometric optimizing algorithms have been introduced, considering that trajectory optimization is the basis for controlling the movement of the robotic arm, and the quality of the trajectory has an important impact on the completion of the operation [10]. The two main components that characterize the trajectory optimization are the selection of the optimizing motion planner and the quality metrics (or cost function) to be optimized (i.e., path clearance, length, general state cost integral, mechanical work or some arbitrary user-defined optimization criterion). Well-known examples of geometric optimizing planners are PRMStar [19] and RRTStar [21], that are still randomized as PRM and RRT, but they use the remaining planning time to improve the first solution found, thus giving an asymptotic optimality guaranteed [12]. Specifically, RRTStar is an improved version of the original RRT planner, that incrementally builds a tree and improves the path quality by utilizing an optimal distance metric and selecting optimal connections [10]. Such

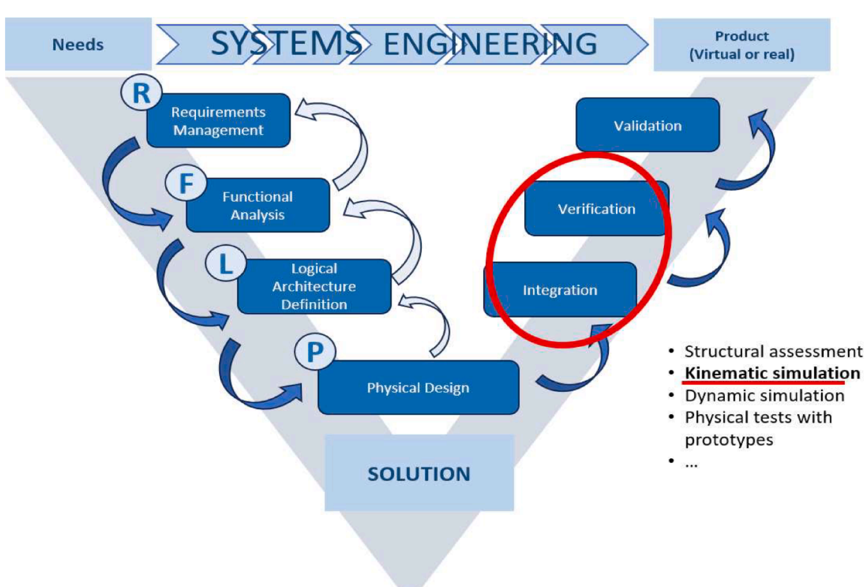



**Fig. 1.** Virtual Verification step of V-model, according to Systems Engineering approach [14].

planner is still widely used because, further to the same advantages of basic RRT planner (efficient in high-dimensional spaces, adaptable to complex environments, with strong global search capability and probabilistic completeness [11–12]), it also converges to an optimal solution, with minimal computational and memory requirements also for complex motion planning problems [11]. Thanks to the combination of node optimization and thinning, RRTStar can select optimal paths between nodes that do not simply extend from a single node, but rather extend from the optimal point by searching among all the nodes of the entire three. In light of these consistent advantages offered by RRTStar planner, in the following sections, we illustrate our IK algorithm based on it and its application to the DTT case study.

## 3. Our IK algorithm

In this section, we describe the main characteristics of our IK algorithm, based on the use Optimal Motion Planning Library (OMPL):

- Is a point-to-point trajectory planning algorithm (Start to Goal).
- Allows for data exchange with CAD models for collision-free trajectory computing.
- Allows for customizing optimization parameters.
- Allows for introducing intermediate waypoints in the path from Start to Goal State (or pose).
- Enables data access for post-simulation analyses.

More details are given in this section for each of the listed features.

For test purposes, it was assumed that the objective of the simulations was to find the shortest path among the possible solutions. For this reason, the selected optimizing planner is RRTStar and cost function in the script has been customized to calculate and compare the lengths of all the feasible trajectories given in output, in order to select the shortest one. However, it is intended that it is possible to easily change the optimizing planner in OMPL and optimization criterion by modifying cost function in the script.

### 3.1. Start and goal states

Based on a geometric planning method, our point-to-point algorithm computes the trajectory from a Start to a Goal State (Target). Precisely, when the simulation is turned on, the manipulator's end-effector follows the same trajectory of the Target in order to make perfectly match its triad with the one centered in the Target (Goal State or pose). Whether the algorithm searches for a single robot configuration that matches the desired pose (Goal State), it may happen that two found end-effector poses result quite identical to each other. For this reason, we hypothesized that if the maximum distance between the two poses is less than 0.001meters), they are considered as one unique pose.

### 3.2. Collision-free trajectory

The computed trajectory form Start to Goal State is collision-free, thanks to the function "sim.getConfingForTipPose". This function calls for the "collisionPairs" parameter to check if there is any obstacle to be avoided. Specifically, whether the "collisionPairs" parameter assumes one of the integer values that are associated to 3D objects (e.g., imported CAD models of VV) that robot must avoid, it detects a possible collision and asks to compute a new trajectory.

The access to CAD models and its usage for collision handling is a crucial aspect for motion planning of a manipulator that shall operate in complex environments as tokamaks, since its motion capabilities are strongly influenced by the presence of several in-vessel components to be avoided, with strict tolerances.

### 3.3. Customizable optimization parameters

As already explained, many well-known geometric optimizing planners use part of the planning time to improve the first solution found, thus giving an asymptotic optimality guaranteed [22]. In light of this, regardless of the selected cost function to be optimized, it is possible to modify two main parameters to manage time and computational burden dedicated to reach the asymptotic optimality. Specifically, the two functions with respective customizable parameters are:

- *"simOMPL.Compute"*: this function is used to compute a solution of the path planning problem. The principal parameter needed by the function is the *"maxTime"*.[1] The greater the *"maxTime"*, the higher is the probability to find a feasible trajectory. If the *"maxTime"* is too small, it's very likely that the path found will not be complaint with the constraint of the scene: the result is that at the end of the *"maxTime"* the robotic arm shall follow an incomplete path. If this happens, this value must be raised. This procedure must be repeated until a proper value of *"maxTime"* is found, which corresponds to find the trajectory. For test purposes, the *"maxTime"* values set for the simulated cases vary from 10 to 10,000 s.
- *"Find Several Collision Free Config And Check Approach"*: this function is used to search for all the possible robot configurations that match the desired pose. The principal parameter needed by the function is the *"trialCnt"*.[2] Due to the randomness of the path procedural generation, intrinsic to the RRTStar algorithm, as the number of attempts increases, the probability of finding an optimal and more simple path increases. Therefore, the higher is the value of *"trialCnt"* parameter, the higher is the number of attempts made by the script to find a collision-free configuration. For test purposes, the *"trialCnt"* values set for the simulated cases vary from 300 to 50,000.

Since we aimed to streamline the Virtual Verification process by providing an easy tool and a quick procedure, such parameters' setting allows to search for optimal collision-free trajectory for a maximum of 8 h. Whether the IK algorithm is not able to find at least one solution, we have considered that system configuration unsuitable in terms of collision avoidance. However, such values are only recommended, but fully customizable at the users' needs.

### 3.4. Waypoints

In line with RRTStar planner and overall OMPL potentialities, we applied the "accumulation method". According to it, the cost of an entire path can be divided into an accumulation of the costs of the smaller motions that make up the path [20]. In light of this, some "waypoints" can be introduced between the Start and the Goal States of the manipulator, in order to cut down the computational time needed for such demanding simulations (e.g., with several objects to avoid or a very long path to be followed). Waypoints can be seen as intermediate poses (or states) introduced to divide a demanding simulation in sub-simulations, each of them characterized by the execution of the path-finding script between two sequent waypoints. In this way, the computational time of the simulation has been significantly reduced, by calculating the cost of the entire path as the sum of each sub-path's costs. Furthermore, this feature can be useful also to those cases in which the manipulator shall be able to conduct a part of the motion with specific requirements and constraints (e.g., Waypoint #2 for DTT case, described in Section 4).

[1] maxTime: "maximum time used for the path searching procedures, in seconds" [22].

[2] trialCnt: is the count (Cnt stands for count) of attempts that the planner makes in the procedural generation from a node to the following.

### 3.5. Post-simulation analyses

As already mentioned, the kinematic simulations in this case are intended for Virtual Verification phase of the manipulator's design process. For this reason, it is highly plausible that an analysis of the results and a review of the simulation is required in a more or less near future. With this regard, we have implemented an interesting feature in our IK algorithm to save the results of the conducted simulation (which in some cases may have reasonably taken some hours to output the optimal trajectory), in order to allow easy "replay" any time, without the need of computing again the whole trajectory.

Furthermore, a secondary script is called by the main one for real-time display of the manipulator joint values. These values are displayed, in the form of textual information, in a secondary window (called "console window") that appears once the optimal trajectory has been found, during its simulation. The frequency with which joint values are shown is easily customizable; we suggest inserting a sample value of 0.1 s to obtain a continuous stream of joint values in the console window. In line with the aim of offering a decision-making tool for post-simulation analyses (as offline or online programming activities of the robot), manipulator's joint values obtained from the optimal trajectory (that are visible during the simulation itself), are also stored in a log file and can be exported as ".pdf" or ".csv" file formats.

## 4. The dtt case study: hyrman

In this section, the application of our IK algorithm to the case study of DTT will be showed. For test purposes, we verified the effectiveness of the proposed IK algorithm by simulating some of the most critical Remote Maintenance tasks for the HyRMan: the Hyper Redundant Manipulator developed for DTT project. Specifically, we have selected the DTT case study, as our IK point-to-point algorithm has resulted to be perfectly suitable for the purposes of the HyRMan kinematic simulations. In fact, in the HyRMan simulation plan, two of the main features required to be tested were the manipulators positioning on predefined points and its motion along predefined paths.

Fig. 2 offers a pictorial view of DTT machine, including the numbering of 5 ports for each sector. According to DTT Remote Handling strategy [8], HyRMan should enter from port#3 (horizontal port in Fig. 2) in the VV to conduct some inspection and maintenance tasks. From previous virtual simulations conducted with an FK approach within DELMIA V5 platform [7], it seemed that the current kinematic chain of the manipulator was not able to conduct some handling tasks of some Inboard First Wall (IFW) (Fig. 3) and Outboard First Wall (OFW) (Fig. 4) modules, as several reachability issues and collisions between the manipulators, the modules and the in-vessel components were encountered. Therefore, in order to obtain reliable results in the Virtual Verification phase (Fig. 1) of the iterative design of the HyRMan, it has been necessary to also adopt an IK approach to confirm (or not) what has

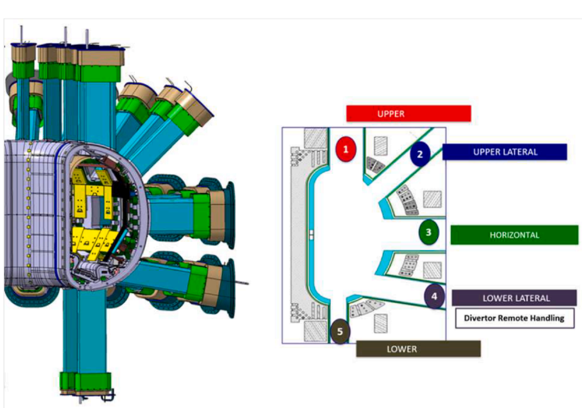



**Fig. 2.** Left: DTT 3D model. Right: Names and numbering of each sector's ports.

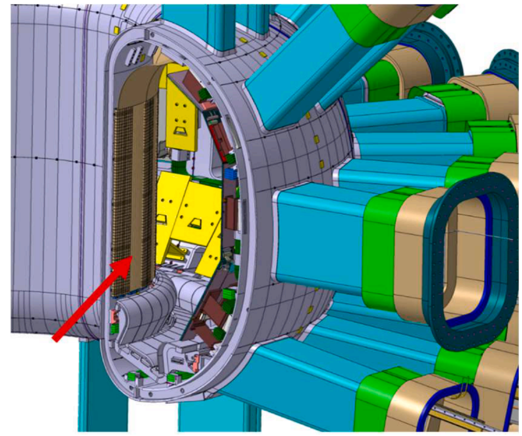


**Fig. 3.** DTT IFW modules.

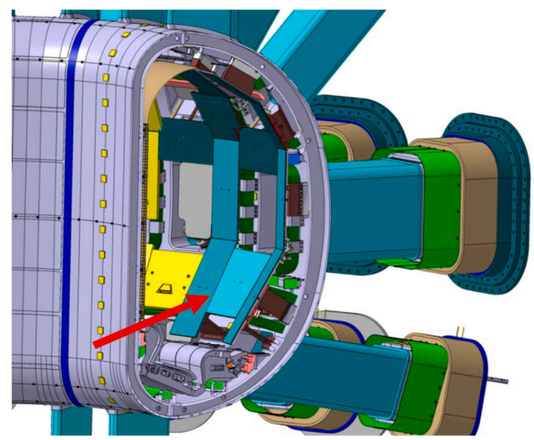


**Fig. 4.** DTT OFW modules.

been found with FK simulations [7].

The DTT HyRMan is composed of two stand-alone kinematic chains [7–9], showed in Fig. 5:

- a *planar arm*, provided with one prismatic joint (for movements along the port duct) and 4 rotative joints for positioning the dexterous arm in the Vacuum Vessel (VV) horizontal plane;

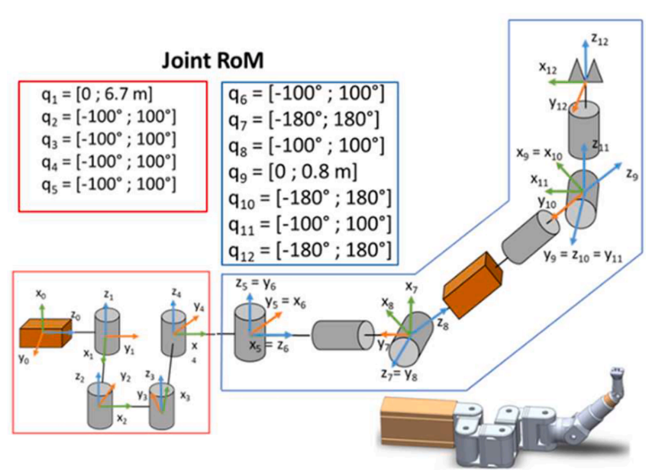



**Fig. 5.** HyRMan's 12-DoF kinematic chain with joints Range of Motion: planar (red) and dexterous (blue) arms.

- a *dexterous arm*, a spatial end-effector manipulator provided with 7 DoFs (6 rotative joints and 1 prismatic) for reaching all the points in the poloidal section.

Before confirming that major modifies to the current kinematic chain of the HyRMan or even a redesign of the RM strategy may be necessary [7], an alternative solution has been proposed. Two different grippers at the manipulator's end-effector have been designed: the Axial (Fig. 6a) and the Ort (Fig. 6b) grippers, whose names take inspiration from the different position of the fingers employed for objects' grasping. Specifically, the Axial gripper's fingers and the Ort gripper's fingers are respectively coaxial and orthogonal to the rotation axis of the HyRMan's last joint (Fig. 7).

Virtual simulations have been conducted to verify if the Axial gripper configuration was compliant with reachability and collision avoidance requisites; otherwise, the Ort gripper alternative configuration has been tested. Due to a lower mechanical complexity of the Axial gripper, it has been designed and considered for virtual testing as the basic configuration; indeed, the Ort gripper configuration has been designed as an extreme alternative to be employed only if the basic one did not allow to reach some OFW modules. However, the mechanical and electrical integration of both the grippers with the HyRMan's end-effector is the same.

With reference to the simulated cases presented in [7], two main critical tasks were identified: IFW-R-1 and OFW-1 modules handling for removal. Since several collisions and reachability issues were encountered, further virtual simulations with an IK approach were necessary to confirm (or not) HyRMan's inability to perform such tasks.

The conducted IK simulations of these two critical cases with Axial and Ort gripper configurations are discussed in this section. At the end of it, Table 1 is provided to summarize what has been done so far in the HyRMan's Virtual Verification, showing the results of the previous FK simulations [7] and the results of the IK simulations performed with our algorithm for the two simulated cases. Furthermore, QR codes are attached to show videos of virtual simulations.

### 4.1. Premises

#### 4.1.1. Selected waypoints

As already mentioned, we applied the "accumulation method" to properly cut-down the computational burden of the IK simulations. Specifically, we divided the entire path from Start to Goal State into some sub-paths by introducing some waypoints, and let the algorithm

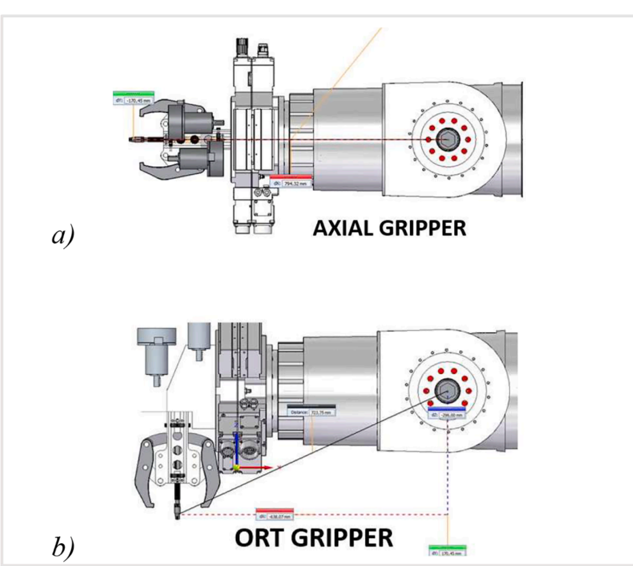



**Fig. 6.** Top view of the Axial (a) and Ort (b) grippers.

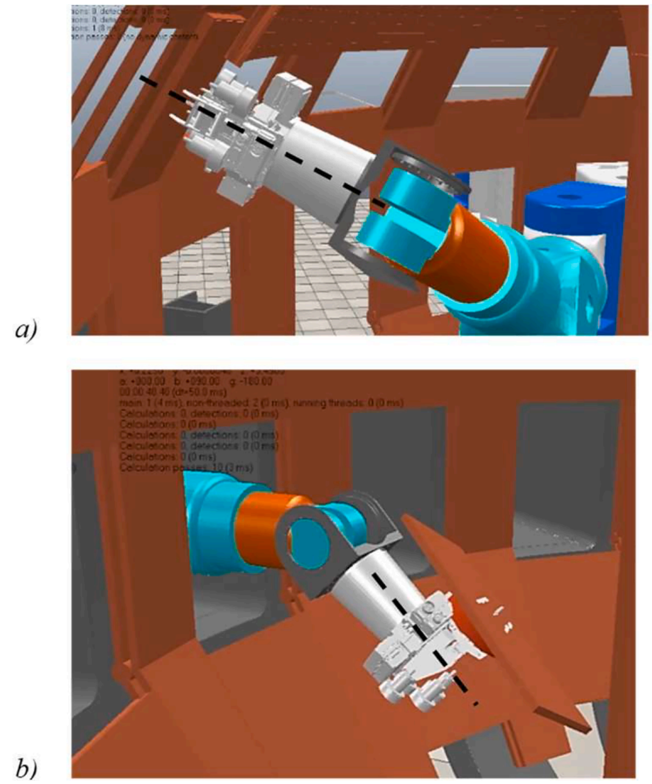



**Fig. 7.** a) The Axial gripper's fingers are coaxial with the rotation axis of the HyRMan's 12th joint (black dotted line); b) the Ort gripper's fingers are orthogonal to the same axis.

calculate the cost function as a sum of each sub-paths' costs functions.

With reference to the HyRMan simulations, for all the RM tasks, two common waypoints have been introduced (Fig. 8), coherently with the two main output specification to be fulfilled:

- Waypoint#1: each task should start from the same "home position", that is Left or Right Deploy position.
- Waypoint#2: the last joint of the HyRMan planar arm shall be positioned on the equatorial line, as it is often required that the manipulator use only planar arm to enter the VV.

For each specific case, other waypoints have been added, including IFW and OFW grabbing interface to simulate modules' approaching.

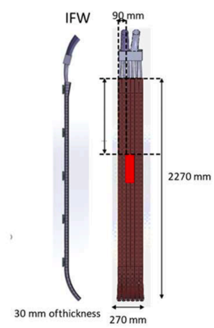
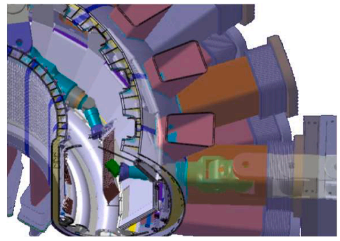
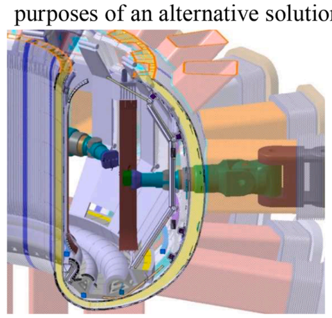
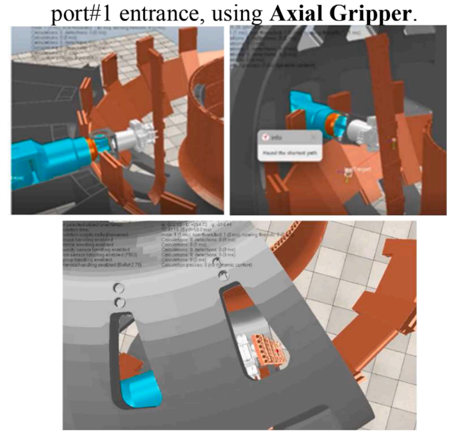
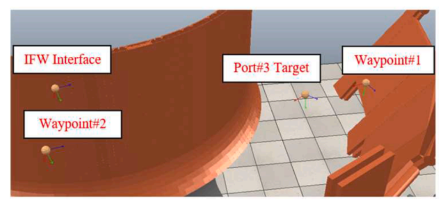
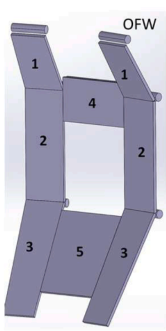
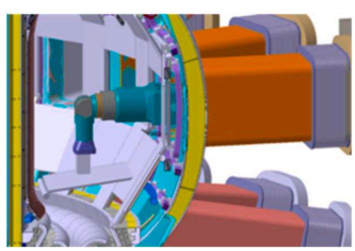
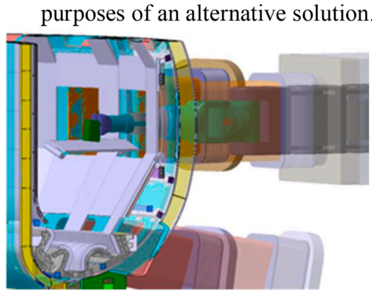
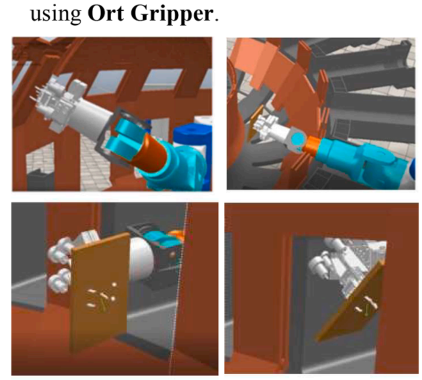
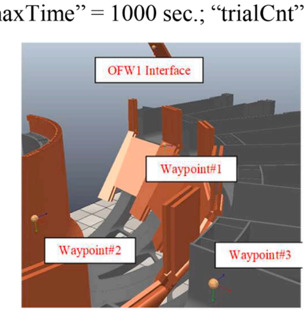
#### 4.1.2. Software platform

Even though DELMIA V5 offers both FK and IK algorithms, the latter can be used with limitations: it does not support redundant kinematic chains, as the HyRMan [7]. For this reason, it has been necessary to choose another platform to perform the IK virtual simulations of the FW handling operations. The selected platform is CoppeliaSim[3]: a robotic simulator, written in LUA programming language, for fast algorithm development, factory automation simulations, fast prototyping and verification, robotics related education, remote monitoring, safety double-checking, as digital twin. Even if CoppeliaSim is just one of the several software compatible with OMPL, it has been selected as its interface ("Sim.OMPL.Algorithm.RRTstar"[4]) enables to access and modify the OMPL functionalities easily via scripting functions. This has

[3] CoppeliaSim 4.3.0 User Manual: https://mde.tw/pjcopsim/content/index.html

[4] OMPL Plugin API reference: https://www.coppeliarobotics.com/helpFiles/en/simOMPL.htm

**Table 1**
Summary of the results of previous FK [7] and current IK simulations on most critical cases for DTT HyRMan.

| Simulated case | FK simulation | IK simulation |
|---|---|---|
| IFW-R-1 module<br><br>Video of the simulations | • **Reachability** issues at port#1 entrance, where the IFW lifting system should take the module.<br><br>• 13 $^{th}$ rotative joint added for test purposes of an alternative solution.<br> | • HyRMan succeeded to take the module at port#1 entrance, using **Axial Gripper**.<br><br>• No modifies to the kinematic chain resulted to be necessary.<br>• Simulation setup: 4 waypoints; “maxTime” = 20 sec.; “trialCnt” = 10000.<br> |
| OFW-1 module<br><br>Video of the simulations | • Several **collisions** between OFW module and in-vessel components.<br><br>• 13 $^{th}$ rotative joint added for test purposes of an alternative solution.<br> | • HyRMan succeeded to handle the module, using **Ort Gripper**.<br><br>• No modifies to the kinematic chain resulted to be necessary.<br>• Simulation setup: 4 waypoints;<br>• “maxTime” = 1000 sec.; “trialCnt” = 50000.<br> |

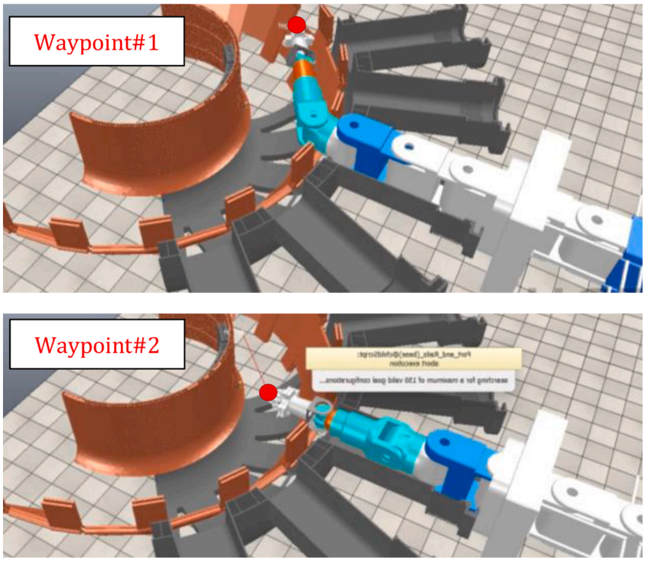



**Fig. 8.** The two reference waypoints for kinematic simulations: HyRMan in Right Deploy position as starting configuration (Waypoint#1) and planar arm in equatorial line (Wapoint#2).

allowed to quickly test various scenarios, without the need to recompile/load test code over and over again.

*4.1.3. Collision detection*

It is acknowledged that calculating collisions with non-convex shapes is exponentially more complex in term of mathematical calculations than convex shapes for a physics engine [24–25]. In fact, in order to lower computational burden required by the physics engine to detect any possible, it is generally recommended to focus only on relevant regions where interference is most likely to occur. In light of this, collision detection in CoppeliaSim was not directly computed on the complex non-convex tokamak environment (including VV and HyRMan), since it was not optimized nor recommended for dynamic collision response calculation. A second invisible layer was added, made of simple geometries called primitive shapes,[5] associated to the respective complex 3D models (Fig. 9). Such shapes were mainly functional, used by physics engine to perform collision detection faster, while complex 3D shapes have been employed for virtual scene rendering.

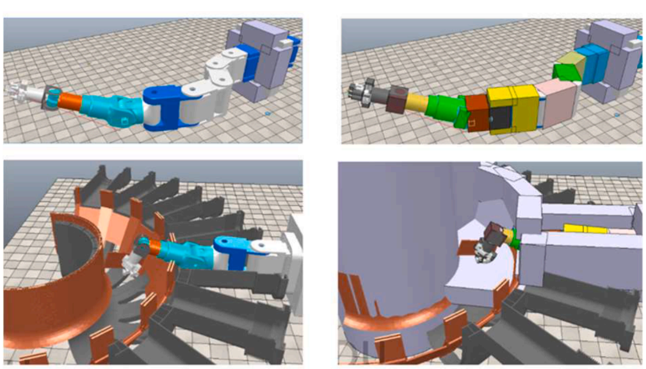


**Fig. 9.** Imported 3D models (left) and respective primitive shapes of HyRMan and VV.

[5] Primitive shapes: https://www.coppeliarobotics.com/helpFiles/en/primitiveShapes.htm

### 4.2. Kinematic simulation of near ifw handling for removal

This simulation aimed to verify the capability of the HyRMan to handle IFW modules that may be close to the manipulator's access port, as well as far from it. For instance, considering that HyRMan enters in the VV from port#3-S1,[6] two simulations shall be performed: the first aims to test the HyRMan's ability to pick and place IFW modules that are close to that access port (e.g., IFW-R-1 module); while the second simulation aims to verify the same ability to handle IFW modules far from the selected access port (e.g., IFW-l-5) [7].

In this section, we describe kinematic simulation of IFW-R-1 module (near IFW), since it resulted to be the most critical. From the FK simulations conducted within DELMIA V5, the IFW-R-1 module seemed to be stuck at the entrance of the equatorial cask: the HyRMan resulted to be not able to properly rotate the IFW module without causing several collisions with in-vessel components [7].

A summary of the results of previous FK and current IK simulations for IFW-1 module handling is given in Table 1. With this regard, we precise that present simulations of the IFW modules do not focus on the IFW modules' handling outside port#1, as the concept design of the system responsible for it (the IFW Lifting System) is still ongoing.

The IK simulations have been carried out with the "Axial Gripper", and the HyRMan in the "Right Deploy Position". In this case, the whole simulation is made of 3 steps:

I. Right Deploy Position (waypoint#1) to waypoint#2
II. waypoint#2 - IFW interface - waypoint#2
III. waypoint#2 – Final target positioned under port# of the third sector.

For each step, the main parameters to input in the path-finding script were set as follows:

- "maxTime" = 20 s.
- "trialCnt" = 10,000.

It resulted that, for IFW-R-1 module, the first usable port#1 to properly place the module and make it available for IFW lifting system, is port#1-S3. This means that, with the current kinematic chain, HyRMan is not able to properly place the IFW module at the entrance of port#1 (upper port) for the 2 adjacent sectors (S1,S2).

This case has been of special interest, as it has an investigative purpose in order to understand the actual possibilities of the designed RH system. In particular, the aforementioned limitations of the FK analysis have partly affected the results obtained, giving a "false alarm". While IK simulations have confirmed that HyRMan is not able to adequately position the IFW-1-module at the entrance of port#1-S1 and port#1-S2, the impossibility resulting from FK simulations for port#1-S3 input has been averted with current IK simulations.

### 4.3. Kinematic simulation of far ofw removal

This simulation aimed to verify the capability of the HyRMan to reach and grasp OFW modules far from the entrance port and move it along the duct to finally accommodate it into the dedicated cask.

Far OFW-1 module removal resulted to be critical from the FK simulations conducted within DELMIA V5, as several collisions between OFW-1 module and in-vessel components were encountered [7]. A summary of the results of previous FK and current IK simulations for IFW-1 module handling is given in Table 1.

In this case, for IK simulation, 3 waypoints have been introduced, further to OFW-1 module grabbing interface: Right Deploy position as waypoint#1, waypoint#2 to position the dexterous arm in the Vacuum

[6] Port#3-S1: "S" stands for "Sector". DTT has 18 Sectors (Fig. 2) [3].

Vessel horizontal plane and waypoint#3 as the final position to take the OFW-1 module inside the cask. The IK virtual simulation was therefore made of 4 steps:

I. From Waypoint#1 to OFW1 interface
II. From OFW1 interface to waypoint#1
III. From waypoint#1 to waypoint#2
IV. From waypoint#2 to waypoint#3.

First simulations were conducted with Axial Gripper and the following parameters:

- "maxTime" = 1000 s.
- "trialCnt" = 50,000.

Despite virtual simulations with Axial Gripper gave negative results (as HyRMan seemed to be not able to take OFW-1 module inside the cask), an optimal collision-free trajectory for the manipulator with the adoption of the Ort Gripper has been found, with the same values.

Compared to the IFW-R-1 simulation, we can certainly say that the remarkable overall encumbrance of the OFW-1 module required a much greater computational effort in calculating a collision-free trajectory. In fact, whether the algorithm already found numerous possible trajectories with the Axial Gripper and gave the optimal in output in less than one hour (OFW-R-1 simulation), in this case the Axial Gripper resulted to be unsuitable, and it took 8 h of calculation to get the first optimal trajectory output with Ort Gripper.

## 5. Conclusion

In this work, we presented our IK algorithm for obstacle avoidance path planning to simulate Remote Maintenance tasks conducted by hyper redundant manipulators in complex narrow environments, as tokamaks. Our IK algorithm is based on the use of a geometric optimizing planner and a customizable cost function, with customizable parameters in order to output the optimal trajectory for the hyper-redundant manipulator. Furthermore, our IK algorithm allows for data exchange with CAD models for collision-free trajectory computing and enables data access for post-simulation analyses. Beyond the application presented in this paper, the present IK algorithm based on OMPL is compatible with the main virtual environments and programming languages [15], demonstrating its flexibility of use.

The novelty of our work consists in proposing the use of an IK algorithm as a design tool to support engineers and researchers during the iterative design process of such complex robotic machines. For this reason, we applied our IK algorithm to DTT case study, simulating most critical RM tasks for HyRMan. As expected, the application of our IK algorithm has allowed us to discover some "fake alarms" in the Virtual Verification step of the HyRMan's design process, currently in progress. The achievement of an asymptotically guaranteed optimal trajectory through our IK algorithm allowed us to verify the actual feasibility of RM tasks previously considered impossible for the current HyRMan kinematic chain. This has shown that the quality and reliability of the employed tools may greatly affect the outcome of the results and the decisions taken on the design of such first-of-a-kind robots.

Future improvement of our IK algorithm are already planned, as we aim to introduce more than one quality metric for multi-objective optimization (i.e., minimum joint torque, shortest time, etc.).

## CRediT authorship contribution statement

**Sara Buonocore:** Writing – original draft, Software, Methodology, Data curation. **Andrea Zoppoli:** Investigation, Conceptualization. **Giuseppe Di Gironimo:** Writing – review & editing, Supervision, Project administration.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgements

This work has been carried out in the frame of the DTT activity https://www.dtt-project.enea.it/. The authors are grateful to EniProgetti (Venezia, Italy) for the significant contribution given to the mechanical design of HyRMan and the commitment to provide support for the conducted kinematic simulations. Authors are grateful to Andrea Reale and Gioacchino Micciché for helpful discussions on DTT Remote Handling System. Authors will be happy to provide more information on this work on request.

## References

[1] G.S Chirikjian, Conformational modeling of continuum structures in robotics and structural biology: a review, Adv. Robotics Int. J. Robot. Soc. Japan 29 (13) (2014) 817.

[2] C. Lauretti, S. Grazioso, E. De Marchi, T. Grasso, D. Indrigo, M. Favaretto, D. Marzullo, G. Micciché, M. Parisi, A. Reale, G. Di Gironimo, Remote handling system for the DTT Fusion reactor: a SE approach for preliminary conceptual design of the main robotic equipment, OMC Med Energy Conference and Exhibition (2021).

[3] R. Martone, R. Albanese, F. Crisanti, A. Pizzuto, P. Martin, DTT - Divertor Tokamak Test facility: a testbed for DEMO, Fusion Eng. Des. 167 (2019) 112330, https://doi.org/10.1016/j.fusengdes.2021.112330.

[4] Tang Chaoquan, Ma Shugen, Li Bin, et al., Self bionic control strategy for snake like robot environment based on neural stepping incentive mechanism, J. Mech. Eng. 49 (1) (2013) 53–62.

[5] A.C. Lai, P. Loreti, P. Vellucci, A fibonacci control system with application to Hyper-redundant manipulators, Mathematics of Control Signals and Systems 28 (2) (2016) 1–32.

[6] A. Mohammadi, E. Rezapour, M. Maggiore, et al., Maneuvering control of planar snake robots using visual holonomic constraints, IEEE Trans. Control Syst. Technol. 24 (3) (2016) 884–899.

[7] S. Buonocore, G. Di Gironimo, M. Favaretto, T. Grasso, F. Zanon, S. Grazioso, Systems engineering approach for the iterative concept design and virtual simulation of the DTT hyper redundant manipulator, Fusion Eng. Des. 190 (2023) 113534, https://doi.org/10.1016/j.fusengdes.2023.113534. ISSN 0920-3796.

[8] G. Di Gironimo, S. Grazioso, The DTT device: preliminary remote maintenance strategy, Fusion Eng. Des. (2021) 172, https://doi.org/10.1016/j.fusengdes.2021.112762.

[9] C. Lauretti, T. Grasso, E. de Marchi, S. Grazioso, G. Di Gironimo, A geometric approach to inverse kinematics of hyper-redundant manipulators for tokamaks maintenance, Mech. Mach. Theory 176 (2022) 104967, https://doi.org/10.1016/j.mechmachtheory.2022.104967. ISSN 0094-114X.

[10] Zesheng Wang, Yanbiao Li, Peng Sun, Yiqin Luo, Bo Chen, Wentao Zhu, A multi-objective approach for the trajectory planning of a 7-DOF serial-parallel hybrid humanoid arm, Mech. Mach. Theory 165 (2021) 104423, https://doi.org/10.1016/j.mechmachtheory.2021.104423. ISSN 0094-114X.

[11] H. Long, G. Li, F. Zhou, T. Chen, Cooperative dynamic motion planning for dual manipulator arms based on RRT*Smart-AD Algorithm, Sensors 23 (2023) 7759, https://doi.org/10.3390/s23187759.

[12] Sertac Karaman, Emilio Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Robot. Res. - IJRR 30 (2011) 846–894, https://doi.org/10.1177/0278364911406761.

[13] L. Sciavicco, B. Siciliano, A dynamic solution to the inverse kinematic problem for redundant manipulators, in: Proceedings. 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, USA, 1987, pp. 1081–1087, https://doi.org/10.1109/ROBOT.1987.1087921.

[14] E. Brusa, A. Calà, D. Ferretto, Systems Engineering and Its Application to Industrial Product Development, Springer International Publishing, Cham, 2018, https://doi.org/10.1007/978-3-319-71837-8.

[15] "Remote A.P.I.": https://www.coppeliarobotics.com/helpFiles/en/remoteApiOverview.htm.

[16] A. Reiter, A. Müller, H. Gattringer, On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators, IEEE T. Inform. 14 (4) (2018) 1681–1690.

[17] T. Fraichard, Dynamic trajectory planning with dynamic constraints: a 'state-time space' approach, in: Proceedings of 1993 IEEE/RSJ International Conference on

Intelligent Robots and Systems (IROS '93), Yokohama, Japan 2, 1993, pp. 1393–1400, https://doi.org/10.1109/IROS.1993.583794.
[18] D.H. Kim, S.J. Lim, D.H. Lee, J.Y. Lee, C.S. Han, A RRT-based motion planning of dual-arm robot for (Dis) assembly tasks, in: Proceedings of the 2013 44th International Symposium on Robotics (ISR), Seoul, Republic of Korea, 2013, pp. 1–6, 24–26 October.
[19] P. Leven, S Hutchinson, Toward real-time path planning in changing environments, in: B.R. Donald, K. Lynch, D. Rus (Eds.), Algorithmic and Computational robotics: New directions 2000 WAFR, Natick, MA: A K Peters, 2001, pp. 363–376.
[20] "Optimal Planning", https://ompl.kavrakilab.org/optimalPlanning.html.
[21] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Rob. Res. 30 (2011) 846–894.
[22] D. Connell, H.M. La, Dynamic path planning and replanning for mobile robots using RRT, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 2017, pp. 1429–1434, https://doi.org/10.1109/SMC.2017.8122814.
[23] B. Siciliano, Kinematic control of redundant robot manipulators: a tutorial, J Intell Robot Syst 3 (1990) 201–212, https://doi.org/10.1007/BF00126069.
[24] Arne Roennau, Sutter, F., Heppner, G., Oberlaender, J. and Dillmann, R. (2013). Evaluation of physics engines for robotic simulations with a special focus on the dynamics of walking robots. https://doi.org/10.1109/icar.2013.6766527.
[25] P. Jiménez, F.T. Thomas, C. Torras, Collision Detection Algorithms For Motion Planning, Springer eBooks, 1998, pp. 305–343, https://doi.org/10.1007/bfb0036075.
[26] Z. Mu, et al., Hyper-Redundant Manipulators for Operations in Confined Space: typical Applications, Key Technologies, and Grand Challenges, in: IEEE Transactions on Aerospace and Electronic Systems 58, 2022, pp. 4928–4937, https://doi.org/10.1109/TAES.2022.3217746. Dec.
[27] Z. -c. Du, G.-Y. Ouyang, J. Xue, Y. -b. Yao, A Review on Kinematic, Workspace, Trajectory Planning and Path Planning of Hyper-Redundant manipulators, in: 2020 10th Institute of Electrical and Electronics Engineers International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Xi'an, China, 2020, pp. 444–449, https://doi.org/10.1109/CYBER50695.2020.9279171.
[28] E. Szczerbicki, L. Janson, M. Pavone, Optimal Sampling-Based Motion Planning Under Differential constraints: The driftless Case, Europe PMC (PubMed Central), 2015, https://doi.org/10.1109/icra.2015.7139514.
[29] Steven Michael Lavalle, Planning Algorithms, Cambridge University Press, Cop, New York (Ny), 2014.
[30] A. Melingui, R. Merzouki, J.B. Mbede, C. Escande, B. Daachi, N. Benoudjit, Qualitative approach for inverse kinematic modeling of a Compact Bionic Handling Assistant trunk, in: 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 2014, pp. 754–761, https://doi.org/10.1109/IJCNN.2014.6889947.
[31] B.A. Jones, I.D. Walker, Practical Kinematics for Real-Time Implementation of Continuum Robots, IEEE Trans. Rob. 22 (6) (2006) 1087–1099, https://doi.org/10.1109/TRO.2006.886268. Dec.
[32] B. Karlik, S. Aydin, An improved approach to the solution of inverse kinematics problems for robot manipulators, Eng. Appl. Artif. Intell. 13 (2) (2000) 159–164, https://doi.org/10.1016/s0952-1976(99)00050-0.
[33] A. Orthey, M. Toussaint, Rapidly-Exploring Quotient-Space Trees: Motion Planning Using Sequential Simplifications. *Arxiv*, Cornell University, 2019, https://doi.org/10.48550/arxiv.1906.01350.
[34] A. Orthey, S. Akbar, M. Toussaint, Multilevel Motion Planning: A Fiber Bundle Formulation. *Arxiv*, Cornell University, 2020, https://doi.org/10.48550/arxiv.2007.09435.