Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 2, February : 2023

Path Planning and Trajectory Optimization for Agile and Dexterous Robotic Manipulators

[1] Dr. R Lokanadham, [2]A Daniel Praneeth, [3]PVV Srinivas Rao

[1]Professor, Department of Mechanical Engineering, Narsimha Reddy Engineering College, Secunderabad, Telangana
[2, 3] Assistant Professor, Department of Mechanical Engineering, Narsimha Reddy Engineering College, Secunderabad, Telangana

Abstract

This research delves into the critical domain of path planning and trajectory optimization to enhance the agility and dexterity of robotic manipulators. Agile and dexterous robotic systems hold immense potential for a wide array of applications, from industrial automation to intricate surgical procedures. The key challenges lie in enabling these robots to navigate complex environments with obstacles and execute tasks with speed and precision. This study addresses these challenges through a multifaceted approach. The research begins with the development of accurate kinematic and dynamic models for robotic manipulators, considering joint configurations, link lengths, and payload dynamics. Environmental perception plays a pivotal role, requiring the implementation of robust sensors and advanced perception algorithms, including computer vision and lidar, to facilitate real-time mapping of the surroundings. A significant focus is placed on the design and implementation of path planning algorithms that can generate optimal trajectories, accounting for the kinematic constraints of the robotic manipulator and the intricacies of the environment. The utilization of rapidly exploring random trees (RRT), A* algorithms, and machine learning-based approaches contributes to adaptive planning strategies. Through this comprehensive investigation, the research aims to provide a foundation for the advancement of agile and dexterous robotic systems, offering insights into the development of cutting-edge algorithms and methodologies. The outcomes of this study hold the potential to revolutionize the capabilities of robotic manipulators, unlocking new possibilities for efficient and precise operations in dynamic and challenging environments.

1. Introduction

In recent years, the field of robotics has undergone a paradigm shift with an increasing emphasis on the development of agile and dexterous robotic manipulators. These advanced robotic systems are characterized by their ability to navigate complex environments, avoid obstacles, and execute intricate tasks with a level of adaptability and precision previously unseen. This shift is driven by the growing demand for robotics in diverse applications, including industrial automation, healthcare, search and rescue operations, and beyond.

The agility and dexterity of a robotic manipulator are pivotal in determining its effectiveness across various domains. An agile robotic system can swiftly respond to dynamic changes in its environment, enabling it to perform tasks efficiently and with reduced time latency. On the other hand, dexterity, characterized by the ability to manipulate objects with precision, is crucial for applications such as delicate surgical procedures or tasks in unstructured environments.One of the primary challenges in realizing the full potential of agile and dexterous robotic manipulators lies in developing sophisticated path planning and trajectory optimization techniques. These techniques play a foundational role in orchestrating the motion of robotic manipulators, ensuring not only efficient navigation but also the ability to perform complex tasks with the required accuracy.

The complexity of real-world environments, coupled with the intricacies of robotic kinematics and dynamics, demands advanced solutions for path planning. Traditional methods often fall short in addressing the agility and dexterity requirements of modern robotic systems. Hence, there is an imperative need to explore and develop novel algorithms and methodologies that can unlock the full potential of agile and dexterous robotic manipulators. This research seeks to contribute to the evolving landscape of robotics by investigating and advancing the state-of-the-art in path planning and trajectory optimization for agile and dexterous robotic manipulators. By doing so, we aim to overcome the existing limitations, provide new insights, and pave the way for the practical implementation of highly efficient and adaptable robotic systems.

1.1 Motivation

The motivation behind this research stems from the transformative impact that agile and dexterous robotic manipulators can have across various industries. In industrial settings, the ability to swiftly and precisely maneuver robotic arms can significantly enhance manufacturing processes, leading to increased productivity and flexibility. In the healthcare sector, agile robotic systems can revolutionize surgical procedures, making them less invasive and more precise. Moreover, in scenarios such as disaster response and search and rescue operations, dexterous robotic manipulators can navigate complex terrains and manipulate objects to aid in critical tasks. Despite these potentials, the realization of truly agile and dexterous robotic systems requires a fundamental understanding and advancement of their path planning and trajectory optimization capabilities. The motivation to address this challenge lies in the broader goal of harnessing the full potential of robotics to make significant contributions to human well-being, industrial efficiency, and the overall advancement of technology.

1.2 Scope of the Research

This research has a broad scope that encompasses both theoretical developments and practical implementations in the realm of path planning and trajectory optimization for agile and dexterous robotic manipulators. The focus is not only on devising algorithms that can generate optimal paths but also on considering real-world constraints and uncertainties that are inherent in complex environments. The investigation extends to the integration of perception systems that allow robotic manipulators to interpret and adapt to their surroundings dynamically. This involves the utilization of advanced sensors, computer vision techniques, and environmental mapping algorithms to provide the robotic systems with a comprehensive understanding of their operational context. Furthermore, the research explores the interplay between path planning algorithms and the kinematic and dynamic models of robotic manipulators. The goal is to develop methodologies that optimize trajectories while accounting for the specific capabilities and limitations of the robotic hardware.

In terms of applications, the research addresses a diverse range of scenarios, including industrial automation, healthcare, and disaster response. The intention is to provide insights and solutions that are not only theoretically sound but also practically applicable across different domains.

2. Literature review

Classical motion planning Hart et al. (1968) developed one of the first search algorithms in the context This is the well-known A* algorithm for robot motion planning. This fundamental work was further improved by Lozano-Perez (1987), who converted the configuration-space obstacles of a manipulator with n degrees of freedom (DoF) into a collection of slices of (n −1) dimensions. By using the A* algorithm, the manipulator was able to plan the route for each slice, so constructing a path that was free of collisions. This methodology is computationally demanding and may not be useful for planning operations on an industrial scale since it searches the whole search space in order to locate the nodes that will result in a route that has the lowest projected cost. A time-efficient form of the A* method was presented by Guruji et al. (2016). This variant examines all nodes while computing the fitness values only during the collision phase. Thus, incorrect solutions may be eliminated from consideration. The aforementioned approaches are computationally inefficient by virtue of the sampling-based approach that is used in the modeling and pre-processing of the dual-arm procedures. When it comes to dealing with real-world complicated problems that include physical limits and environmental impediments, these techniques are challenging to put into effect. Randomized heuristics may be used to handle real-world operational issues in a more flexible manner and to find (near-) optimal solutions to the large-scale industrial problems (Juan et al., 2013). This is a technique that can be implemented to remedy this limitation.

These approaches are adaptable, allowing for the use of collision detection systems, which helps to avoid doing an extensive search of various pathways in order to discover the optimal answer. [Mac et al., 2016] found that heuristics are especially helpful in the context of robot motion planning since they speed up the planning phase and help discover the shortest route. Elbanhawi and Simic (2014) have recently acknowledged the efficacy of randomized motion planning approaches in resolving intractable issues. This has led to the methods' recent acknowledgment. A good example of a randomized motion planning approach is the PRM (Kavraki et al., 1996), as well as the RRT (LaValle, 1998). A graph-based search technique known as the PRM method discretizes continuous spaces and uses a heuristic to discover the (near-) optimum route (Boor et al., 1999). This approach was developed by the PRM method. Through the use of PRM-based methods for the motion planning of single-arm robots, a number of research were inspired by this important work. The Lazy Probabilistic Road Map (LPRM; Bohlin&Kavraki, 2000) makes it possible to decrease the amount of time spent planning by removing random nodes and edges from the roadmap in the event that they collide with an obstacle. This helps to limit the number of collision checks and streamline the planning process. Instead of connecting a predetermined number of nodes, the enhanced PRM that Karaman and Frazzoli (2011) created continuously raises the number of connection tries as the roadmap evolves. This allows for the possibility of obtaining a route that converges to the optimum path. An expanded PRM was created by Rodríguez et al. (2014). This PRM is capable of classifying detachable items by taking into account the obstacles that they can possibly collide with. Additionally, it looks for new pathways by detecting the barriers that need to be eliminated in order to make the path legitimate. Methods that are based on PRM take samples from a very small number of random nodes, which is sufficient to cover a significant portion of the space that is possible for establishing the optimal route in problems of minimal size. Therefore, if inadequate sample nodes are taken into consideration or if their distribution is inappropriate, the chance of finding the optimum route in the complicated and large-scale operational settings drops.

RRT (LaValle, 1998) is a structured method that operates based on the incremental expansion of a random tree from the sample nodes with a bias towards the mainly unknown portions of the search space. It is a notable alternative to PRMs for motion planning. The RRT was developed by LaValle in 1998. According to Devaurs et al. (2014), Kala (2013), and Noreen et al. (2018), RRTs are more successful than PRMs because they are especially well-suited for exploring nonconvex high-dimensional search spaces and for dealing with difficult issues that are severely restricted. The inversion of kinematics (Vahrenkamp et al., 2009), conducted by K.-C. Ying and colleagues, was published in Computers & Industrial Engineering 160 (2021) 107603. Early examples of the uses of

RRTs include solving plans for three nonholonomic and Kino-dynamic planning issues in autonomous robotic motion planning (Elbanhawi&Simic, 2014). For the purpose of enhancing the effectiveness of the RRTs, heuristics that use biased sampling have been presented. Depth-First-Search (DFS) is a concept that is used by the RRTConnect that was established by Kuffner and LaValle (2000). This idea extends a node to its closest neighbors in a continuous manner until an impediment is overcome. An independent random tree incorporates knowledge from earlier trees to boost its development, hence enhancing the quality of the route that is produced as a consequence of the Everytime Random Tree, which was proposed by Ferguson and Stentz (2006). Through the process of gradually rewiring the connections until the shortest route is established, the enhanced RRT that Karaman and Frazzoli (2011) created ensures that the asymptotic optimality is accomplished. By splitting the jobs into assembly and regrasping pathways, Kim et al. (2013) created an RRT-based planner for the purpose of creating assembly paths. After beginning with a graph search for the purpose of initializing a viable route in a low-dimensional space, the A*-RRT, which was created by Brunner et al. (2013), is a two-phase technique that continues with the application of the RRT algorithm to search in the high-dimensional space. This model is another prominent example. These algorithms, which are based on RRT, demonstrated a great track record of search capacity when it came to tackling the motion planning challenges that the robot was experiencing.

When compared to the other algorithms that are now available, RRTs are especially successful when it comes to planning in high-dimensional and complicated motion patterns. In addition to this, they are adaptable and may be modified to correspond with the particular operating circumstances of each individual instance. The RRTs that are now in use have a limitation in that they investigate the configuration space (also known as C-space) by means of random node sampling. This approach leads to sluggish convergence or erroneous pathways. On top of that, these techniques are notoriously difficult to generalize and often rely on hand engineering in order to adapt to the new environment. We would like to direct readers who are interested in a full study of the classical motion planning techniques to the review that was written by Wahab et al. with the year 2020. Biased sampling and the curse of dimensionality in the motion planning optimizer continue to be the most significant problems (Qureshi et al., 2021). Motion planning methods should be effective and computationally efficient in order to bridge the gap between theory and practice and enable robots to take over more complex tasks, particularly in labor-intensive activities such as assembly. Other problems include biassed sampling and the curse of dimensionality. 2.2. Motion planning that is formed via learning In contrast to the relatively recent machine learning applications in this area, the literature on classical motion planning has been around for quite some time. In recent years, researchers have been investigating contemporary methods such as learning (Berenson et al., 2012) and neural network-based methods (Bency et al., 2019; Duguleana&Mogan, 2016; Qureshi & Yip, 2018) with the goal of enhancing the efficiency of approximation methods or accelerating the planning process in complex operational scenarios. A learning-based strategy that promotes hand-eye synchronization for single-arm robot grasping was investigated by Levine et al. (2018), who drew upon the current contemporary motion planning literature. In 2017, Yahya and colleagues came up with an innovative method for robotic door opening jobs that was based on a distributed policy search.

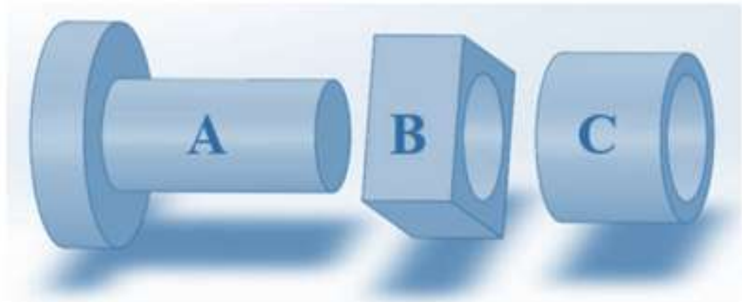
For the purpose of dexterous handling of single-arm assembly robots, Popov et al. (2017) suggested a deep reinforcement learning algorithm. The use of deep reinforcement learning was further extended for the purpose of task planning for single-arm robots operating in three-dimensional settings (Ichter et al., 2018). When Ji et al. (2019) were looking for a collision-free route for single-arm robotic operations, they devised a different reinforcement learning-based technique that they called Q-learning. Neural networks were included into the learning-based planner that Qureshi et al. (2021)

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 2, February : 2023

developed in order to improve the search technique for finding the (near-) optimal route. This planner was designed to solve motion planning issues for generic robots. Using a learning-based neural planner, Watt and Yoshiyasu (2020) were able to construct paths in two-dimensional settings while avoiding obstacles. This occurred more recently. An method for broad robotic tasks that is based on soft actor–critic planning was developed by Prianto et al. (2020). There are not many research that are relevant to the planning of collision-free trajectories in this context. Path planning of dual-arm assembly robots in 3D settings is substantially more complicated and computationally more comprehensive than the planning of single-arm operating environments. Our motivation to design a computationally efficient deep learning-based optimization technique for the motion planning of dual-arm assembly robots in both two-dimensional and three-dimensional settings came from the fact that this gap, combined with the necessity for real-time planning methods, motivated us the most.

3. Proposed Mechanism

This is the well-known A* algorithm, which is used for planning the mobility of robots. Lozano-Perez (1987) made a significant advancement in this foundational work by transforming the configuration-space barriers of a manipulator with n degrees of freedom (DoF) into a collection of slices with dimensions of (n −1), so enhancing the overall quality of the work. The A* method allowed the manipulator to design the route for each slice, therefore generating a path that was devoid of any collisions. This was accomplished by the manipulator. Because it examines the whole search space in order to discover the nodes that would result in a route that has the lowest predicted cost, this approach is computationally expensive and may not be suitable for planning operations on an industrial scale. This is because it explores the entire search universe. The A* approach was provided by Guruji et al. (2016) in a version that was more efficient with regard to time. This form only computes the fitness values during the collision phase, but it does an examination of all nodes on the network. Therefore, it is possible to exclude from consideration any solutions that are erroneous. As a result of the sampling-based method that is used in the modeling and pre-processing of the dual-arm operations, the techniques that have been discussed so far are computationally inefficient. When it comes to dealing with complex issues that occur in the real world, such as those that include physical limitations and environmental obstacles, it is difficult to put these strategies into practice. It is possible to utilize randomized heuristics to address real-world operational concerns in a more flexible way and to discover (near-) optimum solutions to large-scale industrial problems (Juan et al., 2013). Randomized heuristics may be used to solve these issues. This is a method that may be used to get the desired result of overcoming this barrier.



$$OS = \begin{array}{c c} & \begin{array}{ccc} C & B & A \end{array} \\ \begin{array}{c} S_1 \\ S_2 \\ S_3 \end{array} & \begin{bmatrix} 0 & 0 & 2 \\ 0 & 1 & 2 \\ 1 & 2 & 2 \end{bmatrix} \end{array}$$

Fig. 1. Operating and supporting matrix representation of an illustrative example.

The adaptability of these techniques makes it possible to make use of collision detection systems, which helps to prevent the need to conduct a comprehensive search of a variety of paths in order to get the best response. The research conducted by Mac et al. (2016) discovered that heuristics

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 2, February : 2023

are particularly useful in the context of robot motion planning. This is due to the fact that they expedite the planning process and assist in finding the shortest path. Recent research conducted by Elbanhawi and Simic (2014) has shown that randomized motion planning strategies are effective in addressing problems that had been known to be intractable. The methodologies have just recently been acknowledged as a result of this. Both the PRM (Kavraki et al., 1996) and the RRT (LaValle, 1998) are excellent examples of randomized motion planning approaches. Both of these papers were published in 1996. According to Boor et al. (1999), the PRM method is a graph-based search approach that discretizes continuous spaces and employs a heuristic in order to find the (near-) optimal path. Through the use of the PRM technique, this strategy was created. This significant study served as a source of inspiration for a number of research projects, which used PRM-based algorithms for the motion planning of single-arm robots among other applications. By deleting random nodes and edges from the roadmap in the event that they collide with an obstacle, the Lazy Probabilistic Road Map (LPRM; Bohlin&Kavraki, 2000) makes it feasible to reduce the amount of time spent planning. This is accomplished by removing the nodes and edges from the roadmap. This helps to reduce the amount of collision tests that are performed and further streamlines the planning process. The upgraded PRM that Karaman and Frazzoli (2011) developed will not link a predefined number of nodes; rather, it will continually increase the number of connection attempts as the roadmap develops. The prospect of having a path that converges to the best possible path is made possible when this occurs. Rodríguez et al. (2014) developed a PRM that was extended via their work. This particular PRM is able to identify detachable things by taking into consideration the potential impediments that they may come into contact with. Additionally, it searches for new paths by identifying the obstacles that must be removed in order to make the path legitimate. This is done in order to find new directions. This is sufficient to cover a considerable amount of the space that is available for creating the optimum route in problems of minimum size. Methods that are based on PRM take samples from a relatively small number of random nodes, which is adequate to cover the space. Therefore, if insufficient sample nodes are taken into account or if their distribution is improper, the possibility of discovering the best route in the intricate and large-scale operational settings lowers.

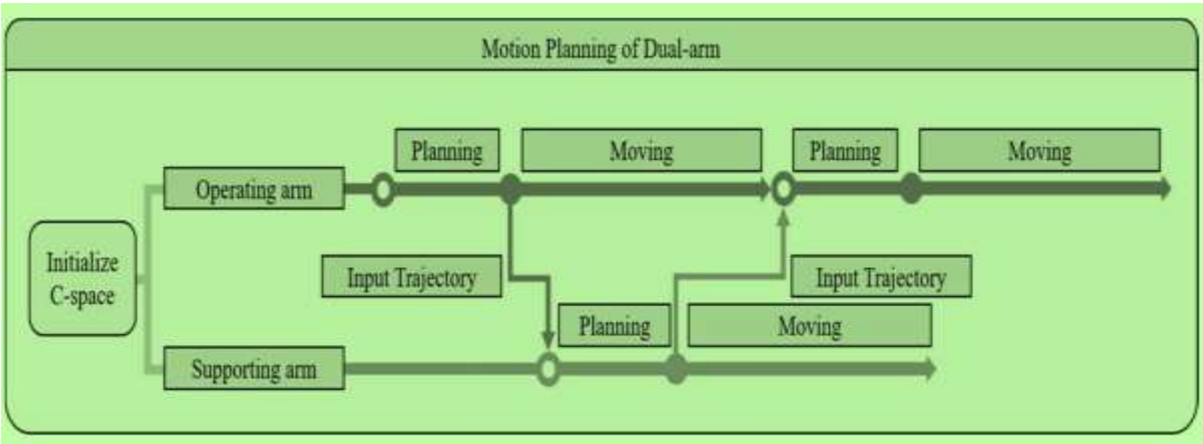


Fig. 2. The dual-arm motion planning schedule in LSTM-BiRRT.

The Random Tree Technique (RRT) is a structured approach that works based on the incremental development of a random tree from the sample nodes with a bias towards the mostly unknown sections of the search space. This method was developed by LaValle in 1998. When it comes to motion planning, it comes as a noteworthy alternative to PRMs. It was in the year 1998 when LaValle came up with the RRT. This is because RRTs are particularly well-suited for exploring nonconvex

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 2, February : 2023

high-dimensional search spaces and for dealing with challenging challenges that are highly limited, as stated by Devaurs et al. (2014), Kala (2013), and Noreen et al. (2018). This is the reason why RRTs are more successful than PRMs. Inversion of kinematics (Vahrenkamp et al., 2009) was a study that was carried out by K.-C. Ying and colleagues, and it was published in Computers & Industrial Engineering 160 (2021) 107603. According to Elbanhawi and Simic (2014), one of the first instances of the applications of RRTs is the resolution of plans for three nonholonomic and Kino-dynamic planning difficulties in the context of autonomous robotic motion planning. The presentation of heuristics that make use of biased sampling has been made with the intention of improving the efficiency of the RRTs. Kuffner and LaValle (2000) developed the RRTConnect, which makes use of the Depth-First-Search (DFS) idea. This concept was established by the RRTConnect. Through the implementation of this concept, a node is extended to its immediate neighbors in a continual way until an obstacle is overcome. Through the process of gradually rewiring the connections until the shortest route is established, the enhanced RRT that Karaman and Frazzoli (2011) created ensures that the asymptotic optimality is achieved. This is accomplished by incorporating knowledge from earlier trees into an independent random tree in order to boost its development. This allows for an improvement in the quality of the route that is produced as a result of the Everytime Random Tree, which was proposed by Ferguson and Stentz (2006). Kim et al. (2013) developed an RRT-based planner for the goal of facilitating the creation of assembly routes. This planner was produced by dividing the tasks into assembly and regrasping pathways. After commencing with a graph search for the aim of initializing a feasible route in a low-dimensional space, the A*-RRT, which was developed by Brunner et al. (2013), is a two-phase approach that continues with the application of the RRT algorithm to search in the high-dimensional space. This technique was designed by Brunner et al. A further noteworthy example is this particular model. It was proved that these algorithms, which are based on RRT, have an excellent track record of search capacity when it comes to addressing the motion planning issues that the robot was encountering.

When compared to the other algorithms that are currently available, RRTs seem to be particularly effective when it comes to planning in motion patterns that are both high-dimensional and complex. Additionally, they are flexible and may be altered to coincide with the specific operational conditions of each given occurrence. This is a significant advantage. One of the limitations of the RRTs that are now in use is that they examine the configuration space, which is also referred to as C-space, via the use of random node sampling methodology. It is possible that this strategy will result in a slow convergence or incorrect paths. To add insult to injury, these methods are notoriously difficult to generalize, and they often depend on hand engineering in order to adjust to the new environment. For those readers who are interested in a comprehensive analysis of the traditional motion planning methodologies, we would like to point them in the direction of the review that Wahab and his colleagues wrote with the year 2020 in mind. According to Qureshi et al.'s research from 2021, the most major issues that continue to arise in the motion planning optimizer are biased sampling and the curse of dimensionality. In order to bridge the gap between theory and reality and enable robots to take over increasingly complicated jobs, especially in labor-intensive activities such as assembly, motion planning techniques should be effective and computationally efficient. This will allow robots to take over more complex tasks. The curse of dimensionality and biassed sampling are two more issues that need to be addressed. Motion planning that is developed via the process of learning While the applications of machine learning in this field have only been around for a relatively short period of time, the literature on classical motion planning has been around for quite some time. In recent years, researchers have been investigating contemporary methods such as learning (Berenson et al., 2012) and neural network-based methods (Bency et al., 2019; Duguleana&Mogan, 2016; Qureshi & Yip, 2018) with the intention of either improving the effectiveness of approximation methods or improving the speed at which the planning process is carried out in complex operational scenarios. An investigation on a learning-based technique that encourages hand-eye synchronization for single-arm robot grasping was carried out by Levine et al. (2018). They relied upon the modern motion planning literature that is currently available within the field. In 2017, Yahya and his colleagues developed a novel approach to robotic door opening tasks that was based on a distributed policy search. This technique was called the distributed policy search.

A deep reinforcement learning approach was proposed by Popov et al. (2017) for the goal of achieving dexterous handling of single-arm assembly robots to achieve the desired result. Deep reinforcement learning was further extended for the aim of task planning for single-arm robots working in three-dimensional environments (Ichter et al., 2018). This was done in order to improve the efficiency of the robots. When Ji et al. (2019) were seeking for a collision-free path for single-arm robotic operations, they came up with a distinct reinforcement learning-based approach that they dubbed Q-learning. This technique was developed in order to fulfil their search. In order to enhance the search method for locating the (near-) optimum route, Qureshi et al. (2021) built a learning-based planner that utilized neural networks. This was done in order to improve the search methodology. This planner was developed to address the challenges that generic robots have when it comes to motion planning. Watt and Yoshiyasu (2020) were able to design routes in two-dimensional environments while avoiding obstacles by using a neural planner that was based on learning. All of this took place not too long ago. A approach for wide robotic tasks that is based on soft actor–critic planning was created by Prianto et al. (2020). This method was developed by the researchers. Only a small number of studies have been conducted that are pertinent to the design of collision-free trajectories in this particular scenario. When compared to the planning of single-arm operating environments, the process of path planning for dual-arm assembly robots in three-dimensional environments is much more involved and computationally more thorough. Our motivation to design a computationally efficient deep learning-based optimization technique for the motion planning of dual-arm assembly robots in both two-dimensional and three-dimensional settings came from the fact that this gap, in conjunction with the requirement for real-time planning methods, was the most likely to motivate us to design such a technique.

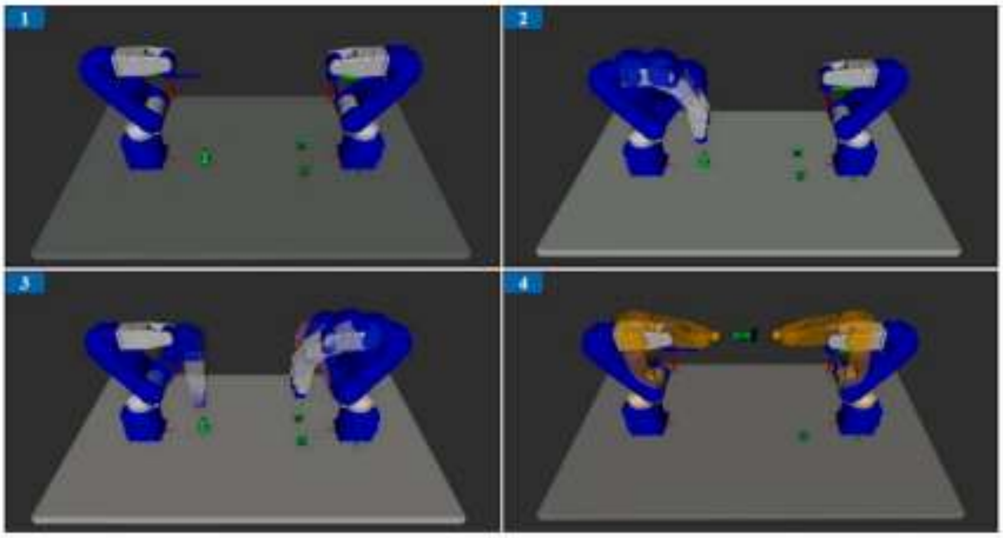
The location of the arms should be translated to goal configurations as input to the LSTM-BiRRT algorithm for the purpose of generating the overall plan. Provided that there are two target portions in the OS matrix, this should be done. When planning the trajectory of the operating arm in the configuration space, the algorithm takes into account the trajectory of the supporting arm. This involves planning the trajectory of the operating arm from the moment the operating arm begins to move until the moment the supporting arm stops moving. Whenever the algorithm has finished planning the path that the operating arm will take, it will then return it to the configuration space in order to update the knowledge on the obstacles. At this moment, the connection of the operational arm has been fixed, and as a result, the supporting arm planner considers it to be an impediment. Figure 2 illustrates the progression of the LSTMBiRRT technique for the dual-goal planning issue. The schedule may be found here. During the planning process, there are two possible outcomes that are shown in Figure 3. In the first case, the robotic arms do not reach the overlapping zone, which means that they are able to function independently. As a result, the planner finds both trajectories concurrently in order to speed up the planning operation. However, in the second case, the objective of two robotic arms overlaps, which means that the arms enter the overlapping zone and have the potential to crash with one another. In this scenario, one of the arms is considered to be the moving obstacle, and the motion of the other arm is planned taking into consideration a trajectory that is free of collisions within a dynamic environment that is already known. It is of utmost importance that the motion planning algorithm be both efficient and accurate when it comes to dual-goal applications in high-dimensional situations. This is because one planning trajectory has to be altered in conjunction with the other trajectory. In order to improve the motion planning process, the algorithm that was created for this research makes use of a learning-based mechanism that is embedded inside a bidirectional-sampling-based planner.

4. Result Analysis

In this part, a numerical analysis is presented to assess the effectiveness of the LSTM-BiRRT in terms of its ability to solve motion planning issues in both two-dimensional and three-dimensional contexts. When carrying out the trials, two separate sets are taken into consideration: (1) a. The first is a Single Environment with Multiple-initial-and-goal Configurations (SEMC), while the second is a Multiple Environments with Single-initial-and-goal Configurations (MESC). For the purpose of establishing communication between the hardware environment and the motion planning algorithm, the "MoveIt!" platform (Chitta, 2016) is used. This allows for the selection of the desired exercise to be performed at a certain workspace while taking into consideration the same configuration. Among the methods that are regarded to be benchmark algorithms are the LSTMBiRRT, RRT-Connect, RRT*, and bidirectional RRT* (Bi-RRT*) algorithms. The results are compared taking into consideration the amount of time required for computing and the length of the route that is produced. Taking into consideration SEMC instances, the results are summarized. However, with the exception of one of the 2D SEMC situations in which RRT* records the lowest computational time, LSTM-BiRRT outperforms in every other case when both the computational time and the solution quality are taken into consideration to evaluate performance.

As can be seen in, the disparity in performance between the LSTM-BiRRT and the other benchmark algorithms becomes considerably more significant in situations involving 3D SEMC. This condition may be explained by the fact that three-dimensional settings are more complicated, and the learning impact becomes more significant as the dimensionality of the issue increases. Cases involving MESC are examined. The LSTM-BiRRT is capable of producing all of the best solutions in both the 2D and 3D version of the MESC problem. The fact that our newly created method is able to achieve a greater solution quality in a shorter amount of computing time in comparison to RRT-Connect, RRT*, and Bi-RRT* is one of the aspects that makes this achievement especially noteworthy. The second significant discovery is that LSTM-BiRRT maintains a consistent performance when the computing time it takes to process MESC and SEMC instances in the same environment is taken into consideration. With an average processing time of 19 milliseconds, the LSTM-BiRRT method works more efficiently than the Bi-RRT* algorithm, which takes 53 milliseconds, the RRT* technique takes 56 milliseconds, and the RRT-Connect algorithm takes 67 milliseconds. In addition, the planned routes that are provided by LSTMBiRRT are, on average, seven percent shorter than those that are provided by Bi-RRT*, which performs better than RRT-Connect and RRT*. It has also been noticed that the performance of RRT* and Bi-RRT* is similar, however the performance of RRT-Connect was shown to be much less competitive in comparison to the other benchmark algorithms, regardless of the size of the challenge. An illustrated example for building a product with three pieces is carried out in both simulated and real-world contexts in order to examine whether or not the findings are applicable to real-world situations. Taking into consideration the product shown in Figure 1, the components were created with the assistance of a 3D printer. demonstrates the simulation model that is now operating based on the LSTM-BiRRT planner in order to carry out the described assembly operation. All of the components were arranged on a, with an area that overlapped in the center of the arrangement. The OS matrix is taken into consideration when the robot takes hold of the components that have been allotted to it. With a computational time of less than 0.02 seconds, it is evident that the planner was effective in guiding the end-effectors of the dualarm robot to the desired position in order to grab the component parts that were given to them without causing any collisions.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 2, No. 2, February : 2023



After that, the planner carried out its tasks by using the Robot Operating System while simultaneously connecting to the control modules of two independent robotic arms. The motion was planned by the planner using preset 3D models that simulated a real-world workspace. This was done in light of the trajectories that were planned utilizing the LSTM-BiRRT framework. In order to communicate with the control modules, the planner used the Robot Operating System as a communication bridge and sent the predicted path. As shown in Figure 10(b), the LSTM-BiRRT is responsible for navigating the robotic arms in order to grab the components and put together the finished product. This method is comprised of the following four stages: (1) The operational arms and the supporting arms are responsible for grabbing components A and B, respectively, in order to carry out the necessary motion for the assembly process. Secondly, the component A is assembled onto the component B by the operating arm. (3) The component C is collected and moved by the working arm, while the supporting arm handles the completed section, which is denoted by the letter AB. (4) Assembly of component C on AB is performed by the operating arm. provides a representation of the amount of time required for the robotic motion planning process in each assembly phase, taking into account five replications.

5. Conclusions

In this study, a motion planning method that is based on a heuristic and is based on deep learning was provided for the purpose of implementation in intelligent manufacturing systems applications. Deep learning's capacity to extract high-level features is combined with the exploration power of randomized motion planning algorithms in this algorithm, which allows it to solve path planning problems in complex environments. In particular, it is useful for planning collision-free trajectories for dual-arm robots that are performing assembly tasks. Within the context of assembly activities, LSTMBiRRT is useful for tackling issues including dual-goal and high-dimensional planning. During the process of evaluating the created method's performance in various planning scenarios in both two-dimensional and three-dimensional workspaces, it was benchmarked against the most advanced algorithms that are currently available in the literature. According to the numerical findings, the LSTM-BiRRT runs at a rate that is, on average, four times quicker than the benchmark algorithms. Not only that, but the length of the optimal route that was calculated by the LSTM-BiRRT was around seven percent lower than the length of the bidirectional RRT* method, which was the approach that performed the best. It is important to highlight two of the fundamental reasons why the algorithm that was built is better than other algorithms. To begin, the End-to-End design of the LSTMBiRRT method, which has input and output of the same format, makes it easier to use the algorithm in a variety of operating situations without necessitating difficult pre-processing of the data. Second, the deep learning component of the LSTM-BiRRT algorithm enhances the mechanism for avoiding obstacles by anticipating the future location of the route nodes according to the training data that has been collected before. When taken as a whole, these characteristics play a key part in enhancing the effectiveness of the motion planning algorithms while also guaranteeing collision-free and (near-) optimal traverses.

6 References

[1] Wahab, M. N. A., Nefti-Meziani, S., &Atyabi, A. (2020). A comparative review on mobile robot path planning: Classical or meta-heuristic methods? Annual Reviews in Control, 50, 233–252.

[2] Bao, H. P., &Liou, K. (1990). Space maps manipulation for robot motion planning. Computers & Industrial Engineering, 18(2), 235–245.

[3] Basile, F., Caccavale, F., Chiacchio, P., Coppola, J., &Curatella, C. (2012). Task-oriented motion planning for multi-arm robotic systems. Robotics and Computer-Integrated Manufacturing, 28(5), 569–582.

[4] Bency, M. J., Qureshi, A. H., & Yip, M. C. (2019). Neural path planning: Fixed time, nearoptimal path generation via oracle imitation. ArXiv Preprint ArXiv:1904.11102.

[5] Berenson, D., Abbeel, P., & Goldberg, K. (2012). A robot path planning framework that learns from experience. In 2012 IEEE International Conference on Robotics and Automation (pp. 3671–3678).

[6] Bohlin, R., &Kavraki, L. E. (2000). Path planning using lazy PRM. In Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065) (Vol. 1, pp. 521–528).

[7] Boor, V., Overmars, M. H., & Van Der Stappen, A. F. (1999). The Gaussian sampling strategy for probabilistic roadmap planners. In Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C), 2, 1018–1023.

[8] Brunner, M., Brüggemann, B., & Schulz, D. (2013). Hierarchical rough terrain motion planning using an optimal sampling-based method. In 2013 IEEE International Conference on Robotics and Automation (pp. 5539–5544).

[9] Ceriani, N. M., Zanchettin, A. M., & Rocco, P. (2016). Collision avoidance with task constraints and kinematic limitations for dual arm robots. In Intelligent Autonomous Systems 13 (pp. 1285–1299). Springer.

[10] Chen, C.-P., Wang, P.-J., Wang, H., Wang, H.-P., & Chen, C.-C. (2015). Developing industrial dual arm robot for flexible assembly through reachability map.

[11] Cheng, C.-Y., Pourhejazy, P., Ying, K.-C., Li, S.-F., & Chang, C.-W. (2020). Learning-based metaheuristic for scheduling unrelated parallel machines with uncertain setup times. IEEE Access, 8, 74065–74082. https://doi.org/10.1109/Access.628763910.1109/ ACCESS.2020.2988274

[12] P., Ying, K.-C., & Lin, C.-F. (2021). Unsupervised Learningbased Artificial Bee Colony for minimizing non-value-adding operations. Applied Soft Computing, 105, 107280. https://doi.org/10.1016/j.asoc.2021.107280

[13] Chitta, S. (2016). MoveIt!: An introduction. In Robot Operating System (ROS) (pp. 3–27). Springer.

[14] De Pace, F., Manuri, F., Sanna, A., &Fornaro, C. (2020). A systematic review of Augmented Reality interfaces for collaborative industrial robots. Computers & Industrial Engineering, 149, Article 106806.

[15] Devaurs, D., Sim´eon, T., &Cort´es, J. (2014). A multi-tree extension of the Transitionbased RRT: Application to ordering-and-pathfinding problems in continuous cost spaces. In 2014 IEEE/RSJ international conference on intelligent robots and systems (pp. 2991–2996).

[16] Do, H. M., Park, C., & Kyung, J. H. (2012). Dual arm robot for packaging and assembling of IT products. In 2012 IEEE International Conference on Automation Science and Engineering (CASE) (pp. 1067–1070).

[17] Duguleana, M., &Mogan, G. (2016). Neural networks based reinforcement learning for mobile robots obstacle avoidance. Expert Systems with Applications, 62, 104–115.

[18] Elbanhawi, M., &Simic, M. (2014). Sampling-based robot motion planning: A review. IEEE Access, 2, 56–77.

[19] Farber, M. (2017). Configuration spaces and robot motion planning algorithms. ArXiv Preprint ArXiv:1701.02083.

[20] Ferguson, D., &Stentz, A. (2006). Anytime rrts. In 2006 IEEE/RSJ international conference on intelligent robots and systems (pp. 5369–5375).

[21] Gao, W., Tang, Q., Yao, J., & Yang, Y. (2020). Automatic motion planning for complex welding problems by considering angular redundancy. Robotics and ComputerIntegrated Manufacturing, 62, 101862. https://doi.org/10.1016/j.rcim.2019.101862

[22] Geismar, N., Manoj, U. V., Sethi, A., &Sriskandarajah, C. (2012). Scheduling robotic cells served by a dual-arm robot. IIE Transactions, 44(3), 230–248.

[23] Guruji, A. K., Agarwal, H., &Parsediya, D. K. (2016). Time-efficient A* algorithm for robot path planning. Procedia Technology, 23, 144–149.

[24] Hart, P., Nilsson, N., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), 100–107.

[25] Hochreiter, S., &Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

[26] Hockstein, N. G., Gourin, C. G., Faust, R. A., &Terris, D. J. (2007). A history of robots: From science fiction to surgical robotics. Journal of Robotic Surgery, 1(2), 113–118.

[27] Ichter, B., Harrison, J., &Pavone, M. (2018). Learning sampling distributions for robot motion planning. IEEE International Conference on Robotics and Automation (ICRA), 2018, 7087–7094.

[28] Ji, M., Zhang, L., & Wang, S. (2019). A Path Planning Approach Based on Q-learning for Robot Arm. In 2019 3rd International Conference on Robotics and Automation Sciences (ICRAS) (pp. 15–19).

[29] Jordan, M., & Perez, A. (2013). Optimal bidirectional rapidly-exploring random trees. Juan, A. A., Faulin, J., Ferrer, A., Lourenço, H. R., & Barrios, B. (2013). MIRHA: Multistart biased randomization of heuristics with adaptive local search for solving nonsmooth routing problems. Top, 21(1), 109–132.

[30] Kala, R. (2013). Rapidly exploring random graphs: Motion planning of multiple mobile robots. Advanced Robotics, 27(14), 1113–1122.

[31] Karaman, S., &Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research, 30(7), 846–894.

[32] Kavraki, L. E., Svestka, P., Latombe, J.-C., &Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation, 12(4), 566–580.

[33] Kim, D.-H., Lim, S.-J., Lee, D.-H., Lee, J. Y., & Han, C.-S. (2013). A RRT-based motion planning of dual-arm robot for (Dis) assembly tasks. IEEE ISR, 2013, 1–6. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. ArXiv Preprint ArXiv:1412.6980.

[34] Krüger, J., Schreck, G., &Surdilovic, D. (2011). Dual arm robot for flexible and cooperative assembly. CIRP Annals, 60(1),

[35] Kuffner, J. J., &LaValle, S. M. (2000). An efficient approach to single-query path planning. In IEEE international conference on robotics and automation. San Francisco (pp. 473–479).