Robotics and Autonomous Systems 194 (2025) 105118

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Trajectory planning system for bimanual robots: Achieving efficient collision-free manipulation

Francisco José Martínez-Peral [a], Jorge Borrell Méndez [a], Dennis Mronga [b], José Vicente Segura-Heras [c], Carlos Perez-Vidal [a,*]

[a] Engineering Research Institute of Elche (I3E), Miguel Hernandez University of Elche, Avda. Universidad, s/n, Elche, 03202, Alicante, Spain
[b] Robotics Innovation Centre, German Research Centre for Artificial Intelligence (DFKI), Robert-Hooke Str. 1, Bremen, 28359, Germany
[c] Operations Research Center (CIO), Miguel Hernandez University of Elche, Avda. Ferrocarril s/n, Elche, 03202, Alicante, Spain

ARTICLE INFO

Keywords:
Pick-and-place Operations
Robotic Sequence Order Optimization
Dual-arm Collision Avoidance
Binary Integer Linear Programming
ROS

ABSTRACT

Pick-and-place operations are non-value-added activities but essential in many industrial processes. Some of these operations must be performed by dual-arm robots, which represent new challenges in terms of collision-avoidance due to the use of a shared workspace. This work addresses these two issues by proposing a Task and Motion Architecture (TMA) designed to optimize pick-and-place tasks, ensuring efficient and safe operation through collision-free movements. This architecture consists of two interconnected sublayers, the Task Planner (TP) and the Global Motion Planner (GMP). The TP calculates the optimal sequence of operations, minimizing the total execution time and guaranteeing a collision-free sequence. The GMP plans the trajectories of the robotic arms using predefined motion strategies and following the calculated optimal sequence. This work presents a novel solution for enhancing the efficiency of robot coordination in real-world settings by integrating an intercommunicated TP and MP. Results from simulations demonstrate improved task efficiency, reduced operational times, and successful collision avoidance between robots.

1. Introduction

Nowadays, robotic systems play an important role in increasing efficiency and reducing industrial production costs. In this context, bimanual robots provide a technological solution with a higher potential, especially in handling and assembly tasks where the coordination of movements or speed are critical. Unlike non-concurrent robotic systems, bimanual robots can perform more complex tasks mimicking human manipulation, such as the assembly of small components that require two arms for assembly [1], the simultaneous manipulation of a single object due to its size or mechanical characteristics [2] or the independent manipulation of objects sharing the workspace [3]. In the latter case, the generation of independent trajectories introduces additional complexities in trajectory planning and task optimization due to the risk of collisions between robots and the need to synchronize their movements efficiently. These challenges highlight the importance of developing robust optimization methods that not only avoid collisions, but also minimize the total handling time [4], thus maximizing system productivity.

The aim of this work in the field of bimanual robotics is to advance in the development of algorithms that are able to plan online and execute the trajectory without collisions using an optimal sequencing strategy, such as [5]. In the present study, a novel two-layered architecture has been developed. The first layer, called Task Planner (TP), gets the optimal sequence to execute the task, and the second layer, called Global Motion Planner (GMP), calculates robotic arm trajectories and provides information to the TP if a collision occurs for a specified sequence. This iterative step is performed until a collision-free sequence is obtained and the movements are programmed following some movement strategy. This approach aims to significantly improve the assembly speed and operational efficiency of bi-manual robots by providing an optimal motion sequence of trajectories that reduces downtime, avoids collisions and increases productivity. This article details the development and validation of such a model, evaluating its effectiveness by comparing different architectures.

The paper is organized as follows. Section 2 provides an overview of the current state of the art in trajectory planning and optimization, as well as collision detection and collision avoidance. Section 3 defines the architecture proposed in this research by outlining its two control layers: the Task Planner (TP) and the Global Motion Planner (GMP). Results of the research, together with a brief analysis, are presented

* Corresponding author.
E-mail address: carlos.perez@umh.es (C. Perez-Vidal).

https://doi.org/10.1016/j.robot.2025.105118
Received 8 January 2025; Received in revised form 29 May 2025; Accepted 20 June 2025
Available online 15 July 2025

in Section 4, as well as simulations and experiments. Finally, Section 5 describes conclusions and outlines possible avenues for future research.

2. Related work

Although they still account for a low percentage of installed robots, bi-manual robots are emerging as advanced solutions for complex tasks that require the coordination of two arms, leading to increased interest and investment in this field. Research on bimanual robots has focused on tasks such as path planning, arm coordination, and task optimization to avoid collisions and maximize operational efficiency. According to Stavridis and Doulgeri [6], the design of algorithms for coordinating the movements of concurrent bimanual robots is essential to fully exploit their capabilities in dynamic and potentially unpredictable environments. Moreover, bimanual systems offer advantages in terms of manipulation of objects of various geometries and sizes, which is especially useful in industries such as electronics, textile or automotive, where the objects to be manipulated can vary considerably. For example, Beetz et al. [7] describes how bimanual robots can be programmed to perform complex tasks by translating human instructions from the web into robotic action plans. This requires advanced perceptual and reasoning capabilities, enabling the robots to interpret ambiguous instructions, identify objects, and coordinate movements effectively. However, a key challenge remains in optimizing manipulation time and reducing inefficient movements.

Azimirad and Shorakaei [8] proposed a dual hierarchical control approach that fuses genetic algorithms and optimal control for the global planning of robotic trajectories. This methodology overcomes the limitations of analytical and evolutionary techniques by integrating them to ensure compatibility with dynamic equations and circumvent local minima. The deployment of this methodology in non-holonomic mobile manipulators has resulted in notable improvements in efficiency and cost-effectiveness. In order to optimize the sequencing of robotic assembly operations and guarantee collision-free paths, Givehchi et al. [9] proposes a methodology that integrates evolutionary optimization techniques and virtual manufacturing. This methodology addresses the challenge of optimizing efficiency in automated processes by generating optimal sequences that minimize total assembly time and ensure safe robot movements. In the context of bimanual robots, the study [10] on optimizing manipulation sequences in bimanual robots provides a framework for understanding how deep learning methods can be applied to significantly improve the operational efficiency of these systems. The integration of these advances in bimanual robots aims not only to improve the efficiency of automated production processes, but also to expand the possibilities for automation in areas that were previously considered too complex to be automated.

Collision detection is a fundamental aspect in the planning and execution of concurrent robotic tasks, for which different solutions have been proposed in recent years, where much research has been based on representing robotic systems in convex volumes. For example, in [11], the *TrajOpt* algorithm is proposed as a solution to the problem of finding collision-free trajectories from collision-prone straight-line initialization. Based on a sequential convex optimization methodology that penalizes collisions, the penalty coefficients are adjusted as necessary to avoid collisions. In [12], the implementation of swept volumes of spheres has been employed as a means to more accurately model manipulators and obstacles. Karush-Kuhn–Tucker (KKT) conditions are integrated into the calculation of the minimum distance between objects and additional control variables are introduced to eliminate the non-smoothness of the distance constraints. Furthermore, additional controls and constraints are used for each possible collision, the number of which increases in proportion to the number of obstacles. Other methods for programming the trajectories of two robotic arms have been developed based on the discretization of the workspace. In [13], an approach to avoid collisions and dynamic deadlocks between robotic manipulators in shared spaces is proposed. A real-time motion control algorithm based on non-linear predictive control is used, which is formulated as a multi-agent system. Each robot behaves independently, taking into account the movements of other robots, which provides a scalable and distributed solution for path planning and real-time coordination of multiple robotic manipulators. However, this single-agent approach generates an inefficient solution in terms of execution time. In [3], a system is defined that studies all possible position configurations that the arms of an ABB-YuMi robot could adopt in the workspace and uses a Markov Decision Process (MDP) to obtain the best sequence of motions to execute the task.

Trajectory planning on bi-manual robots is essential to ensure safe and efficient movements, but its complexity is significantly increased due to the need to coordinate two arms operating in a shared space. An optimal trajectory should minimize task time or energy consumption while maintaining safe operations. In [14], a review of different methods for scheduling and executing assembly tasks is given, highlighting the importance of achieving collision-free operations efficiently in unstructured environments. Research in this field has proposed several methods for path planning, including dynamic programming, genetic algorithms and machine learning. Genetic algorithms have been successfully employed in [15] to optimize trajectories in complex industrial environments, demonstrating significant improvements in efficiency and collision risk reduction. In addition, real-time trajectory planning is an area of research that seeks to adapt trajectories in response to dynamic changes in the environment or in the robot's tasks. In [16], a machine learning-based framework was developed, incorporating motion planning algorithms to accelerate learning, proposing a reinforcement learning (RL) approach.

Trajectory optimization on bi-manual robots is an extension of path planning, where not only feasible trajectories are pursued, but also the most efficient ones in terms of time, energy and safety. This process involves checking trajectories to minimize the overall task duration, while ensuring that movements are collision-free. One of the approaches in this field is the use of optimization techniques based on physical and mathematical models. In [17], a robust optimization framework for industrial robotic systems is proposed, considering kinematic and dynamic uncertainties. The design objective is established by incorporating positioning and joint torque errors. A moment-based method is used in conjunction with computational optimization techniques and the effectiveness is demonstrated through practical examples. Another approach in this direction is based on reinforcement learning (RL) to improve policies in high dimensional systems. In [18], a trajectory model-based RL framework is proposed. This approach allows a bimanual humanoid robot to learn to efficiently manipulate objects, even in dynamic environments, using a limited number of interactions with the environment. In [19], a deep learning approach for robotic manipulation tasks is presented. This method, which uses off-policy training of deep Q-functions, can scale to complex manipulation tasks and learn policies from deep neural networks efficiently enough to be trained on real physical robots. In addition, multi-objective optimization has gained attention in trajectory planning for robotic arms. In [20], a multi-objective optimization approach, based on deep reinforcement learning and optimal planning, is proposed. This approach uses a combination of direct and inverse kinematics together with deep neural networks to generate and optimize optimal trajectories.

The combination of sequencing and path planning leads to more complexity in achieving further system optimization. As in [21], the complexity of combining sequencing and path planning in automated PCB assembly is addressed using a meta-learning-based approach. The proposed method integrates probabilistic modeling and interactive feedback to optimize robotic positioning and assembly processes, thereby overcoming challenges such as occlusions and component variability. By enabling continuous learning and adaptation, this strategy simplifies the optimization of task sequences and motion trajectories, thereby achieving greater efficiency and flexibility in complex, multivariate production systems.

The architecture proposed represents a significant advance in the optimization of pick-and-place tasks on two-arm robots, offering a comprehensive solution that combines task planning and motion planning in an intelligent and coordinated manner. In contrast to conventional methodologies, which typically lack a task planner, or, in the few instances where one is present, it is limited to unidirectional communication with the path planner. The proposed approach integrates a task planner based on the CR-BILP algorithm and a Global Motion Planner that considers the simultaneous coordination of both robots. The system's bidirectional communication capability enables it to establish an optimal object picking order, which minimizes execution times and prevents collisions. The architecture significantly reduces cycle times, ensures task safety and improves operational efficiency, thus establishing itself as a robust and reliable tool for a wide range of robotic applications.

3. Methodological development

3.1. Nomenclature

The following is the notation used in this paper:

- **TP**: Task Planner.
- **GMP**: Global Motion Planner.
- $\mathbf{MP_k}$: Motion Planner robot k.
- **K**: Set of robots. There shall be as many elements in this set as there are robots available.
- **S**: Set of robots available in each case.
- **I**: A set of initial pieces nodes. All have an assigned and known position.
- **J**: A set of place nodes of parts. All have an assigned and known position.
- **p**: Number of pieces per configuration.
- $\mathbf{x_{kij}}$: Binary variable that determines the route followed by each robot. If this variable takes the value 1, it indicates that the robot k shall perform the movement described by the consecutive nodes i and j, where i is the initial node and j the final node.
- $\mathbf{t_{kij}}$: Estimated movement time of the route followed by each robot k form the node i to node j.
- **r**: Number of iterations in solving the binary linear programming problem.
- $\mathbf{X^{r*}}$: Binary vector. Movements following the optimal sequence.
- $\mathbf{X_k^{r*}}$: Binary vector. Movements following the optimal sequence X^{r*} performed by the robot k.
- $\tau_{\mathbf{k}}^{\mathbf{c}}$: Coordinated trajectory programmed by MoveIt! of the robot k.
- $\delta_{\mathbf{k_a},\mathbf{k_b}}$: Collision detected between the robot k_a and the robot k_b.

3.2. System architecture

The architecture proposed in this work, called Task and Motion Architecture (TMA), is based on a control scheme in charge of optimizing the pick-and-place task to be performed by several robotic arms. The proposed architecture has been designed to be applicable to different robotic systems, taking advantage of the flexibility of ROS, which allows robots to be easily exchanged and movements to be efficiently planned using MoveIt!.

Thanks to this, the Task Planner (TP) dynamically adjusts the data used from the robotic model, such as estimated execution times, reachability and collision detection. Furthermore, the Global Motion Planner (GMP) maintains its functionality regardless of the hardware, as the planning is adapted by simply modifying the parameters of the robot used. This approach is not limited to a specific system and can be applied in different robotic environments without requiring structural changes to the architecture. This system is made up of two sub-layers with bidirectional communication between them, the Task Planner and the Global Motion Planner.

As its name suggests, the Task Planner is a task optimization tool that, based on the positions of the pieces, together with the estimated time, reachability and collision detection information of the system, defines the optimal sequence for picking up the pieces. On the other hand, the Global Motion Planner starts from the sequence determined by the TP, and its objective is to program and plan the movements of the robots to perform the task in an optimal and collision-free way, avoiding collisions not only between the robots but also with the environment. The architecture has been developed as shown in Fig. 1. The communication between the two sublayers is bidirectional. The TP generates a sequence that is sent to the MP, which is responsible for planning a collision-free path. In turn, the GMP provides information to the TP about possible collisions in the defined sequence.

The TP sends to the GMP the sequence to execute, where a series of planning checks are performed by MoveIt! to verify if each robot can perform the movements, so that no collision with the other robot or with the environment is detected. In case of collisions while executing the task with linear movements by both robots, the GMP tries to adapt the trajectories to avoid collisions. Being is responsible for coordinating the different robots to perform the task effectively and without collisions.

Once the optimal, collision-free trajectories for each of the robotic arms have been planned, they are sent to the robots for execution. Once optimal collision-free trajectories have been planned for each of the robotic arms, they are sent to the robots for execution. The planning part of the architecture is shown in black, while the execution is shown in blue, as shown in Fig. 1. This methodology establishes a set of criteria to execute the optimal sequence set by the TP and ensure that no collisions are encountered.

3.3. Task planner

In contemporary industry, programming and defining sequences for robotic arms to perform specific tasks is a meticulous and structured process that adheres to a number of fundamental steps. Before starting the programming process, it is vitally important to have a thorough understanding of the task to be performed by the robot. This requires clearly defining the objectives, identifying the initial and final conditions, and determining the resources required, including tools and materials. This preliminary analysis ensures that all aspects of the task are well defined, thus facilitating subsequent planning and scheduling. Task scheduling can be done in a fixed or random sequence, depending on the specific nature and requirements of the task. In systems where objects arrive at random positions, the choice of a random sequence means that the completion time of the sequence can be either the optimal or the slowest. The use of a TP is of vital importance when it comes to sequencing robotic arms, as it allows to organize and optimize the actions needed to complete a task efficiently.

In [5], the sequence is calculated by solving a Binary Integer Linear Programming (BILP) problem, selecting the best possible sequence. The algorithm is dynamically adapted to each arrangement of pieces, taking into account the distance between their positions. As a result, a sequence is generated that minimizes the total distance covered by the robotic arms to complete the task. The BILP algorithm was originally designed for general robotics applications, especially in scenarios where motion constraints and range limitations were not a significant factor.

The R-BILP (Reachability-based Binary Integer Linear Programming) algorithm represents a considerable advance over the original BILP, as it addresses the main limitations of its design. The R-BILP algorithm developed in this paper incorporates the reach limitations of anthropomorphic robotic arms, ensuring that all computed task sequences remain feasible within the physical constraints of the system. In addition, instead of relying on the distance between positions to optimize task execution, the R-BILP algorithm uses movement times

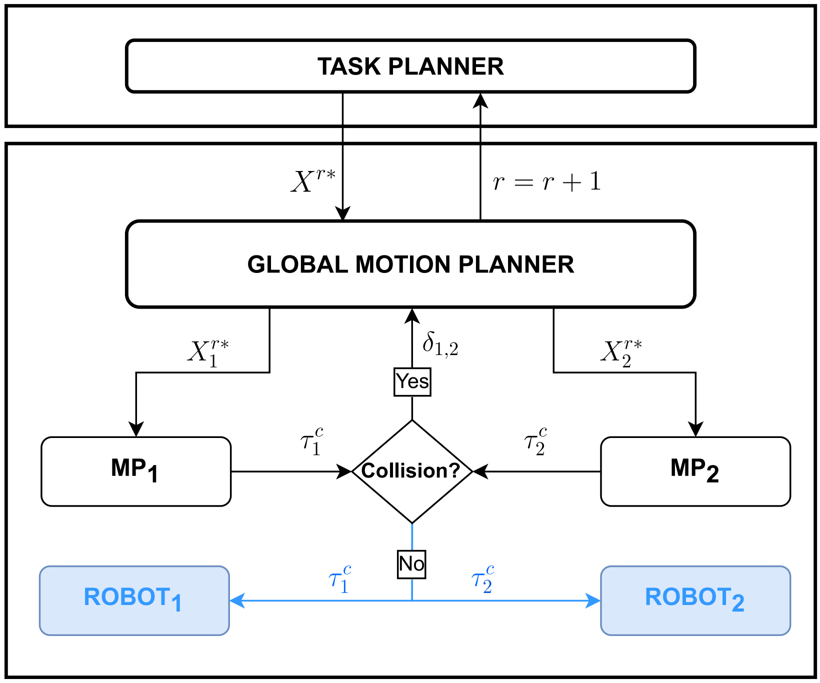


Fig. 1. Task and motion architecture scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

estimated from MoveIt! to improve the accuracy and efficiency of motion planning, especially in spatially constrained environments. The modification of the optimization criteria allows for more realistic and efficient planning, significantly reducing the overall task duration. Without this modification, many of the sequences generated by the original BILP would be invalid due to the inability of the robotic arms to reach certain objects. In contrast, R-BILP guarantees the validity of the sequences generated by the task planner in various system configurations, overcoming previous approaches that relied on fixed or random sequences.

Among the equations defining the algorithm, Eq. (1) establishes the objective function of the algorithm to minimize the total task execution time. Eq. (10) describes the reachability constraint, which takes into account the information whether a piece is reachable by one of the robots, by both or by neither. In a BILP algorithm, nodes represent key points, such as pick and place positions of the pieces. They indicate the locations that the robot must visit, and the connections between them are optimized to minimize routes and times by considering constraints. The algorithm's constraints are specified in [5] for a better understanding.

$$\min \sum_{k\in K} \sum_{i,j} t_{kij} x_{kij} \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{p} x_{ksi} = 1 \qquad \forall k = s, k \in K, s \in S \tag{2}$$

$$\sum_{k=1}^{2} x_{kij} = 1 \qquad \forall i = j, i \in I, j \in J \tag{3}$$

$$\sum_{j=1}^{p} x_{kjs} = 1 \qquad \forall k = s, k \in K, s \in S \tag{4}$$

$$\sum_{k=s=1}^{2} x_{ksi} + \sum_{k=1}^{2} \sum_{j=i} x_{kji} = 1 \qquad \forall i \in I \tag{5}$$

$$\sum_{i=j=1}^{p} x_{kij} \leq p - 1 \qquad \forall k \in K \tag{6}$$

$$\sum_{k=s=1}^{2} \sum_{i\neq j} x_{kji} + \sum_{k=s=1}^{2} x_{kjs} = 1 \qquad \forall j \in J \tag{7}$$

$$\sum_{i\neq j} x_{kji} + x_{kjs} - x_{kmj} = 0 \qquad \forall j = J, k = s, m = j, m \in I \tag{8}$$

$$\sum_{j=1}^{p} x_{kji} + x_{ksi} - x_{kim} = 0 \qquad \forall i \in I, k = s, m = i, m \in I \tag{9}$$

$$\sum_{k=s=1}^{2} x_{ksi} + \sum_{i=j=1}^{p} x_{kij} + \sum_{i\neq j}^{p} x_{kji} + \sum_{k=s=1}^{2} x_{kjs} = 0 \qquad \forall k \in K; \forall i \in I; \forall j \in J \tag{10}$$

After running a series of simulations, it was observed that the improvements incorporated in the R-BILP algorithm led to a significant enhancement in task performance, achieving both complete reachability of all pieces and a reduction in execution time. However, in scenarios where the generated sequence caused collisions, the total simulation time increased considerably due to the need for additional steps, such as motion replanning. In the case of the R-BILP algorithm, as it does not consider possible collisions as explicit constraints, its application is more suitable in systems where the probability of collision with other objects is low, such as in mobile robotics in open environments or in aerial applications.

These conceptual and practical limitations motivated the development of a new algorithm, designated as Collision-free Reachability-Based Binary Integer Linear Programming (CR-BILP), which incorporates collision checking directly into the planning process. This enhancement ensures the generation of feasible, collision-free, and time-efficient sequences, making it particularly suitable for systems where, due to the configuration of the manipulators or the proximity between elements, there is a higher probability of collisions during task execution. In addition to the two improvements already introduced by R-BILP, CR-BILP adds the ability to evaluate and avoid collisions during planning, a feature absent in the original BILP formulation [5] and in the R-BILP algorithm. This added capability improves the safety and efficiency of task execution, allowing the Task Planner (TP) to refine the final sequence accordingly.

The implementation of the CR-BILP algorithm incorporates a two-way feedback and communication system between the Task Planner (TP) and the Global Motion Planner (GMP), enabling dynamic interaction during the planning process. The algorithm employs an iterative

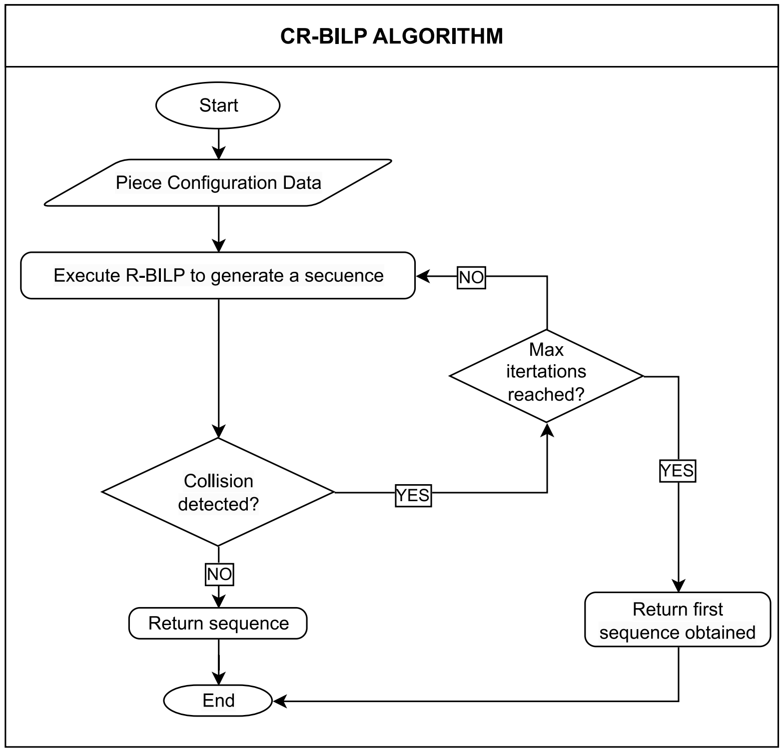


Fig. 2. Flowchart of the CR-BILP algorithm.

procedure: it begins by executing the R-BILP algorithm to generate a candidate sequence, which is then evaluated for potential collisions. If any collisions are detected, the algorithm is re-executed while excluding the previously invalid sequences. Eq. (11) describes the constraint introduced into the optimization model, ensuring that the selected solution is both collision-free and minimizes the total execution time among all feasible alternatives.

$$\sum_{k,i,j} x_{kij} < 1 \qquad x_{kij} \in X^{r*} \tag{11}$$

In summary, Eq. (11) allows us to eliminate from the problem the optimal solutions associated with collisions that are obtained in the first r iterations. This approach allows the calculation of a sequence that not only guarantees the absence of collisions between the robotic arms in most cases, but also optimizes the execution time by selecting the most efficient sequence among all valid options. In cases where it is not possible to identify a collision-free sequence within the specified limits, the sequence obtained in the initial iteration will be considered as a valid solution, giving priority to the minimization of the execution time in the GMP. Fig. 2 presents a flowchart illustrating the operation of the CR-BILP algorithm.

3.4. Global motion planner

In this work, the Global Motion Planner layer plays a fundamental role in guaranteeing the precision and coordination of the robot's movements in a shared environment. This component, programmed in Python and simulated in RVIZ, has the main objective of executing the sequence of movements previously defined by the TP, thus ensuring a collision-free execution. The TP uses the CR-BILP algorithm to determine the optimal sequence of task execution. In most cases, the algorithm is able to generate a collision-free sequence, thus allowing both robots to move simultaneously during the execution of a task. In the case that CR-BILP is not able to identify a collision-free sequence, the sequence provided is the optimal one calculated by the R-BILP algorithm. The GMP is in charge of planning the movements of both robotic arms. For this purpose, three movement strategies have been considered: sequential, simultaneous hybrid and simultaneous parallel.

In situations where the sequence provided by the TP corresponds to the one calculated by the R-BILP, a collision detection process is performed. This process is based on the calculation of the movement plan for each robot of the system, enabling verification of potential collisions in any state of the plan. Using MoveIt!, the planned joint values in the trajectories are analyzed to determine whether each state is valid or whether a collision has been detected or is unreachable for the robots. In this way, all necessary trajectories are evaluated, and the GMP defines the appropriate type of movement to be performed.

3.4.1. Sequential movement

A simple method of avoiding collisions between robots is to ensure that they do not occupy the same workspace. In this approach, each component is handled sequentially by the robotic arm whose workspace encompasses the designated pick and place positions. Sequential movement has been defined as a process in which each robotic arm completes its designated set of tasks before the next robot begins its own. This approach effectively eliminates the possibility of collisions by preventing both robotic arms from operating simultaneously within the same workspace. However, this approach also has the disadvantage of reducing operational efficiency by reducing the possibility of parallel operations, as the tasks assigned to the robots are performed in order rather than at the same time.

3.4.2. Simultaneous hybrid movements

Following the implementation of sequential movements, a movement strategy has been proposed that allows the robots to operate at the same time, with the aim of reducing task completion times. However, the main challenge associated with this approach is to ensure the prevention of collisions that may occur during the execution of the tasks assigned to each arm. To address this issue, a hybrid motion model has been developed that establishes a set of priorities for robot coordination.

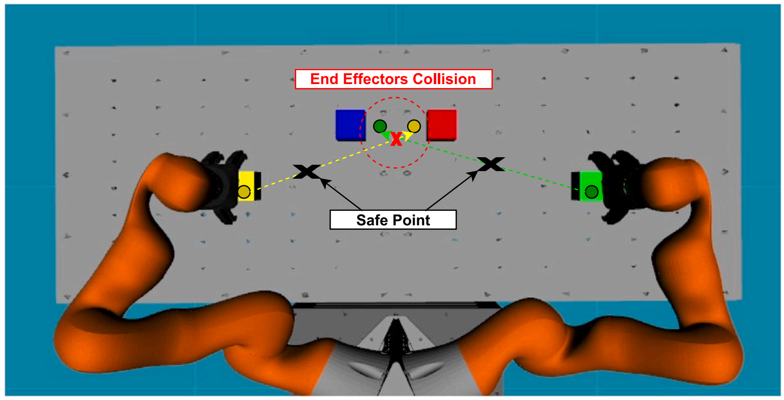


Fig. 3. Collision detected between end effectors in the pieces' place process.

The GMP determines which pieces are to be picked up by each robot and in which order, based on the sequence provided by the TP. It is the responsibility of the MPs of each robot to plan the movements and trajectories. In case of collision during the movement, the information is transmitted to the GMP. In the hybrid movement model, the planning is adjusted to allow the robots to move simultaneously for as long as possible, until a safe point is reached, which is a position away from the detected collision point. Subsequently, the robots switch to a sequential strategy to avoid collisions and complete the task efficiently.

The order of the movements is determined on a sequential basis, with the robot closest to the target being selected as the initial movement. This approach ensures that, in the event of a collision, the robots can perform the pick-and-place task without obstacles. As illustrated in Fig. 3, the robots can cross each other to deposit each part in its final position, which could lead to a collision. The safe point allows the robots to move simultaneously to that position and then switch to the sequential movement strategy to complete the task.

The Safe Point was calculated by following the planned trajectory for each robot from its end point to its start point, checking at which point there is no collision and defining this position as the Safe Point. To do this, we compared the performance of three search algorithms: Linear [22], Binary [23] and Exponential [24]. Linear search is the simplest method, in which each element of a list is traversed sequentially to locate the target. This method is ideal for small or unordered lists. Binary search, which is a more efficient method, applies to ordered lists and consists of repeatedly dividing the search interval in half until the target value is found. Exponential search is applicable to exceptionally large or potentially infinite lists and combines the identification of a search area by progressive doubling of an index with a binary search. In the present study, these three algorithms have been adapted in order to verify the state of the robotic arms at each iteration. This is achieved by comparing the trajectories to identify collision-free points and allowing simultaneous movement of the robots. Several tests using the different algorithms were conducted using five different configurations of pieces and the three different search algorithms. The mean time required to identify the safe point was 0.66 s for Binary search, while the other algorithms required 0.99 s for Exponential search and 2.90 s for Linear search. The results demonstrate that the Binary search algorithm is the most effective in fulfilling this task, exhibiting a 78% improvement over the Linear search algorithm and a 31.9% improvement over the Exponential search algorithm in terms of average time. In conclusion, Binary search is the most efficient algorithm for large and ordered data sets due to its logarithmic time complexity. The Binary search method involves repeatedly dividing the search space into two equal parts, which allows for the identification of the desired safe point in a relatively short time.

The hybrid strategy not only minimizes downtime with respect to the sequential movement of the robots defined in Section 3.4.1, but also maximizes operational efficiency by avoiding collisions. The application of this strategy can significantly improve the productivity and operational safety of multi-robot systems. This, in turn, facilitates the automation of complex processes and increases response capacity to variations in the working environment.

3.4.3. Simultaneous parallel movements

In terms of time optimization, the hybrid strategy improves overall times compared to sequential movement, but shows room for improvement, as having one robot immobile while waiting for the other to perform its tasks leads to significant time loss. Therefore, a parallel simultaneous movement strategy has been defined. This new movement strategy for a multi-robot system is not only able to avoid collisions during task execution, but also further minimizes task duration compared to the hybrid movements defined in Section 3.4.2. Following a parallel movement approach, it allows the robotic arms to move simultaneously, avoiding collisions and completing tasks in the shortest possible time. The main objective of this movement strategy is to keep the robots in an operational state for as long as possible.

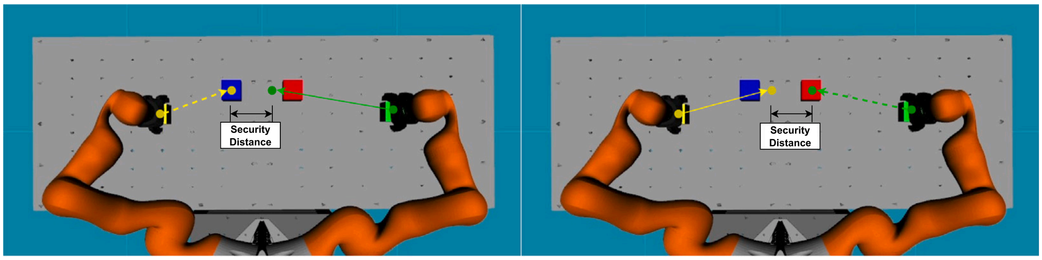
In the event of a collision between the robots, the end effectors move to predefined positions, which are defined according to a criterion that takes into account both the distance and the safety of the movements. An arbitrage has been defined, in which first, the robot closest to the target position is defined as the master robot and moves directly to that position. The second arm, the slave robot, adopts a position relative to the master, maintaining a safety distance in which both end-effectors remain aligned on the same axis in parallel. This safety distance has been determined as the minimum necessary for both effectors to execute their movements without risk of collision, optimizing the planning and reducing the total time of the movements. Once the master robot has completed the task of picking up the corresponding pieces, the process is reversed: the slave robot assumes the role of master and moves to its destination point, while the other robot takes on the role of slave. This ensures a safe and efficient execution of the process. A representation of this movement can be seen in Fig. 4.

The parallel movement strategy allows simultaneous operation of both robots for a greater proportion of the task compared to hybrid strategy. This system significantly improves operational efficiency by optimizing the continuous operational duration of the robots, thereby reducing the overall time required to complete operations. In addition, the risk of collisions is mitigated by assigning master and slave functions to the robots, while maintaining predefined safety distances. This ensures smoother and safer operation in shared environments, optimizing both the efficiency and safety of the multi-robot system.

4. Experimental results

4.1. Experimental setup

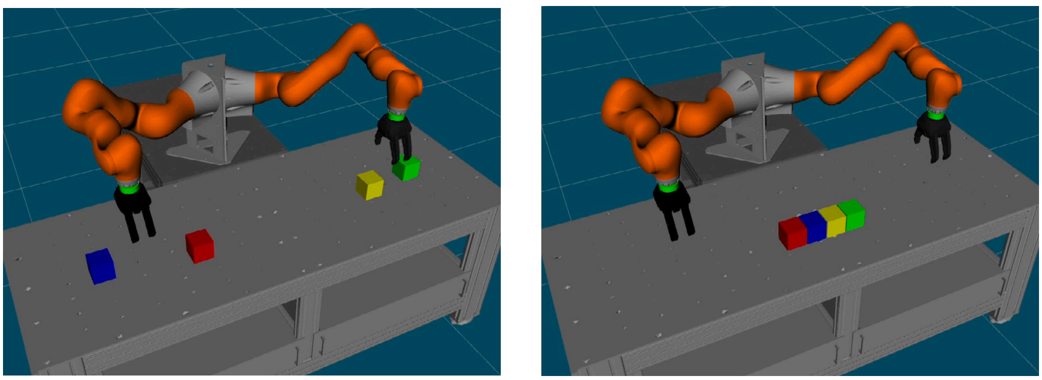
The robotic system used in this research consists of two KUKA LBR iiwa robots [25], which are 7 DOF robots with sensitive capabilities,



(a) Master robot positioning (right arm).

(b) Slave robot positioning (left arm).

Fig. 4. Defined positions in parallel movement to avoid collisions.



(a) Beginning of the task (picks).

(b) End of the task (places).

Fig. 5. Composition of the pick-and-place system. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

making them ideal for working collaboratively with people in assembly tasks. Both robotic arms are equipped with 3-finger grippers from ROBOTIQ [26], which are suitable for greater flexibility and versatility in manipulation tasks. The KUKA LBR iiwa is an industrial manipulator that is usually operated via the KUKA Sunrise Cabinet. In this work, it has been integrated into ROS (Robot Operating System) [27], an open-source robotics platform. ROS provides a modular and flexible environment for the development, control and coordination of complex robotic systems, allowing seamless integration of components and facilitating the development of advanced control and motion planning algorithms. ROS 1 Noetic has been used with MoveIt! 1, integrating RVIZ for simulation and OMPL for path planning. The environment was configured on Ubuntu 20.04 using MoveIt! 1 version 1.1.13. A PC with AMD Ryzen 9 5900X 12-Core Processor 3.70 GHz, 32 GB RAM, NVIDIA GeForce RTX 4060 Ti graphics card and 500 GB of memory was used for its execution.

As shown in Fig. 5, the system is composed of two KUKA LBR iiwa robots equipped with 3-finger grippers from ROBOTIQ. The task involves picking up four pieces from random initial positions and placing them in predefined final positions.

This work has focused on solving a four-piece manipulation problem (see Fig. 5), each problem to be solved is called configuration. These pieces have been identified with different colors and each one is in a random pick position (pick as shown in Fig. 5(a)) and must be transferred to a fixed place position (place as shown in Fig. 5(b)). The main objective of the system is to place each part in its final position (place) in the shortest possible time and without generating any collision of the robots with each other or with the environment. This system has applications in various industries, including manufacturing, logistics, food industry and assembly process automation.

The developed architecture has been implemented in an experimental real system within the iMRK project (https://robotik.dfki-bremen.de/en/research/robot-systems/imrk), carried out by the DFKI in Bremen, Germany. The aim of this project is to optimize the workspace utilization between collaborative robots in pick-and-place tasks, ensuring that they can share a workspace without compromising efficiency or safety. Fig. 6 shows an image of the real robot system, illustrating one of the configurations tested within the iMRK project.

Once the Task Planner (TP) and Global Motion Planner (GMP) have been defined, a series of simulations have been performed to evaluate the sequencing algorithms of the TP and to compare the improvements in accuracy and efficiency in the manipulation of objects by the GMP. The TP determines the order in which the pieces are picked, and the GMP uses this sequence to program and simulate the execution of the task.

In the initial phase of the analysis, 1,000 simulations were conducted to examine the effects of different configurations. In the first part of the analysis, as detailed in Section 4.2, the sequence generated by the R-BILP algorithm was compared with three other sequences representing potential solutions to the problem. These sequences were obtained by calculating all feasible solutions of picking order for each of the 1,000 configurations, resulting in a range of 24 to 96 different viable sequences. From them, the best sequence (BSEQ), the worst sequence (WSEQ) and 500 randomly selected sequences (RSEQ), representative of common practice in the industry, were selected. To compare these sequences, an estimated simulation time calculated by MoveIt! was analyzed, rather than the actual execution time on a physical robot. This theoretical estimation, used within the Task Planner, serves as a criterion to evaluate and select the most efficient sequences.

In the second part of the analysis, focusing on the optimization of the movements defined in the GMP (Section 4.3), the 1000 configurations were used again, together with the sequences mentioned in the previous paragraph, with the exception of the RSEQ, from which one of the 500 sequences used previously was selected. The effectiveness of the architecture developed in this study is illustrated by a comparative analysis of the total simulation duration. The Simulation Time used in the second part has been used to compare different simulations, taking into account all the movement strategies of the GMP, together with all

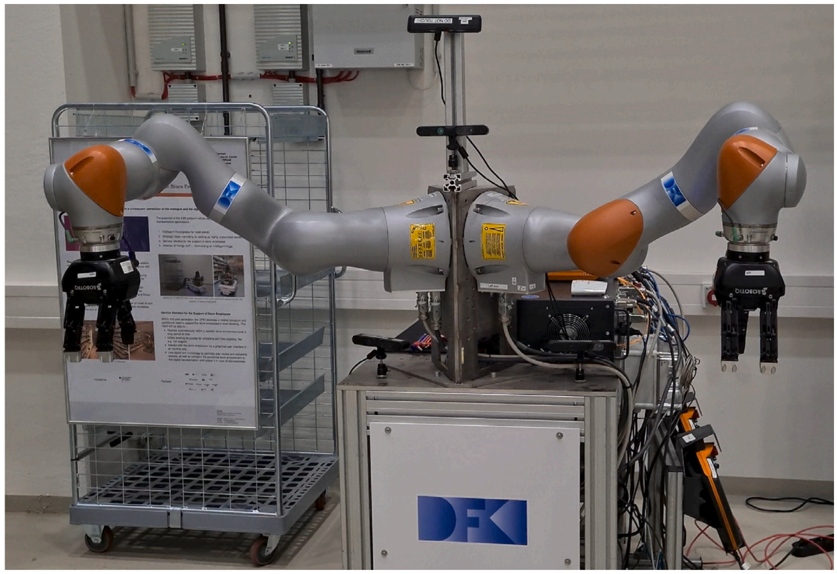


Fig. 6. Real robotic setup.

the possible sequences generated by the TP. This time covers the entire simulation process, from the planning of movements and detection of possible collisions by the GMP, to the execution of the simulation with MoveIt! in RVIZ. For this purpose, the different movement strategies capable of avoiding collisions between the robotic arms, described in Section 3.4, were used to determine the extent of the improvements that each of them could offer. Furthermore, these movements have been compared with those of the CR-BILP, which employs a collision-free sequence and is designed as a hybrid simultaneous movement. Finally, the results of simulations and real-world implementations are presented.

4.2. Analysis of task planner

In this first phase of the analysis, the efficiency of the sequencing algorithms defined in this study, as described in Section 3.3, was evaluated. The results showed that the 500 random solutions generated had an average time of 37.00 to 37.25 s when considering the 1000 configurations. However, the average execution time with the CR-BILP algorithm was significantly lower at 35.58 s, representing an improvement of 3.84% to 4.46% depending on the random sequence used as a benchmark. In the least favorable scenario, the WSEQ algorithm resulted in an average time of 38.69 s, highlighting an 8.03% improvement achieved by the CR-BILP algorithm. Statistical analysis confirmed that these differences are significant, with a p-value < 0.001. Notably, the BSEQ sequence, representing the best sequence, was identical to the sequence calculated by the CR-BILP algorithm. This confirms the effectiveness of the CR-BILP algorithm in identifying the optimal sequence for task performance. Additionally, the mean times of the 500 random sequences were consistent within the specified range, with the WSEQ algorithm consistently showing the poorest performance. The results of this analysis confirm the clear superiority of the CR-BILP algorithm over existing alternatives, demonstrating a marked improvement in the optimization of the total task time.

It has been observed that the execution times of the original BILP [5] and the R-BILP are very similar in terms of sequence generation, with values of 0.00443 s and 0.00481 s, respectively. This suggests that the enhancements introduced in the R-BILP, particularly the integration of estimated movement times and reachability constraints, do not significantly affect computation time. However, they do improve the fidelity of the optimization process to the actual robotic execution.

In contrast, the CR-BILP algorithm shows a substantial increase in total computation time, reaching an average of approximately 1.133 s. As shown in the table, the majority of this time (around 1.128 s) is spent on collision verification, which involves planning robot motions and validating that the trajectories are collision-free within the workspace. This additional cost is a direct consequence of incorporating real-time collision checking into the task planning loop.

While the computational cost of the CR-BILP algorithm is higher than that of the others, this overhead is often justified by the benefits it provides during execution. When a valid, collision-free sequence is identified, the robot can execute the task more efficiently and reliably, eliminating the need for motion replanning or task interruption. Therefore, the additional planning time leads to shorter and smoother execution times in practice, as well as greater robustness in constrained environments. This justifies the increased complexity of the CR-BILP approach.

As shown in Table 1, the computation times involved in generating a sequence have been divided into two components: sequence calculation time (T. Seq. Calc.) and collision verification time (T. Col. Check). This separation enables a more precise analysis of the computational load introduced by each stage of the algorithm. In addition, the constraints integrated into each algorithm have been presented. This includes the magnitude they minimize (either distance or time), whether they take into account the reachability of objects, and whether they introduce collision checking.

As demonstrated in Section 4.3.4, a comparison of the simulation times clearly illustrates the efficacy of the TP approach, facilitated by the CR-BILP algorithm, in comparison to random sequences. This observation has been corroborated not only by the estimated times calculated by MoveIt!, but also by the results obtained from the simulated system. The deployment of an algorithm in the TP layer is corroborated by the substantial reductions in execution times observed. The calculated sequence optimizes the robot movements, thus minimizing the total time needed to complete the task. This is achieved by avoiding unnecessary movements and ensuring more direct and efficient trajectories with the sequential strategy.

4.3. Analysis of global motion planner

In the second stage of the analysis, the different movement strategies detailed in Section 3.4 were compared by simulation in RVIZ. This allowed an estimation of the efficiency of task execution under real conditions. In addition, the second algorithm developed in this work, CR-BILP, has been included in the comparison to observe the

Table 1
Average sequence time and constraints handled per algorithm. (Times expressed in seconds).

Algorithm	T. Seq. Calc.	T. Col. Check	Mag.	Reachability	Collision check
BILP [5]	0.00443	–	Distance	No	No
R-BILP	0.00481	–	Time	Yes	No
CR-BILP	0.00487	1.12824	Time	Yes	Yes

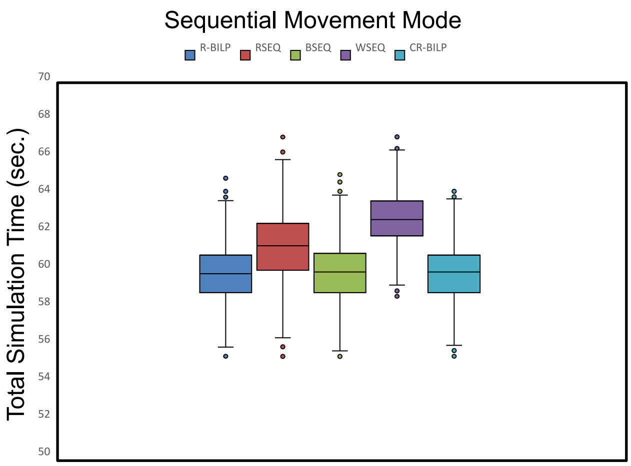


Fig. 7. Total simulation time (sequential movement).

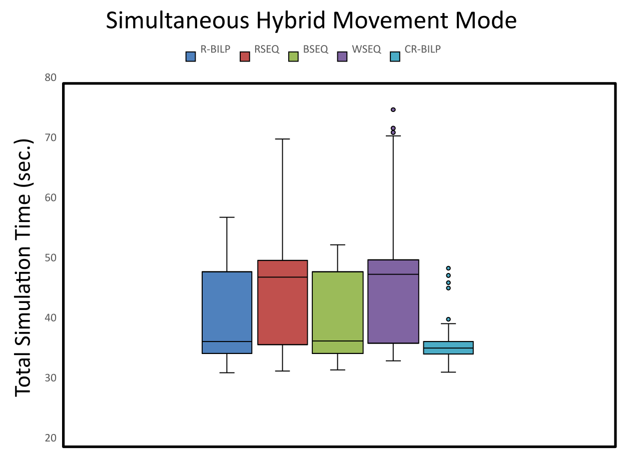


Fig. 8. Total simulation time (Simultaneous hybrid movement).

differences between sequences that have collisions and those that do not. In order to verify the correct functioning and compare the effectiveness of the different movement strategies defined in the GMP, the total simulation time was used. This time includes both the time taken by the TP to calculate the task sequence and the time taken by the GMP to plan the movements with MoveIt! and simulate them in RVIZ. To ensure a fair comparison, a constant speed of movement was set for the robots in all cases.

4.3.1. Results of sequential movements

In the sequential movement strategy, the trajectory is delineated in a way that allows one robot to pick up all the assigned components before the other robot starts its task. This approach guarantees the absence of collisions, thus omitting the need for detailed collision analysis. Therefore, the results of the CR-BILP algorithm are identical to those obtained with the R-BILP in this case. As demonstrated in Section 4.2, the implementation of the TP significantly improves the optimization of pick-and-place operations.

As illustrated in Fig. 7, the R-BILP algorithm guarantees the sequence that minimizes the total simulation time with the sequential motion strategy. The average simulation time of the R-BILP and CR-BILP algorithms is approximately 59.9 s, which is an improvement of 2.44% over RSEQ, whose average time is 61.4 s, and 4.62% over WSEQ, whose average time is 62.7 s.

4.3.2. Results of simultaneous hybrid movements

Hybrid simultaneous movement allows both robotic arms to initiate their respective movements in synchronization. In the event of a collision, the system switches to a sequential movement strategy to avoid the collision. Table 2 illustrates the number of collisions produced by the different algorithms used in the TP. As can be seen, the CR-BILP algorithm did not produce any collisions, with 98.8% of the sequences completing the task in two pick-and-place processes, while the remaining 1.2% were completed in three processes (one of the robots picked up three pieces and the other only one). The R-BILP identified 54.3% of the sequences without collisions, but 46.6% of the time there was at least one collision between the robots, with similar results for the BSEQ. All other possible solutions showed lower rates of collision-free sequences.

In addition, Fig. 8 presents a graphical representation of the total simulation times of all cases, which clarifies the performance disparities between them. In some cases, simulation times above the average were observed for the CR-BILP method, corresponding to situations where the optimal execution sequence was divided into three phases. In these cases, two objects were picked up simultaneously by both arms in a collision-free movement, while the remaining objects were only accessible by one of the manipulators. These particular conditions can be considered as limiting scenarios, where the random spatial configuration of the pieces restricts the temporal efficiency of the method. However, it is important to note that, even in these situations, the algorithm was able to generate completely collision-free and time-optimized trajectories.

4.3.3. Results of simultaneous parallel movements

Finally, the third strategy proposed in this study consists of carrying out trajectories with the end effectors in parallel in the event of a collision, so that all possible collisions are avoided while reducing the time the robots are stationary. Obviously, the possible collision cases correspond to those shown in Table 2. On the other hand, the new simulation times can be seen in Table 3.

Fig. 9 presents in more detail the distribution of total simulation times for all algorithms.

4.3.4. General comparison

The results indicate that the use of a sequencing algorithm leads to a significant performance improvement. Furthermore, when a collision-free sequence, such as that computed by CR-BILP, is available, it enables optimal task execution in terms of total simulation time. Table 4 presents a summary of the average total simulation times, which includes both the planning time for the robotic arm movements based on the Task Planner (TP) sequence and the task simulation time in RVIZ.

For the Sequential Movement strategy, both R-BILP and CR-BILP compute the optimal sequence and achieve an average execution time of 59.93 s, with CR-BILP improving execution time by 2.27% over RSEQ and 4.59% over WSEQ. For the Simultaneous Hybrid Movement strategy, CR-BILP achieves an average time of 36.33 s, improving

Table 2
Collision analysis and simulation times for the simultaneous hybrid motion strategy. *N*: number of executions (absolute number) along with percentage of cases (expressed in %); simulation times and mean simulation time (expressed in seconds).

Algorithm	No collision (2 processes)		No collision (3 processes)		Collision (2 processes)		Collision (3 processes)		Mean Sim. Time
	N	Sim. Time	N	Sim. Time	N	Sim. Time	N	Sim. Time	
R-BILP	533 (53.3%)	34.56	1 (0.1%)	47.53	466 (46.6%)	47.63	0 (0.0%)	–	40.15
RSEQ	483 (48.3%)	35.62	30 (3.0%)	49.51	473 (47.3%)	49.05	14 (1.4%)	66.41	42.22
WSEQ	490 (49.0%)	36.00	0 (0.0%)	–	424 (42.4%)	49.23	86 (8.6%)	66.86	43.36
BSEQ	533 (53.3%)	34.56	1 (0.1%)	47.53	466 (46.6%)	47.64	0 (0.0%)	–	40.15
CR-BILP	988 (98.8%)	35.21	12 (1.2%)	47.64	0 (0.0%)	–	0 (0.0%)	–	35.34

Table 3
Simulation time analysis of simultaneous parallel movement strategy (expressed in seconds).

Algorithm	No collision (2 processes)	No collision (3 processes)	Collision (2 processes)	Collision (3 processes)	Mean simulation time
R-BILP	34.57	55.93	40.68	–	37.31
RSEQ	35.60	49.46	41.85	58.69	39.08
WSEQ	35.97	–	42.22	59.19	40.18
BSEQ	34.57	55.93	40.68	–	37.31
CR-BILP	35.22	47.59	–	–	35.35

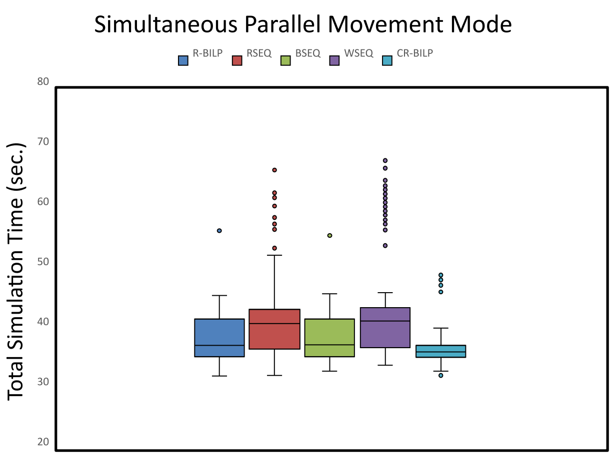


Fig. 9. Total simulation time (Simultaneous Parallel movement).

Table 4
Average total simulation times comparison (expressed in seconds).

	R-BILP	RSEQ	WSEQ	BSEQ	CR-BILP
Sequential movement	59.93	61.32	62.81	59.94	59.93
Simultaneous hybrid movement	41.31	43.45	44.55	41.31	36.33
Simultaneous parallel movement	38.32	40.16	41.22	38.32	36.33

performance by 12.1% over R-BILP by eliminating collisions. The improvement increases to 16.4% over RSEQ and 18.5% over WSEQ. Finally, with the Simultaneous Parallel Movement strategy, CR-BILP maintains the same average time and achieves an improvement of 5.2% over R-BILP, 9.6% over RSEQ and 11.9% over WSEQ.

The implementation of the CR-BILP sequencing algorithm developed in this work ensures that all components are accessible thanks to the consideration of robot reachability within the system. By incorporating estimated simulation times, it enables a more accurate estimation of task duration, reinforcing the importance of a Task Planner (TP) in this architecture. Simultaneous operation of both robots significantly reduces task completion time in a shared workspace. A robust collision management system is essential to ensure safe and efficient motion planning. The simultaneous hybrid and parallel movement strategies developed in this study have shown significant improvements over sequential execution.

In conclusion, the development of CR-BILP has demonstrated that collision avoidance can significantly improve task efficiency. When collisions are unavoidable, the Global Motion Planner (GMP) can adopt a Simultaneous Parallel Movement strategy that effectively handles collisions while reducing task execution time. This approach has been shown to improve efficiency by up to 39.4% compared to sequential movements strategy.

4.4. Implementation of the developed architecture

4.4.1. Simulations

To verify the correct functioning of the system developed in this work, the TMA has been subjected to simulation tests that faithfully reflect the real behavior of the system. These simulations confirm the system's effectiveness under conditions close to reality.

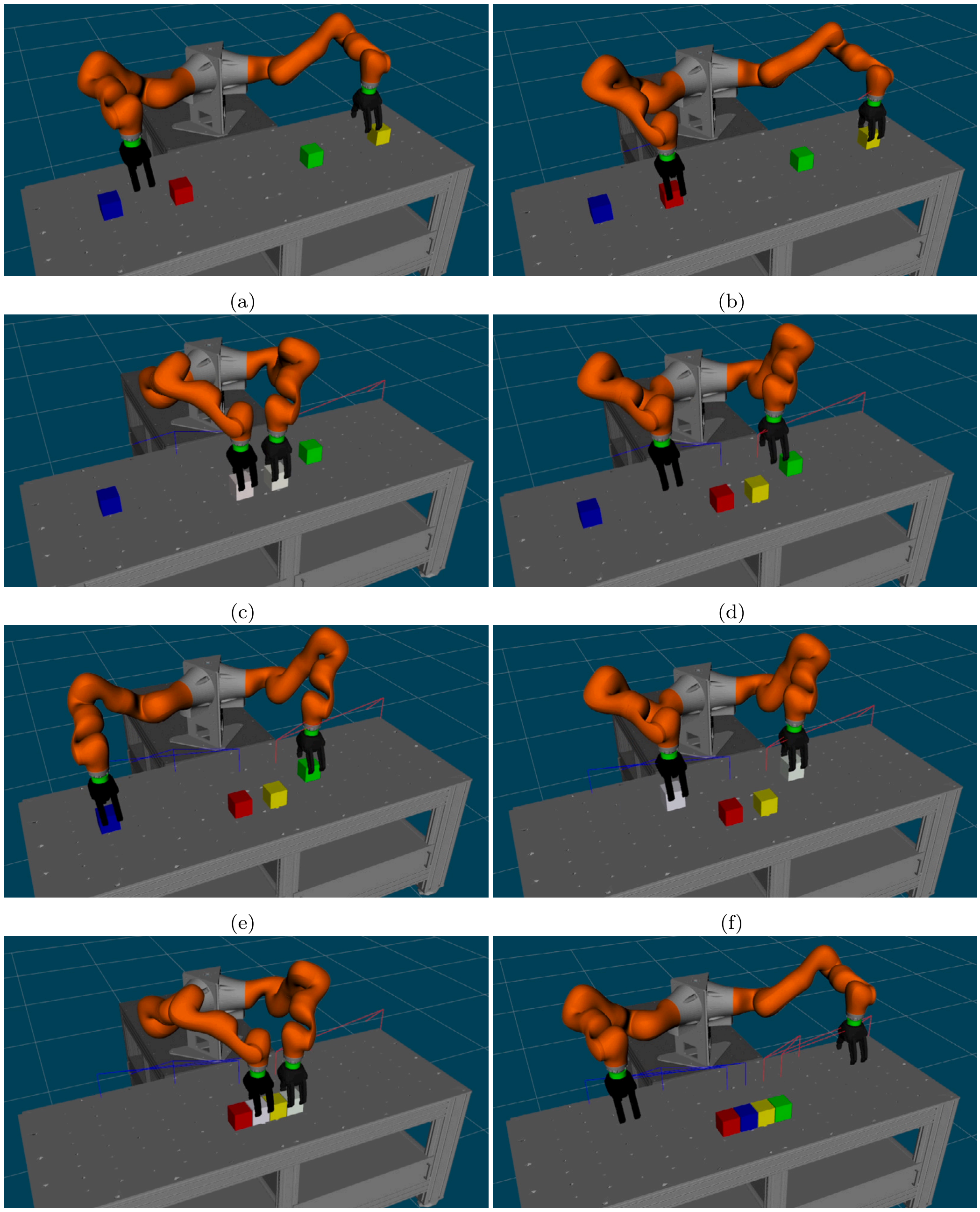
The first movement strategy evaluated was sequential, where one robot picked up all the components and then the other proceeded to do the same. Although this process is slow and not simultaneous, it guarantees the absence of collisions (https://youtu.be/DHUJu9Gg3hs). Regarding simultaneous movement strategies, hybrid and parallel movements were also simulated. According to the sequence calculated by the TP, a collision was detected during the operation of picking up and placing the first two components. In the hybrid strategy, the robots are placed in a safe position before executing the movements sequentially. No collisions were detected during the handling of the second pair of pieces, allowing the operation to be executed simultaneously and safely (https://youtu.be/4SqOmCSEeeI).

In the parallel movement strategy, the robots were programmed to move their end effectors to parallel positions, maintaining a safe distance in the first part of the task to avoid the detected collision. The second part of the task was completed without collisions, allowing the remaining steps to be executed smoothly and simultaneously (https://youtu.be/4scWX4f7QL4). Fig. 10 presents a series of frames is presented, in which the TP was programmed using the CR-BILP algorithm. The execution of the task by the robots was demonstrated, with the trajectories they followed being displayed, thereby providing a clearer visualization of their collision-free movements. A sequence was obtained that is not only collision-free but also optimizes the total simulation time (https://youtu.be/5v2IwyFSmlY).

4.4.2. Real system experiments

In order to verify the correct functioning of the collision detection applied in the motion strategies defined in Section 3.4 of this work, some tests were performed on the real system. These tests were carried out on the real system (Section 4.1) at the German Research Center for Artificial Intelligence (DFKI) in Bremen.

It is important to note that, due to safety concerns, these tests were conducted without the manipulation of any objects. The primary objective of the study was to verify the architecture developed in this



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 10. Simultaneous movements with the CR-BILP algorithm.

research, with the aim of evaluating the effectiveness of the planning and identifying potential collisions during task execution. To verify the effectiveness of the proposed architecture, a test was performed using the sequence calculated by the R-BILP in the TP, according to the simultaneous hybrid movements strategy of the GMP.

The test performed focuses on collision detection from the sequence calculated by TP, and subsequent motion planning following a Simultaneous hybrid strategy (Section 3.4.2). As demonstrated in the video, the robots move in the first picking process of two pieces simultaneously as no collision has been detected and are able to perform the pick-and-place task. However, in the second set of pieces, a collision was detected during the pick process. Consequently, the robots moved to a safe position, enabling them to complete the pick and place tasks sequentially. Prior to the real-world tests, the planned experiment was first simulated to validate its execution. The video of the test run in simulation on the actual system can be found here: https://youtu.be/lBTWwHd_KTY. This demonstration serves to validate the effectiveness of the architecture defined in this work in detecting collisions in a real robotic system. The video recording of the execution on the real system is available at the following link: https://youtu.be/efy6BJxBgoQ.

5. Conclusions

This study proposes a novel Task and Motion Architecture (TMA) to optimize pick-and-place tasks in robotic systems with multiple anthropomorphic manipulators. Unlike conventional methods, the TMA integrates a two-layer planning system with bi-directional communication, enabling dynamic task optimization. While this study focuses on a bimanual setup, its scalable and flexible design allows for seamless integration into multi-arm robotic systems. Exploiting the adaptability of ROS, the architecture enables easy robot replacement and efficient motion planning through MoveIt!, extending its applicability to industrial and collaborative environments.

A key contribution is the Collision-Free Reachability-Based Integer Linear Programming (CR-BILP) algorithm, which extends R-BILP by incorporating collision detection into sequence planning. Through bidirectional communication between the Task Planner (TP) and the Global Motion Planner (GMP), CR-BILP optimizes the order of execution based on MoveIt! time estimates while ensuring feasibility and collision-free operation, significantly improving efficiency.

In addition, this work introduces adaptive simultaneous motion strategies that dynamically adjust execution based on the planned sequence and detected collisions. Implemented within the GMP, these strategies optimize execution while avoiding collisions. The Simultaneous Hybrid Movement strategy allows both robots to move together until a collision is predicted, switching to sequential execution when necessary. Simultaneous Parallel Movement strategy allows full synchronization of both robots. If a collision is anticipated, predefined parallel positions allow continuous movement without interruption. If the sequence generated by the TP is collision-free, both robots perform fully synchronized movements, resulting in smoother and faster task completion.

A comprehensive system evaluation evaluated the TP and CR-BILP. For each configuration, all possible solutions were generated and three key sequences were analyzed: the best (BSEQ), the worst (WSEQ) and 500 random (RSEQ). The results show that CR-BILP improves performance by 4% to 8% over other sequence optimization approaches when using MoveIt! time estimates, findings that were validated by real simulation times.

Further simulations compared algorithm performance with actual execution times. The results show that using Simultaneous Parallel Movements with CR-BILP improves execution time by 11.9% over WSEQ, 9.6% over RSEQ and 5.2% over R-BILP. Overall, integrating CR-BILP into the TP and enabling the GMP to coordinate simultaneous movements reduces execution time by up to 40% compared to sequential motion systems, confirming that the proposed architecture improves efficiency and addresses collision avoidance in dual-arm pick-and-place tasks.

Future work will focus on optimizing the workspace and refining the positioning of the robotic arms to further improve efficiency. The scalability of CR-BILP will be evaluated with larger object sets to ensure computational efficiency while maintaining performance gains. In addition, research will explore adaptive motion planning techniques to improve the adaptability of robot motion in the real world, minimizing collisions and execution times.

CRediT authorship contribution statement

Francisco José Martínez-Peral: Investigation, Methodology, Writing – original draft, Validation, Writing – review & editing, Visualization, Software, Conceptualization. **Jorge Borrell Méndez:** Validation, Software, Investigation, Conceptualization, Visualization, Supervision, Methodology, Data curation. **Dennis Mronga:** Validation, Conceptualization, Writing – review & editing, Project administration, Writing – original draft, Supervision, Formal analysis. **José Vicente Segura-Heras:** Writing – review & editing, Supervision, Project administration, Data curation, Writing – original draft, Software, Funding acquisition, Conceptualization. **Carlos Perez-Vidal:** Validation, Project administration, Data curation, Writing – review & editing, Formal analysis, Conceptualization, Writing – original draft, Supervision, Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT in order to improve language and readability. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Funding

This research has been partially supported by project CPP2021-008593, funded by MICIU/AEI/10.13039/501100011033, and co-financed by the European Union through NextGenerationEU/PRTR. This work was also supported by the Chair of Economic Model Transformation of the Department of Finance, Economy, and Public Administration (*Conselleria d'Hisenda, Economia i Administracio Publica*) of the Valencian Government.

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
AGENCIA ESTATAL DE INVESTIGACIÓN

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

The data supporting this study have been referenced appropriately within the article.

References

[1] M. Polverini, A. Zanchettin, P. Rocco, A constraint-based programming approach for robotic assembly skills implementation, Robot. Comput.-Integr. Manuf. 59 (2019) 69–81, http://dx.doi.org/10.1016/j.rcim.2019.03.005.

[2] L. Gracia, A. Sala, F. Garelli, Robot coordination using task-priority and sliding-mode techniques, Robot. Comput.-Integr. Manuf. 30 (1) (2014) 74–89, http://dx.doi.org/10.1016/j.rcim.2013.08.003.

[3] D. Mateu-Gomez, F. Martínez-Peral, C. Perez-Vidal, Multi-arm trajectory planning for optimal collision-free pick-and-place operations, Technol. 12 (1) (2024) 12, http://dx.doi.org/10.3390/technologies12010012.

[4] J. Borrell, C. Perez-Vidal, J. Segura-Heras, Robotic pick-and-place time optimization: Application to footwear production, IEEE Access 8 (2020) 209428–209440.

[5] J. Borrell, C. Perez-Vidal, J. Segura, Optimization of the pick-and-place sequence of a bimanual collaborative robot in an industrial production line, Int. J. Adv. Manuf. Technol. 130 (2024) 4221–4234.

[6] S. Stavridis, Z. Doulgeri, Bimanual assembly of two parts with relative motion generation and task-related optimization, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2018.

[7] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mosenlechner, D. Pangercic, T. Ruhr, M. Tenorth, Robotic roommates making pancakes, in: 11th IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia, 2011, pp. 529–536, http://dx.doi.org/10.1109/Humanoids.2011.6100855.

[8] V. Azimirad, H. Shorakaei, Dual hierarchical genetic-optimal control: A new global optimal path planning method for robots, J. Manuf. Syst. 33 (1) (2014) 139–148, http://dx.doi.org/10.1016/j.jmsy.2013.09.006.

[9] M. Givehchi, A. Ng, L. Wang, Evolutionary optimization of robotic assembly operation sequencing with collision-free paths, J. Manuf. Syst. 30 (4) (2011) 196–203, http://dx.doi.org/10.1016/j.jmsy.2011.08.003.

[10] K.-C. Ying, P. Pourhejazy, C.-Y. Cheng, Z.-Y. Cai, Deep learning-based optimization for motion planning of dual-arm assembly robots, Comput. Ind. Eng. 160 (2021) 107603, http://dx.doi.org/10.1016/j.cie.2021.107603.

[11] J. Schulman, Y. Duan, J. Ho, Motion planning with sequential convex optimization and convex collision checking, Int. J. Robot. Res. 33 (9) (2014) 1251–1270, http://dx.doi.org/10.1177/0278364914528132.

[12] J. Cascio, M. Karpenko, Q. Gong, P. Sekhavat, I. Ross, Smooth proximity computation for collision-free optimal control of multiple robotic manipulators, in: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 2009, pp. 2452–2457, http://dx.doi.org/10.1109/IROS.2009.5354382.

[13] N. Gafur, G. Kanagalingam, A. Wagner, M. Ruskowski, Dynamic collision and deadlock avoidance for multiple robotic manipulators, IEEE Access 10 (2022) 55766–55781, http://dx.doi.org/10.1109/ACCESS.2022.3176626.

[14] Y. Jiang, Z. Huang, B. Yang, W. Yang, A review of robotic assembly strategies for the full operation procedure: Planning, execution and evaluation, Robot. Comput.-Integr. Manuf. 78 (2022) 102366, http://dx.doi.org/10.1016/j.rcim.2022.102366.

[15] P. Zacharia, N. Aspragathos, Optimal robot task scheduling based on genetic algorithms, Robot. Comput.-Integr. Manuf. 21 (1) (2005) 67–79, http://dx.doi.org/10.1016/j.rcim.2004.04.003.

[16] Z.-Y. Chiu, F. Richter, E. Funk, R. Orosco, M. Yip, Bimanual regrasping for suture needles using reinforcement learning for rapid motion planning, in: Proceedings of the 2021 IEEE International Conference on Robotics and Automation, ICRA, Xi'an, China, 2021, pp. 7737–7743.

[17] S. Shen, D. Zhang, X. Han, C. Jiang, Q. Li, A robust optimization framework for design of robotic system with kinematic and dynamic criteria, Comput. Methods Appl. Mech. Engrg. 423 (2024) 116866, http://dx.doi.org/10.1016/j.cma.2024.116866.

[18] N. Sugimoto, J. Morimoto, Trajectory-model-based reinforcement learning: Application to bimanual humanoid motor learning with a closed-chain constraint, in: Proceedings of the 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Atlanta, GA, USA, 2013, pp. 429–434, http://dx.doi.org/10.1109/HUMANOIDS.2013.7030010.

[19] S. Gu, E. Holly, T. Lillicrap, S. Levine, Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, in: Proceedings of the 2017 IEEE International Conference on Robotics and Automation, ICRA, Singapore, 2017, pp. 3389–3396, http://dx.doi.org/10.1109/ICRA.2017.7989385.

[20] S. Zhang, Q. Xia, M. Chen, S. Cheng, Multi-objective optimal trajectory planning for robotic arms using deep reinforcement learning, Sensors 23 (5974) (2023) http://dx.doi.org/10.3390/s23135974.

[21] J. Peng, D. Wang, J. Zhao, Y. Teng, A. Kimmig, X. Tao, J. Ovtcharova, Meta-learning enhanced adaptive robot control strategy for automated PCB assembly, J. Manuf. Syst. 78 (2025) 46–57, http://dx.doi.org/10.1016/j.jmsy.2024.11.009.

[22] D. Knuth, Sorting and Searching, second ed., Vol. 3, Addison-Wesley Professional, 1998.

[23] L.F. Williams Jr., A modification to the half-interval search (binary search) method, in: Proceedings of the 14th ACM Southeast Conference, 1976, pp. 95–101, http://dx.doi.org/10.1145/503561.503582.

[24] J.L. Bentley, A.C. Yao, An almost optimal algorithm for unbounded searching, Inform. Process. Lett. 5 (3) (1976) 82–87, http://dx.doi.org/10.1016/0020-0190(76)90071-5.

[25] KUKA, Kuka lbr iiwa, 2024, URL: https://www.kuka.com/es-es/productos-servicios/sistemas-de-robot/robot-industrial/lbr-iiwa.

[26] Robotiq, Robotiq, 2024, URL: http://robotiq.com/.

[27] R.O.S. (ROS), ROS, 2024, URL: https://ros.org.