Improved APF strategies for dual-arm local motion planning

Byrne, S., Naeem, W., & Ferguson, S. (2015). Improved APF strategies for dual-arm local motion planning. *Transactions of the Institute of Measurement and Control*, *37*(1), 73-90. https://doi.org/10.1177/0142331214532002

Published in:
Transactions of the Institute of Measurement and Control

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Improved APF strategies for dual-arm local motion planning

Steven Byrne, Wasif Naeem, and Stuart Ferguson

School of Electronics, Electrical Engineering and Computer Science,

Queen's University Belfast, UK

Abstract

Manipulator motion planning is a classic problem in robotics, with a number of complete solutions available for their motion in controlled (industrial) environments. Due to recent technological advances in the field of robotics, there has been a significant development of more complex robots with high fidelity sensors and more computational power. One such example has been a rise in the production of humanoid robots equipped with dual-arm manipulators which require complex motion planning algorithms. Also, the technological advances have resulted in a shift from using manipulators in strictly controlled environments, to investigating the deployment of manipulators in dynamic or unknown environments. As a result, a greater emphasis has been put on the development of local motion planners, which can provide real-time solutions to these problems. Artificial Potential Fields (APF) is one such popular local motion planning technique, which can be applied to manipulator motion planning, however, the basic algorithm is severely prone to local minima problems. Here, two modified APF-based strategies for solving the dual-arm motion planning task in unknown environments are proposed. Both techniques make use of configuration sampling and subgoal selection to assist the APF in avoiding these local minima scenarios. Extensive simulation results are presented to validate the efficacy of the proposed methodology.

1 Introduction

In the field of robotics, the single manipulator motion planning problem has been studied for decades, with planning methods in uncertain environments, such as unknown or dynamic environments, still open areas of research. In recent years, there has also been an upsurge in the study of multi-manipulator motion planning. This is in part due to the development of numerous anthropomorphic robots, such as Justin (Ott et al., 2006),

PR-2 (Bohren et al., 2011) and ARMAR-III (Asfour et al., 2006), which are designed to carry out tasks in a human-like fashion. Like humans, these robots have two manipulators which can be used for carrying out a multitude of tasks. The goal to mimic human-arm behaviour has led to a diverse study of dual-arm systems across a wide range of areas. This is because using a dual-arm robotic system to manipulate objects in a human-like way involves numerous components spanning motion planning, grasping, object manipulation and control theory. Applications for dual-arm systems are theoretically endless but a number of applications have already been considered across domestic and industrial environments as well as for space exploration. Tasks such as laundry folding (Maitin-Shepard et al., 2010), dish-washing (Asfour et al., 2006) and cooking (Zhai et al., 2012) have been investigated in a domestic setting. While in industrial environments, dual-arm system have been used for automotive parts assembly (Park et al., 2006), in addition to satellite servicing within the field of space exploration (Qiu et al., 2009; Xu et al., 2012). This serves as a motivation to develop a novel local motion planning algorithm for dual-arm systems, such as the humanoid robots mentioned above.

The dual-arm motion planning problem is more complex than simply avoiding obstacles and reaching a goal, as is the case with the single manipulator problem. There is a number of different task-based aspects to consider when choosing an appropriate method for dual-arm motion planning problems. For example, the two arms could be working individually on separate tasks while operating in the same workspace. In this case, the arms must avoid each other and obstacles in the environment while completing their respective tasks. Scheduling may need to be addressed in this case, where completion of the two tasks is not possible simultaneously. Another possible motion planning problem with dual-arm systems is the undertaking of a single-task cooperatively by both arms. Here, the arms are working together to complete a single objective, so both arms must avoid the obstacles in the environment while obeying the motion constraints related to their cooperative task.

There are generally two different approaches to these problems; decoupled motion planning and centralised motion planning. With decoupled motion planning, the motion of each manipulator is considered individually. This approach is popular when the two arms are working on separate tasks in the same workspace. The problem is decomposed into two single-manipulator motion planning problems, with the opposite manipulator being represented as an obstacle in the environment, so as to avoid collisions between the two manipulators (Chuang et al., 2006; Curkovic and Jerbic, 2010). Scheduling or prioritisation of the manipulators is necessary to ensure both tasks are successfully completed (LaValle and Hutchinson, 1998). However, decoupled motion planning is not useful for cooperative tasks as the motion of the two arms is highly-coupled in these cases. Thus decomposing the problem in this manner is not possible.

Centralised motion planning involves combining the Configuration Spaces (C-Space) of the two arms and planning a motion within this composite C-Space. This joining of the C-Spaces represents the motion of the two arms in one space, meaning both individual and cooperative tasks can be defined and planned. To plan within this space, graph-based approaches (Fei et al., 2004) and sampling-based methods have been proposed (Tsai and Huang, 2009; Vahrenkamp et al., 2009). The use of closed-chain kinematics is also popular for cooperative tasks (Gharbi et al., 2008; Bolandi and Farhad Ehaei, 2012). When they are applicable, these centralised methods are more reliable than the decoupled approaches. However, to successfully combine the C-Spaces and compute a path, a full knowledge of the environment is required. For this reason, centralised motion planning techniques are only applied to static environments (Fei et al., 2004; Gharbi et al., 2008; Vahrenkamp et al., 2009) or dynamic environments where the obstacles future trajectories are known, i.e. they can be described in the Configuration-Time Space (Tsai and Huang, 2009; Bolandi and Farhad Ehaei, 2012).

The environment in this research is considered to be unknown with the dual-arm system having no prior knowledge of the obstacles locations. In these scenarios, the centralised C-Space methods are not applicable due to the lack of information. Instead, each manipulator must plan its motion separately in a decoupled manner, relying solely on information it collects locally. A popular well-studied method of local motion planning is Artificial Potential Fields (APFs). They are generally applied to point-mass robots where real-time online planning is needed. Using only local information, it is possible to create an attractive field which attracts the robot to its desired goal position, and also create a repulsive field which repels the robot away from any obstacles detected locally. The summation of these two fields gives a total potential field which attracts the robot to the goal while at the same time avoids the obstacles in the environment. It is a simple and fast technique making it ideal for use in uncertain environments, where online planning is needed. However, there are well-documented issues with APF local motion planning.

Local minima issues are prevalent in the use of APFs. These are situations where the attractive and repulsive forces acting on the robot are equal in magnitude but opposite in direction. This causes a zero resultant force on the robot, meaning it will no longer move towards its goal and thus results in a failure of the motion planning task. Many different types of local minima issues have been identified for APF motion planning of point-mass robots (Koren and Borenstein, 1991; Ge and Cui, 2000). Solutions to these issues can be obtained by either modifying the potential field functions (Ge and Cui, 2000; Agirrebeitia et al., 2005; Jia and Wang, 2010) or by employing a secondary planner to assist the APF approach to escape the local minima (Park and Lee, 2003; Zhu et al., 2006, 2009). These solutions generally concentrate on solving a

particular local minimum and cannot guarantee to free the manipulator from all local minima issues.

APFs have also been applied to both single and dual-arm manipulator local motion planning problems (Khatib, 1986; Chuang et al., 2006; Byrne et al., 2012). Similar to APF-motion planning of point-mass robots, the proposed methods for manipulator motion planning are also subject to local minima problems. Previously, a local minimum issue which is unique to the application of APFs to manipulator motion planning has been identified by the authors (Byrne et al., 2012). An online motion planner using configuration sampling and subgoal selection was then proposed to assist the APF in avoiding these scenarios (Byrne et al., 2013). This work will be the basis for the dual-arm strategies proposed in this paper.

In this research, the above online single-manipulator motion planner is modified to solve two separate kinds of decoupled dual-arm motion planning tasks. The first task considered, is the successful decoupled motion planning of two arms working independently within the same workspace. Here, the end-effectors of both manipulators must reach a desired goal position while the manipulators must avoid collisions with each other and any obstacles detected locally in the environment. The second proposed motion planning problem is a general scenario where the two manipulators are undertaking a single task which requires them to meet at a cooperative goal position. This problem has applications in parts assembly (Vahrenkamp et al., 2009) or in passing an object from one arm to another (You et al., 2012).

For each investigated task a standard APF-based approach is first proposed which is shown to have local minima issues. Hence improved local motion planning methods are developed for each task, which are based on the previously proposed online single-manipulator motion planner (Byrne et al., 2013). In order to produce efficient motion planning techniques for the specified dual-arm tasks, issues such as task planning and avoidance of the manipulators are incorporated into the existing online single-manipulator motion planner.

The structure of the paper is as follows: Section 2 defines the setup of the dual-arm system and the standard APF motion planner for applications in unknown environments; Section 3 provides a review of the online single-manipulator motion planner, which is used a basis for the dual-arm motion planning strategies; Section 4 and Section 5 address each of the investigated decoupled dual-arm motion planning tasks and propose motion planners for each task; Section 6 concludes the paper with some final remarks.

2 Dual-Arm System Setup for APF Motion Planning

The design of the dual-arm system, which is used for developing these motion planning strategies, is based on a planar version of a typical humanoid robot's dual-arm system. It consists of two 3-DOF arms, fixed to a base or torso, which can operate in the space in front of them. It is assumed that any obstacles in the environment are unknown to the manipulators at the start of the experiment. Consequently, the manipulators are relying on local motion planning techniques and must be fitted with sensors to detect the obstacles locally. The dual-arm system equipped with IR-sensors and its workspace are described by Figure 1.

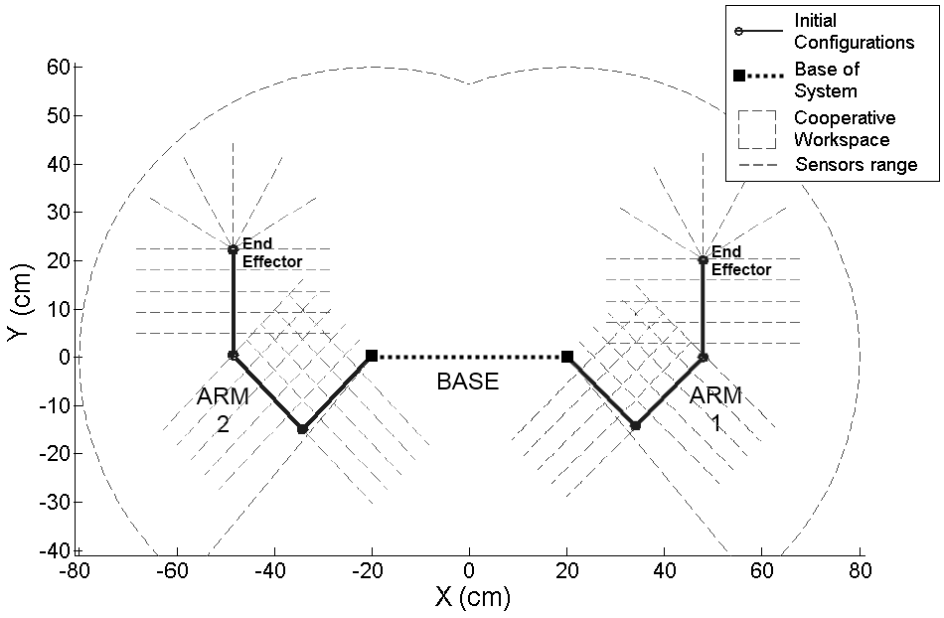


Figure 1: Design of dual-arm system

The chosen local motion planning technique used here is APFs. It is necessary to describe both the attractive and repulsive potential functions which will create the potential fields. The APF approach given here is a modification of Khatib's classic APF functions (Khatib, 1986). The definition of the attractive function is taken directly from this work. An attractive potential field is created at each of the manipulators' end-effectors to attract the manipulators to their respective goal. For each manipulator, the attractive potential function which creates this potential field is given by Eq 1.

$$U_{ATT} = \frac{1}{2} K_p (x_{goal} - x_{ee})^2 \tag{1}$$

where K_p is a positive gain and $x_{goal} - x_{ee}$ is the distance between the end-effector position and the goal.

In single manipulator motion planning, the repulsive field is used to repel the manipulator from any obstacles in the environment. The repulsive potential field is constructed using the sensor data collected at each sensor along the manipulator, given by Eq 3. Subjecting the manipulator to repulsive forces at all these points ensures that the whole manipulator can avoid obstacles while its end-effector moves towards its goal. The points at which these repulsive potential fields are placed are called *points subject to potential* (PSPs). The total repulsive field is then calculated by summing the repulsive fields at each PSP.

In dual-arm motion planning, the repulsive field must also repel the manipulators from each other as well as repelling from obstacles. However, as the IR-sensors do not differentiate between the objects they detect, obstacles and the opposing manipulator will be detected in the same way by the sensors. This means the repulsive function defined by Khatib for single manipulators extends naturally to dual-arm application and repel the manipulators from each other and from obstacles in the environment. The repulsive potential functions are defined in Eqs (2, 3).

$$U_{REP} = \sum_{j} U_{REP_j} \tag{2}$$

$$U_{REP_j} = \begin{cases} \frac{1}{2}\eta \left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)^2 & \text{if } \rho \leq \rho_0 \\ 0 & \text{if } \rho > \rho_0 \end{cases} \tag{3}$$

where η is a positive scaling factor, ρ = distance from the jth sensor to the nearest object in the environment and ρ_0 = limit of influence of an object.

A summation of these attractive and repulsive potential functions produces the total potential field, which attracts the end-effector to its goal, while the repulsions produced at each PSP repel the manipulators from each other and from obstacles. The attractive and repulsive forces which drive the manipulators' motion are obtained by calculating the gradient descent of the potential field. These forces are described in the Cartesian space and must be converted to the manipulator's joint space in order to create motion in the manipulator's joints. This conversion is done using a task-based configuration control, described in Eqs (4, 5) (Patel and Shadpey, 2005). There are three tasks in APF motion planning of manipulators; attracting the end-effectors to their goals, repelling the manipulators away from potential collisions and avoiding singularity configurations. The task-based configuration control is chosen as it minimises the total error across these three tasks, resulting in a regulated motion. The weighting factors W_a, $W_{r_{psp}}$, W_v can be used to prioritise the tasks.

$$q'_{ATT} = (J_e^T W_a J_e + \sum_{psp} J_{c_{psp}}^T W_{r_{psp}} J_{c_{psp}} + W_v)^{-1} (J_e{}^T W_a F_{ATT}{}^d) \tag{4}$$

$$q'_{REP} = (J_e^T W_a J_e + \sum_{psp} J_{c_{psp}}^T W_{r_{psp}} J_{c_{psp}} + W_v)^{-1} (\sum_{psp} J_{c_{psp}}^T W_{r_{psp}} F_{REP_{psp}}{}^d) \tag{5}$$

where J_e is the Jacobian matrix corresponding to the end-effector position and $J_{c_{psp}}$ is the manipulator's Jacobian matrix corresponding to the position of the specified PSP on the manipulator. F_{ATT} and $F_{REP_{psp}}$ are the forces produced by the attractive field at the end-effector and the repulsive APFs, placed at the position of the specified PSP on the manipulator. W_a, $W_{r_{pap}}$are weighting factors for these attractive and repulsive APFs and W_v is the weighting factor for singularity avoidance.

This standard APF approach is well-suited for online local manipulator motion planning however it is subject to local minima issues. One such persistent local minima issue, which is specific to the manipulator motion planning, is the *reacharound local minimum*. This problem was identified and solved for single-manipulators, using an improved online motion planner, based on the integration of goal configuration sampling and subgoal selection to the standard APF approach (Byrne et al., 2013). As this single-manipulator motion planner forms the basis for the proposed dual-arm motion planning techniques, a detailed review of the approach is given in the following section.

3 Review of Online Single-Manipulator Motion Planner

A specific local minimum problem was frequently encountered in obstacle-laden environments using the above APF local motion planner, which impacted greatly on the success of the algorithm. This *reacharound local minimum problem* has been documented in (Kim and Khosla, 1992; Siciliano et al., 2010; Byrne et al., 2012) and a typical example is shown in Figure 2. The issue occurs when the manipulator attempts to naively reach-around obstacles to its goal, via a path which is outside its reachability and thus inevitably arrives at a position where the attractive force to the goal and repulsive force of the obstacles it is reaching around sum to zero. At this point, the manipulator becomes stuck in a local minimum. The cause of this local minimum is an inability of the APF motion planner to identify when the path it is taking around the obstacles has no valid solution. In Byrne et al. (2013), a solution to counteract this shortcoming in the standard APF approach has been proposed, which uses the obstacle's location, acquired through onboard sensors, to identify when the manipulator is attempting an impossible path to the goal and thus alters the

motion of the manipulator to take an alternate feasible path to goal.

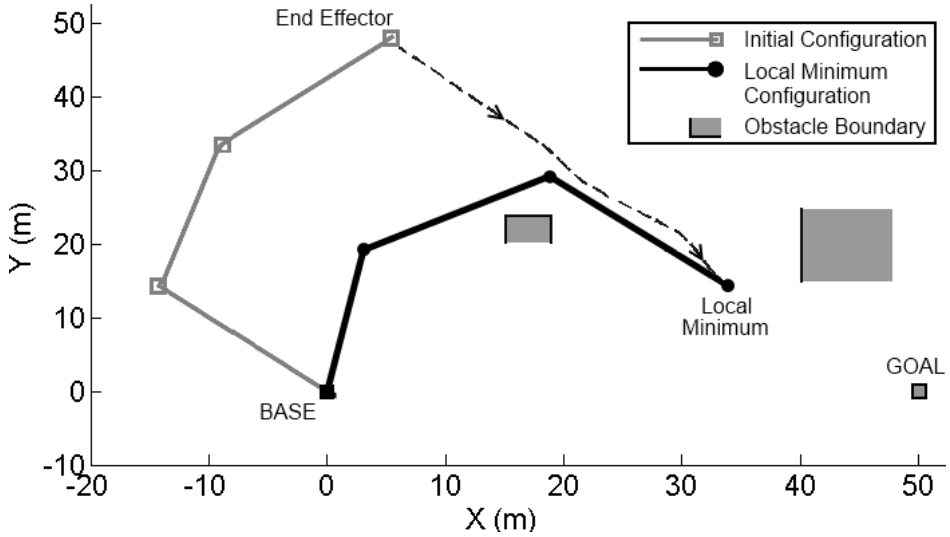


Figure 2: Reacharound Local Minimum Issue

This online motion planner is based on using a combination of goal configuration sampling and subgoal selection in addition to the APF functions described in Section 2. The goal configuration sampling is used to identify feasible configurations which reach the goal, while subgoal selection is used to alter the motion of the manipulator to reach one of these goal configurations. The goal configuration sampling is performed by sampling all configurations which result in the end-effector position being equal to the goal position and are free of collisions with the detected obstacles in the vicinity. The nearest valid goal configuration sample to the current configuration is then chosen as the goal for this motion planning problem.

A subgoal selection algorithm is then used to create a path of subgoals around the obstacles between the current configuration and this goal configuration. The method chosen is based on expanded convex hulls, where a convex hull is first placed around the end-effector, the goal and any obstacles in the way. This creates a path around the obstacles, however the path lies against the boundary of the obstacles. As a result, the convex hull must be expanded to create a collision-free path to the goal.

Figure 3 shows an overview of the online motion planner. Prior to motion, goal configuration sampling and subgoal selection are performed. This gives the manipulator; (a) a valid goal configuration to aim for and (b) an uninterrupted path of subgoals around any obstacles between the initial configuration and this goal configuration. To avoid potential local minima issues, it is necessary that these two conditions remain true throughout the motion.

As the manipulator moves through the unknown environment, it uses the IR-sensors to detect obstacles.

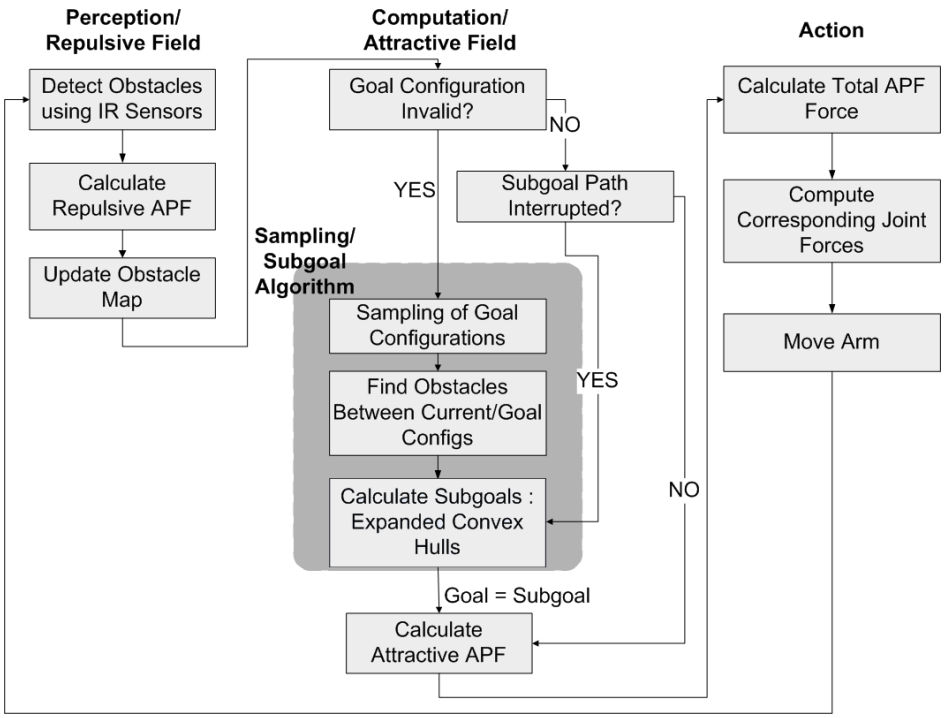


Figure 3: Overview of Online Single-Manipulator Motion Planner

The sensor-data collected is used to calculate the repulsive APF and also to store the obstacle information in a map. The newly detected obstacles are then used to check if the specified conditions have been broken:

- If the goal configuration has become invalid due to a detected obstacle, an alternate goal configuration is chosen from the samples. The subgoal selection process is run to create a path of subgoals from the current configuration and this new goal configuration, avoiding any obstacles in between.
- If the goal configuration remains valid but the newly detected obstacles intersect the path of subgoals, then subgoal selection is re-run to find a new uninterrupted path to the goal configuration

Once the specified conditions are deemed true, the APF functions are used to create potential forces, which are converted to the joint space to move the manipulator. Figure 4 shows how this online motion planner solves the reacharound local minimum problem by replanning the motion online using the local obstacle information.

This online motion planner allows the fast APF to drive the manipulator through the environment, whilst guarding against scenarios where the reacharound local minimum problem will occur, by implementing goal configuration sampling and subgoal selection when necessary. In testing, this technique proved to be more robust than the standard APF approach. As a result, it is chosen as a basis from which to develop improved

local motion planning strategies for the investigated dual-arm motion planning problems which are outlined in Sections 4 and 5.

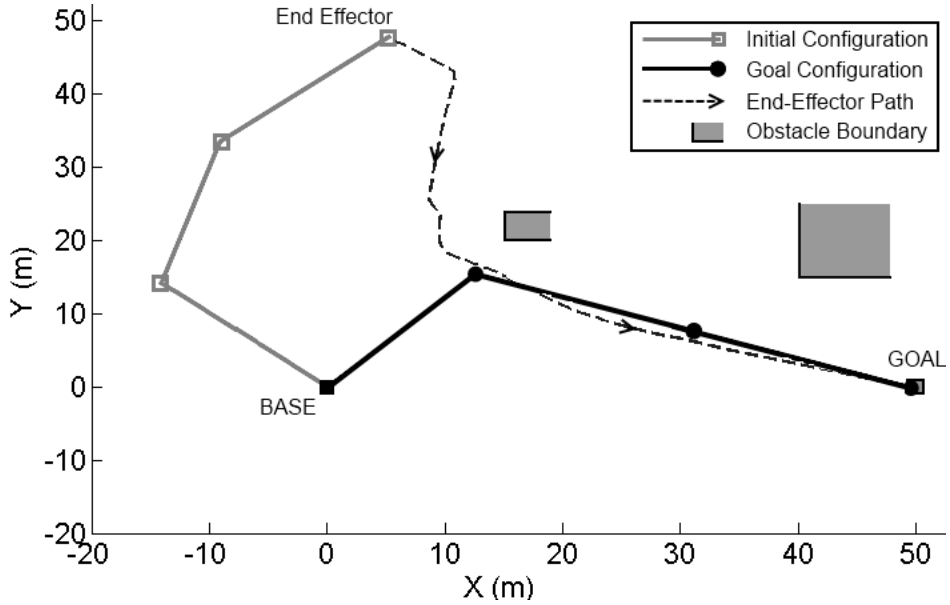


Figure 4: Online Motion Planner solving the reacharound local minimum issue

4 Dual-Arm Motion Planning - Executing Individual Tasks

4.1 Introduction

A strategy for solving the problem of two manipulators operating on separate tasks, within a shared environment is addressed here. This is a well studied problem in the field of manipulator motion planning with both centralised and decoupled motion planning methods proposed for this type of problem (Chuang et al., 2006; Tsai and Huang, 2009; Vahrenkamp et al., 2009; Curkovic and Jerbic, 2010). Centralised methods provide the most complete approach but are not possible for uncertain environments so a decoupled motion planning method is chosen here. Decoupled motion planning allows for the deconstruction of a dual-arm motion planning problem into two single arm motion planning problems and a task-planning problem. The motion of both manipulators can then be considered individually, using a local motion planner for a single arm. A task-planning method is also needed to ensure both tasks are successfully completed.

For local motion planning techniques, such as the standard APF approach outlined in Section 2, high-level intelligent task planners are not always possible due to the lack of information available. For this reason,

prioritisation of the tasks is a popular approach in ensuring the tasks of both manipulators can be completed successfully (Buckley, 1989).

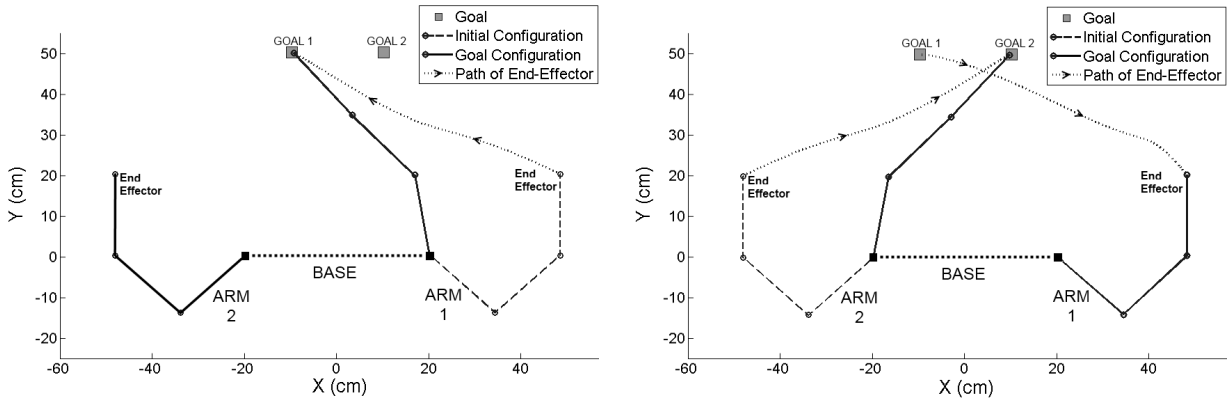
However this APF approach with task priorisation is not an ideal solution to executing individual dual-arm tasks. Firstly, the tasks are not run concurrently with the task prioritisation approach, causing the efficiency of the motion planner to suffer. Also, the standard APF motion planner is subject to local minimum issues which limit the robustness of this approach. To improve both the efficiency and robustness an alternate approach is proposed, with the online motion planner outlined in Section 3 replacing the APF motion planner. Unlike the standard APF approach, the online motion planner collects information, which can be used intelligently to plan better motion than the naive standard APF. Thus, a more efficient high-level task planner can be incorporated into this motion planner, which allows simultaneous execution of the dual-arm tasks.

In this section, both of the above approaches are detailed; the standard APF with task prioritisation and the proposed online motion planner with a high-level task planner. A comparison is drawn between them using Monte Carlo simulations, which shows the improvement made by the proposed online motion planning method, both in terms of efficiency and robustness.

4.2 APF approach with Task Prioritisation

The standard APF approach outlined in Section 2, is used here as the motion planner for this decoupled dual-arm motion planning task. However with dual-arm motion planning, where the manipulators are working on individual tasks, motion planning of each arm is not the only problem which needs solving. Task planning is also necessary, as it may not be possible for both tasks to be completed simultaneously, due to the manipulators' paths intersecting, leading to local minima issues. High-level task planning can easily be incorporated into a centralised motion planning method to allow for simultaneous execution when possible. This is due to sufficient information known *a priori*. However for online decentralised motion planning, such as this APF approach, there is no known information so this is not possible. Simultaneous execution of tasks using APF could lead to local minimum and collision issues. For this reason, it is chosen to prioritise the tasks and run them separately.

This task prioritisation is achieved by establishing a master-slave relationship between the two manipulators within the APF motion planner. An example of this relationship is shown in Figure 5. Initially, Arm 1 is assigned as the master manipulator and moves towards its goal, while Arm 2, the slave manipulator, maintains its position at the initial docking configuration, to allow Arm 1 to complete its mission. After



(a) Arm 1 selected as master manipulator

(b) Arm 1 retracts so Arm 2 can reach goal

Figure 5: Example of Task Prioritisation

that the relationship is reversed where, Arm 1 is set as the slave manipulator and retracts to the docking configuration. Arm 2 becomes the master and is free to proceed unimpeded towards its goal, as shown in Figure 5b.

This master-slave relationship is implemented within the attractive functions of the APF motion planner as defined by Eqs (6, 7). The repulsive APF functions remain as described by Eqs (2, 3). An overview of this task prioritisation algorithm is given in Figure 6.

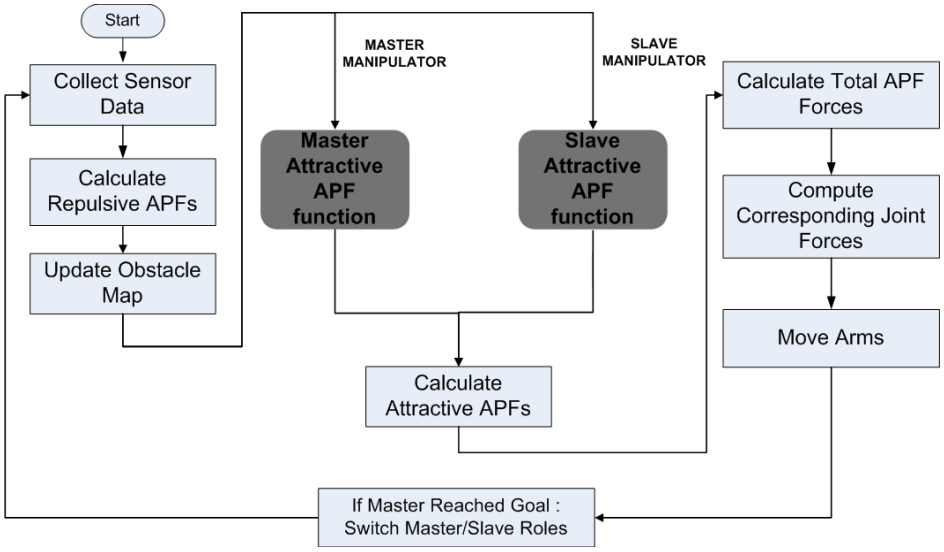


Figure 6: Overview of Task Prioritation Decoupled Algorithm

$$U_{ATT_{MASTER}} = \sum_{psp} \frac{1}{2} K_p (x_{ee} - x_g)^2 \tag{6}$$

where K_p is a positive gain and $x_{ee} - x_g$ is the distance between the end-effector's current position and its desired goal position.

$$U_{ATT_{SLAVE}} = \begin{cases} \sum_{psp} \frac{1}{2} K_p (x_{psp} - x_{dock_{psp}})^2 & \text{if } q \neq q_{dock} \\ 0 & \text{if } q = q_{dock} \end{cases} \tag{7}$$

where K_p is a positive gain and $x_{psp} - x_{dock_{psp}}$ is the distance between the current position and the docking position for each *point-subject-to-point*.

The APF method outlined here is a well-established approach to executing individual tasks within a shared workspace. However due to the poor efficiency of the task prioritisation technique and the local minima issues associated with the standard APF, an alternate approach is proposed, with the aim to increase the success rate of the motion planning algorithm while also permitting simultaneous execution of the tasks when possible.

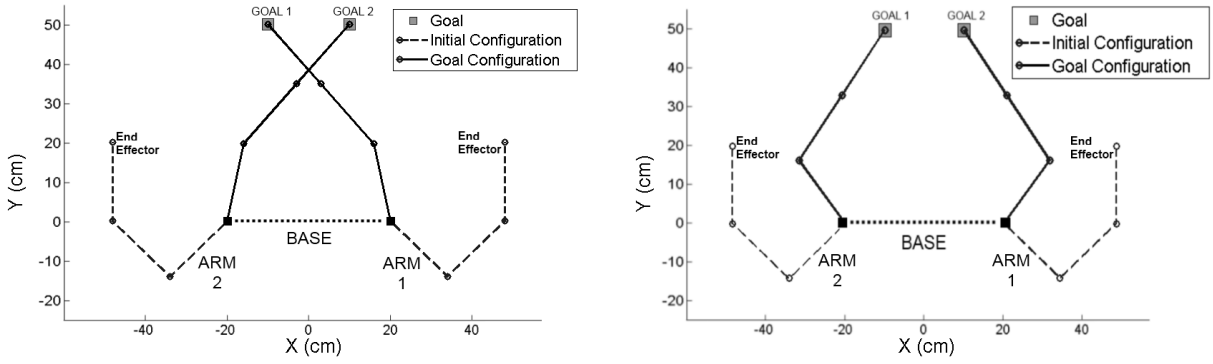
4.3 Online Motion Planner with an Improved Task Planner

The proposed algorithm for executing individual tasks for this dual-arm system uses more effective motion planning and task planning methods. The APF motion planner is replaced by the online single-manipulator motion planner outlined in Section 3. This motion planner uses a higher-level of knowledge and collects obstacle data as the manipulator progresses through the environment. Due to this fact, it is possible to develop a more intelligent task handling strategy to accompany the online motion planner. This will replace the in-efficient task prioritisation technique. The proposal for this new task planner is to use a three-stage process to handle the scheduling of tasks. The first two stages are performed at a high-level, to determine if there is a possibility to complete both tasks simultaneously, avoiding any need for scheduling. The third stage is only used when simultaneous execution of the tasks is deemed impossible and prioritisation is necessary. In the following sections, the three stages of the scheduling process are described.

4.3.1 Stage 1: Reassigning of Goals

In this proposed system, it is assumed that the manipulators in the dual-arm system are identical, like human arms, and thus can perform identical tasks. This assumption can be utilised to remove a large number of scheduling issues. If there are two tasks to be completed, it is possible to assign the tasks to the two arms in a manner that will produce the simplest motion, as both arms can perform identical tasks.

The first stage in the scheduling handling process is to reassign the two goals prior to motion, to remove the possibility of scheduling issues. Figure 7a shows an example scenario where the paths of the two arms must cross to complete their individual tasks. If these tasks were run simultaneously using the standard APF approach, a local minimum or collision would occur. However by reassigning the goals, the scheduling issue is removed. Figure 7b shows the solution to this problem, where Arm 1 and Arm 2 are able to complete the tasks simultaneously and free of interference.



(a) Scheduling problem

(b) Reassigned goals

Figure 7: Solving scheduling issue by the reassignment of goals

The method for assigning goals is simple. Each possible assignment is scored using the Euclidean distance from the initial end-effector positions, $EE1$ and $EE2$, to the goal positions $G1$ and $G2$. The minimum score is then chosen as outlined by Algorithm 1.

It is possible that a goal may not be reachable by one of the manipulators, either due to obstacles in the way or due to it being outside the manipulator's reachable workspace. In this case, the assignment with this manipulator-goal pairing will receive an infinitely high score as it is impossible to complete. Although goal reassignment solves a large amount of scheduling issues, further measures must be taken to remove the remainder of cases which occur due to the presence of obstacles.

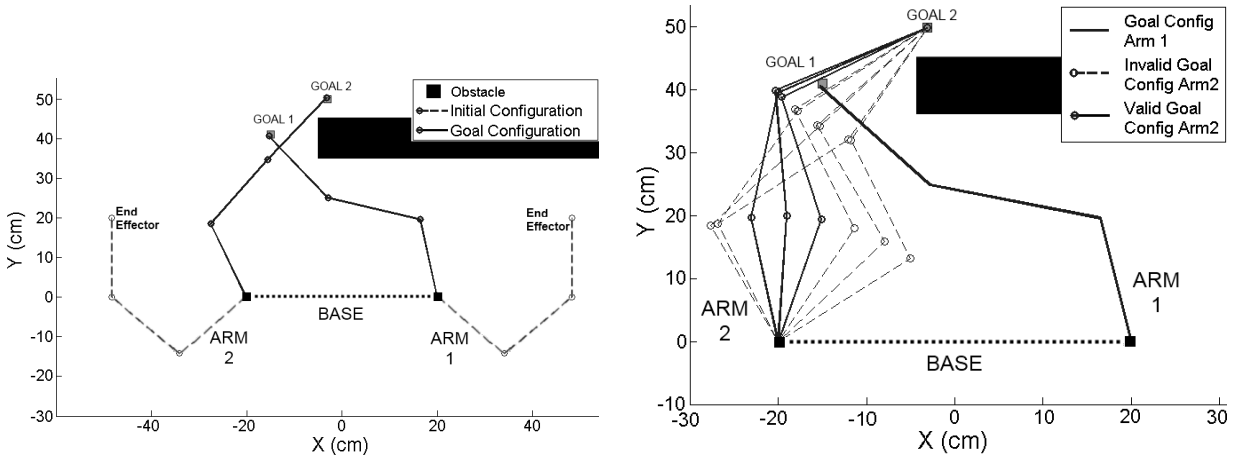
Algorithm 1 Scoring method

Score 1 = $|\overline{EE1\ G1}| + |\overline{EE2\ G2}|$;
Score 2 = $|\overline{EE1\ G2}| + |\overline{EE2\ G1}|$;
Chosen Assignment: min(Score1, Score2);

4.3.2 Stage 2: Cooperative Goal Configuration Sampling

In some cases, due to the position of obstacles in the environment, the only possible assignment of goals will result in the paths of the manipulators crossing. Such a case is shown in Figure 8a. Arm 1 is only able to

reach one of the goals due to the presence of the obstacle. This means the assignment of goals shown in Figure 8a is the only viable option and contains a scheduling issue. To handle scheduling in this scenario, goal configuration sampling is used. Goal configuration sampling is performed for both manipulators, giving a range of solutions to both of the tasks. Where it is possible to obtain a pair of non-conflicting goal configurations, these configurations are set as the desired goals for each manipulator, as depicted in Figure 8b. The single arm motion planner then simultaneously navigates each manipulator to these non-conflicting solutions. Using this method, the motion of the manipulators can be altered to avoid the scheduling issue and complete the specified tasks simultaneously.



(a) Scheduling problem

(b) Non-conflicting goal configurations

Figure 8: Solving scheduling issue using Goal Configuration Sampling

4.3.3 Stage 3: Task Prioritisation

In the rare case, that the goals cannot be assigned in a manner that includes a valid pair of goal configurations, the tasks cannot be completed simultaneously and a return to the original task prioritisation is needed. A master/slave relationship is then invoked to schedule motions so both tasks can be completed. This is performed at a low-level within the APF functions, as described in Section 4.2.

4.3.4 Overview of the High-Level Task Planner

Now that the high-level task planning strategy has been outlined, it is necessary to show how it is incorporated into the design of the online motion planning algorithm. At the initialisation of the online motion planner, the goals are assigned and a cooperative goal configuration pair is chosen from by sampling. The algorithm then proceeds in the same manner as the single-manipulator motion planning algorithm. Sensor data is used

to calculate the repulsive fields for each manipulator and to update the obstacle map, while the subgoal selection algorithm produces a path of subgoal for each manipulator. These subgoals are used to calculate the attractive APFs and move the manipulators to their respective goals.

In single-manipulator motion planning, goal configurations are updated online if a newly discovered obstacle renders the current goal configuration invalid. In this dual-arm algorithm, when one of the goal configurations is deemed invalid by an uncovered obstacle, the high-level task planner chooses a new cooperative pair so that the motion of both manipulators can continue without a scheduling issue. If no valid cooperative pair exists, the goals may be reassigned online to find new non-interfering paths for the manipulators. This high-level task planner is summarised in Figure 9.

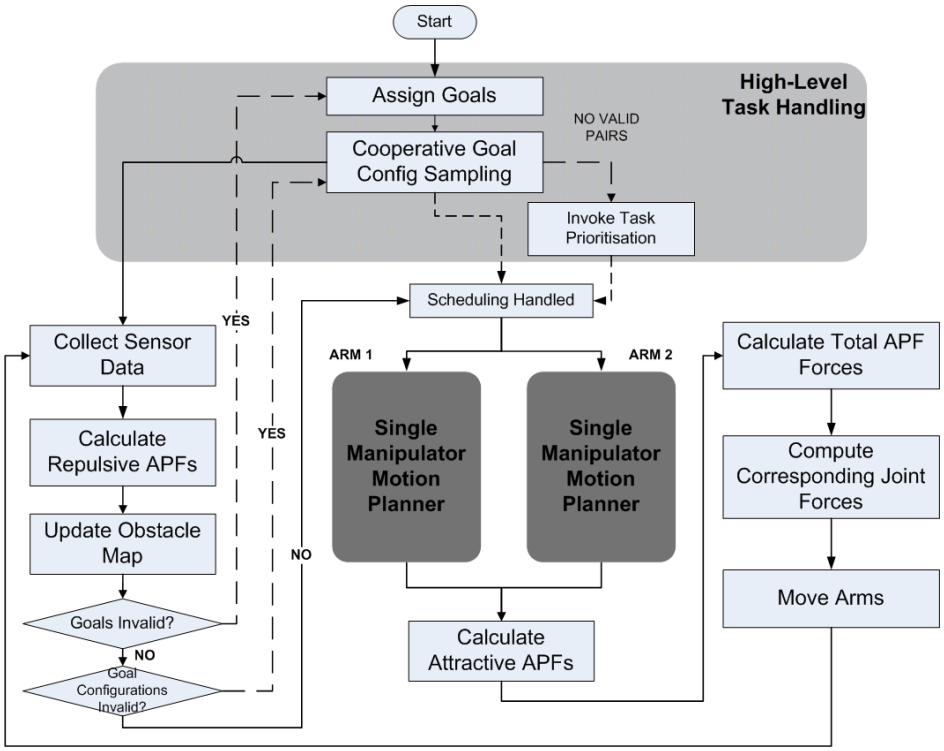


Figure 9: Flowchart of the High-Level Task Planner

In the case where it is not possible to obtain a valid cooperative pair of goal configurations, a simultaneous solution can not be produced and task prioritisation is invoked. The master manipulator continues towards its goal, while the slave manipulator retracts. Once the master manipulator reaches its goal, the manipulators reverse roles and the second arm can complete its task.

Method	Success Rate
APF with Task Prioritisation	60.73%
Online Motion Planner with High-Level Task Planner	87.48%

Table 1: Robustness of each motion planning/task planning technique

4.4 Monte Carlo Analysis

In order to test this proposed online motion planning algorithm for completing two individual motion planning tasks within a shared 2-D workspace, Monte Carlo simulations were performed. In total, $10,000$ scenarios were considered under the following conditions:

- The chosen environment is outlined in Figure 1. Each manipulator consists of links of length 20 cm, with an allowable angular range of $-\pi$ rad $< \theta < \pi$ rad. The manipulators are deployed in the outlined shared workspace, starting from the initial docking configuration.
- 2-4 obstacles and 2 goal positions were placed randomly within the shared workspace.
- It is assumed the obstacles were completely unknown at the beginning of the simulation.
- The sizes of the obstacles was chosen at random within the range $1cm^2 - 400cm^2$.
- All scenarios which were considered, contained a valid solution to the motion planning problem.

The purpose of the Monte Carlo simulations was to investigate the improvements made by both the proposed online motion planner and the high-level task planner. Firstly, the robustness of the online motion planner was tested to see if it produced less local minima issues than the standard APF approach. The new high-level task planning technique was also tested to see if it could successfully execute the individual tasks simultaneously. The effect this simultaneous execution had in terms of the time to converge to a solution and the overall path travelled by the manipulators was also investigated.

Table 1 shows the success rates for both proposed approaches on the tested scenarios. The standard APF motion planner with task prioritisation produced a very low success rate of 60.73%. This is due to the local minima issues associated with the standard APF method in obstacle laden environments. When this standard APF method is replaced by the proposed online motion planner a much higher success rate of 87.48%, is obtained. This proves that the proposed online motion planner is more robust and less susceptible to local minima issues than the standard APF approach.

When testing the proposed high-level task planning technique, each of the tested Monte Carlo simulations could be categorised by the task handling method needed to solve them. This is represented in Table 2.

	No Scheduling Issue	Assigning Goals	Cooperative GC Sampling	Task Prioritisation
% of Cases	53.9%	44.6%	1.2%	0.3%

Table 2: Methods used by the high-level task planner to solve scheduling issues

53.9% of cases had no scheduling issues and could be solved without the need for any high-level input, whereas 44.6% were solved by reassigning the goals to remove the scheduling issue. The remaining 1.5% of cases could not be solved by reassigning goals. In 1.2% of these cases, the necessity for scheduling could be removed by the selection of a valid cooperative goal configuration pair. This means that the proposed high-level task planner was able to ensure that 99.7% of all problems could be completed concurrently. For the remaining 0.3% of problems, this was not possible and a reversion to task prioritisation was necessary.

Method	Success Rate	Execution Time	Average Manipulator Path Length
Task Prioritisation	88.30%	33.47 s	32.49 cm
High-Level Task Planner	87.48%	25.12 s	23.41 cm

Table 3: Comparision of the task planning techniques

The motivation to use this high-level task planner over basic task prioritisation was to improve the execution time of the tasks by permitting the tasks to run concurrently when possible. This improvement is reflected in the results gathered from the Monte Carlo simulations. Table 3 shows the comparison between running the proposed online motion planner with task prioritisation and running it with the improved high-level task planner. The high-level task planner produces much quicker solutions, with shorter overall distances travelled by each manipulator. This is due to the fact that the manipulators were not required to wait to complete their task or needed to retract to their original docking position. The unsuccessful cases are due to the local minima and collision issues inherited from the previously proposed single-manipulator motion planning algorithm. It should be noted there is a slight drop in accuracy when using the high-level task planner. This is due to a small number of errors which occurred with the manipulators working simultaneously in tight spaces.

In this section, the proposed online motion planner for single-manipulator systems has been successfully applied to solve the problem of two manipulators working independently in a shared unknown environment. The proposed method has proved to be more robust than the standard APF approach and also facilitated the development of a high-level task planner, which is more efficient than the classic task prioritisation approach. The following section investigates an alternate type of decoupled dual-arm motion planning problem, where cooperation between the manipulators is required. Again a standard APF approach is given, along an

improved motion planning strategy based on the proposed online motion planner outlined in Section 3.

5 Dual-Arm Motion Planning - Single Task with a Cooperative Goal

5.1 Introduction

Consider the case of a cooperative task of passing an object from one manipulator to another (Vahrenkamp et al., 2009), or the assembly of two parts by autonomous manipulators (You et al., 2012). Clearly this would require two manipulators working cooperatively with some mutual goal position for the end-effectors. This is the next motion planning problem to be considered for the proposed dual-arm system; planning the motion of both manipulators to meet at a mutual goal, in order to perform a single cooperative task.

Given a known environment, this problem could be solved using a centralised dual-arm motion planner. Complete grid-based methods have been established for dual-arm systems with few degrees of freedoms (Koga and Latombe, 1992) and sampling-based methods have also been proposed for dual-arm systems with greater degrees of freedom (Vahrenkamp et al., 2009). In uncertain environments, such as the one investigated here, a local APF-based approach could be used. In this section, a standard APF approach is first employed and its limitations are exposed. An alternate method, based on the online motion planner outlined in Section 3, is then proposed to address the limitations of the standard APF approach.

5.2 APF approach

Classically, in the application of APFs, the desired goal position of the robot is set as the pole of the attractive field. The attractive APF function is then used to guide the robot to this position, while the repulsive fields are used to avoid the obstacles in the environment. However, in this type of cooperative problem there is no definite goal position for each manipulator. Instead, the goal of this motion planning problem is to have the manipulators' end-effectors meet each other at any feasible point in space. The lack of a definite goal position for each manipulator thus requires a modification to the standard APF approach.

This modification is done by simply setting the attractive APF functions to attract the manipulators to each other, rather than to a set position in the workspace. A direct line from one end-effector to the other is established, and the goal position of each manipulator is set as the midpoint of this line, as shown in Figure 10. As the manipulators move, this direct line between the end-effectors and the goal position is updated to

correspond with the updated positions of the end-effectors.

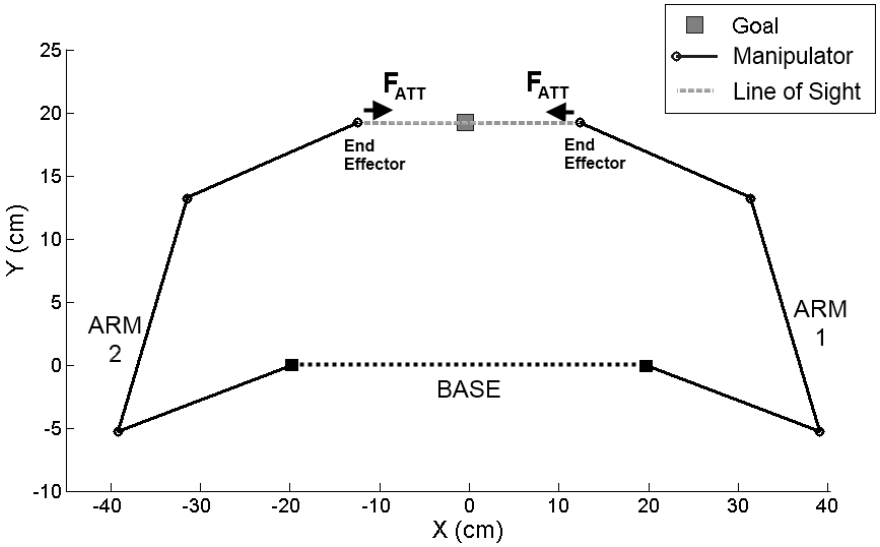


Figure 10: Attractive forces between the two end-effectors

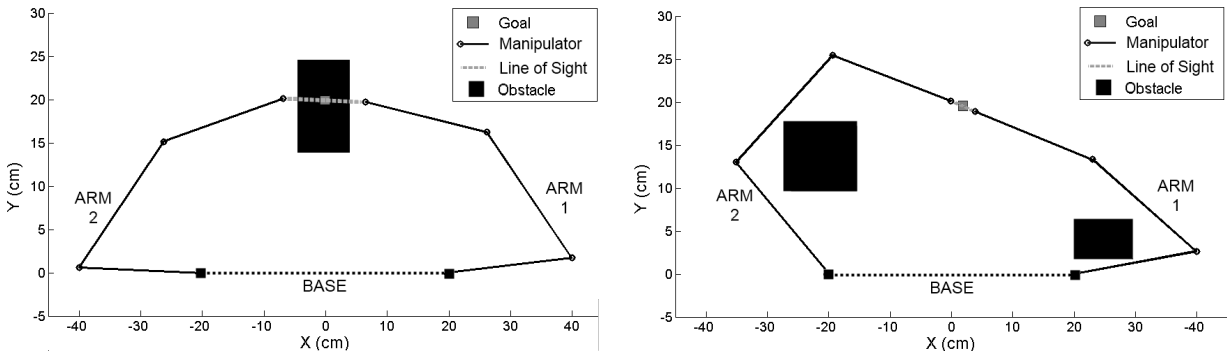
$$U_{ATT} = \frac{1}{2} K_p (x - x_g)^2 \tag{8}$$

where K_p is a positive gain and $x - x_g$ is the distance between the end-effector and its goal, which is the midpoint of the direct line from one end-effector to the other, i.e. $x_g = \frac{x_{EE1} - x_{EE2}}{2}$.

The modified attractive APF function is given by Eq 8 and will result in the movement of the manipulators' end-effectors to each other, in order to meet at a mutual goal position. However as this approach is based solely on the standard APF method, it is subject to a variety of local minima issues. These generally occur if there is an obstacle intersecting the direct line between the two end-effectors or due to the reacharound local minimum. Examples of these possible local minima issues are depicted in Figure 11. The presence of these issues constrains the completeness of this approach. Due to the limitations, it is chosen instead to use the online motion planner described in Section 3, to produce a more robust solution to this type of cooperative dual-arm motion planning task.

5.3 Online Motion Planner for reaching a cooperative goal

As mentioned in the previous subsection, in this type of cooperative motion planning problem there is no definite goal position in the workspace for the manipulators to aim for. The goal is simply to find any mutually reachable point in the workspace. However, the proposed online motion planner, outlined in Section 3, is



(a) Collinear local minimum

(b) Reacharound local minimum

Figure 11: Local Minimum issues with APF approach

dependant on the existence of a definite goal position. For this reason, it is essential to modify the approach so that a cooperative goal position can be chosen, which can be successfully reached by both manipulators.

The selection of a cooperative goal position is done using a goal space sampling method. The goal space is defined as the intersections of the two configuration spaces, i.e. the space in which a mutual goal can exist. This goal space region is then sampled to obtain a set of possible mutual goal positions. Figure 12 shows the goal space along with a number of possible goal positions chosen by the proposed method.

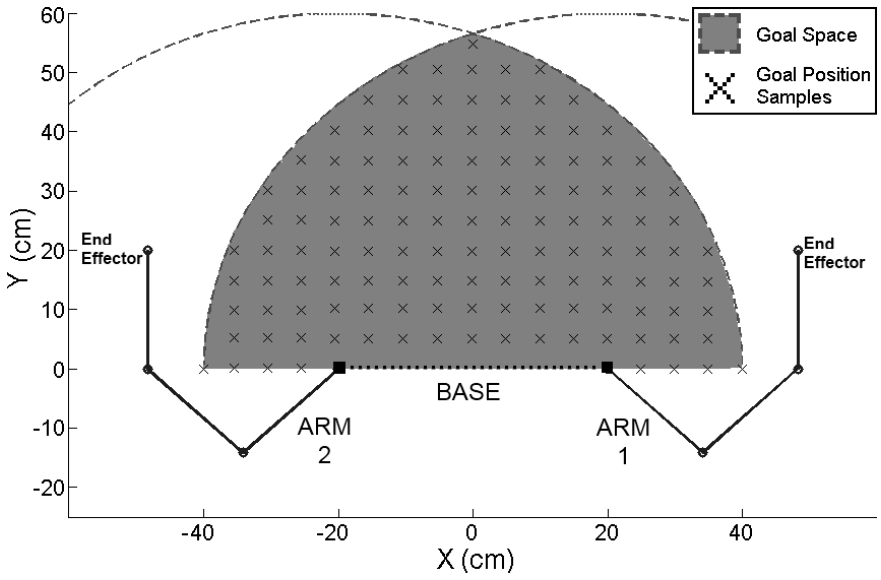


Figure 12: Sampling the Goal Space

To determine which one of these possible goal positions is a viable option, sampling is performed at each possible goal position. If a valid pair of goal configurations is found, this goal position is deemed to be a suitable mutually reachable goal. In an attempt to find the optimal solution, the goal configuration

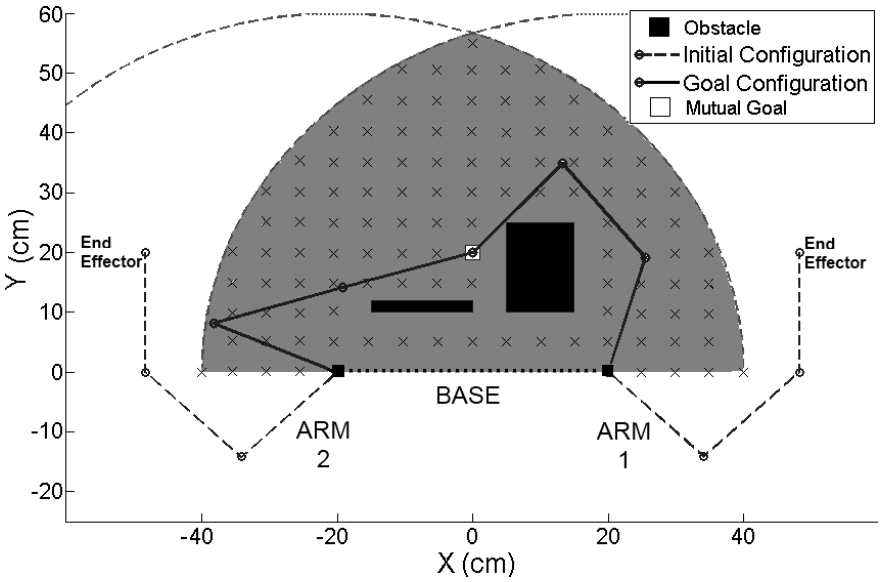


Figure 13: Selecting a mutually reachable goal

sampling process starts with the goal position which minimises the distance travelled by both manipulators. This point is the midpoint of the direct line from one manipulator to the other, as in Figure 10. If this position is not valid, its neighbouring sample positions are checked. The process continues recursively until a valid goal position is found. Figure 13 shows a valid pair of goal configurations found by goal configuration sampling, rendering the tested goal position as a valid mutual goal.

Having found a valid mutual goal for the manipulators, the decoupled motion planning can be performed using the online motion planner outlined in Section 3. It should be noted that this is a single task which must be completed simultaneously by both manipulators so no task planner is required. Similar to the online motion planner, the process of selecting a valid mutual goal must be equipped to deal with the discovery of new obstacles online. As new obstacle information is acquired, it is possible that the goal position may become unreachable by one or both of the manipulators. In this scenario, a new goal position is sought, starting with the neighbours of the old goal position. Figure 14 shows an overview of this online motion planner for reaching a cooperative goal.

5.4 Performance-Enhancing Modifications

The modification outlined in the previous subsection allows for the use of the proposed online motion planner for solving this cooperative motion planning problem. However, the performance is not as efficient as desired. In certain scenarios, the algorithm can perform slowly, causing motion to cease momentarily while a new

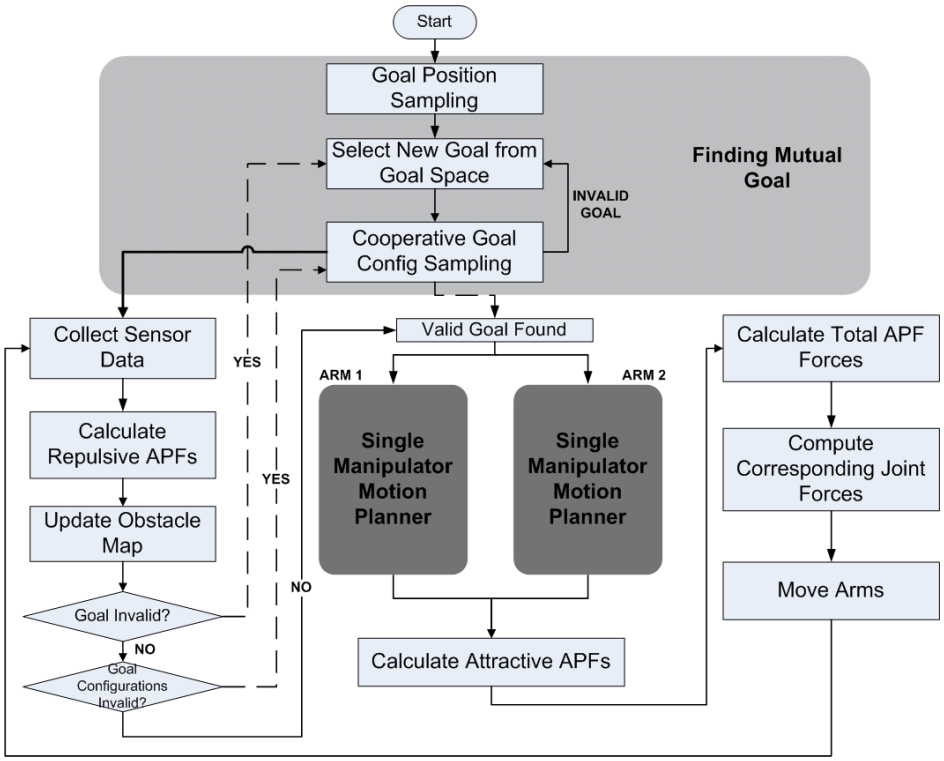


Figure 14: Flowchart of the online motion planner for reaching a cooperative goal

cooperative goal calculated. It is also liable to select goal positions which produce slow convergence to the mutual goal. To combat these performance issues, further modifications are proposed to enhance the overall speed of the algorithm, by reducing the time taken to calculate valid goal positions and by shortening the time taken for the manipulators to reach each other.

5.4.1 Modification 1: Faster goal position determination using a lower goal configuration sampling rate

The most time-consuming process involved in this online motion planning algorithm is the determination of a valid goal position. The method requires sampling to be performed at every possible position within the goal space until a valid pair of goal configurations is found. Although goal configuration sampling is a relatively fast process, it is being applied to two arms and across a number of sample goals here. If there is no valid goal position near the starting point, the process can become time-consuming, as a large portion of the goal space needs to be tested before finding a valid goal. This means the robot's motion can remain static for a significant fraction of time while a valid goal position is found.

A two stage goal configuration sampling approach is proposed to speed up the process. In the first stage,

a lower sampling rate is used however if it fails to find a solution, the sampling rate is increased to find the missed solution. Running goal configuration sampling at a lower rate reduces the computational cost of the goal configuration sampling process, at the expensive of reducing the possibility of finding a valid pair of goal configurations. Through repeated simulations, it could be concluded that the selection of a lower sampling rate does not impact greatly on the accuracy of finding a valid goal position. Due to the large number of possible goal positions, it is highly unlikely that a valid solution will not be found, even using the lower sampling rate.

On the rare occasion that the entire goal space has been searched without finding a valid pair of goal configurations, the process of finding a valid goal position is repeated using a higher sampling rate. The increased number of samples involved will increase the chances of finding a valid goal position. This two stage approach maintains the accuracy in determining valid goal positions, while the use of an initial lower sampling rate allows for these positions to be found more efficiently, in a large search space.

5.4.2 Modification 2: Faster goal position determination by updating the goal space

A second modification was identified to ensure faster calculation of the cooperative goal positions. When used in cooperation with the lower goal configuration sampling rate, this further enhances the performance of the goal position finding process. As obstacles are discovered within the workspace, they can obstruct the motion of the manipulators, thus potentially reducing the reachable workspace of each manipulator. As a result, the intersection of the reachable workspaces, i.e. the goal space, can shrink as obstacles are detected. This can be seen in Figures 15 and 16. Figure 15 shows the goal space at the start of the task. However in Figure 16, it can be seen that an obstacle has been discovered and is restricting the motion of Arm 1. This reduces Arm 1's reachable workspace and consequently, the cooperative goal space is greatly reduced from the original goal space (Figure 15), rendering a large number of the goal samples as invalid solutions.

If the goal space is not updated as obstacles are discovered, these unreachable samples would still be investigated by the algorithm as a possible mutual goal. This would be computationally burdensome, unnecessarily slowing down the process of finding a cooperative goal. However by continually updating the goal space as obstacles are discovered, the goal configuration sampling process can be bypassed for these redundant samples and the algorithm can run more efficiently, resulting in faster calculations of cooperative goals.

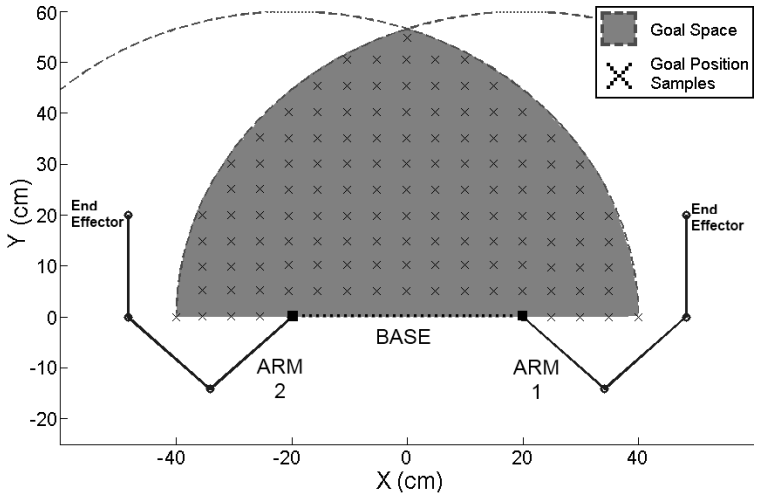


Figure 15: Original goal space

5.4.3 Modification 3: Reverting to naive APF when safe

Previously, in Section 5.2, it was stated that this problem could be solved via the standard APF method. This method is known to be problematic due to the local minima issues which occur when traversing obstacles in the environment. However the standard APF approach is very fast and attracts the end-effectors to each other by the shortest route possible. This means when the method succeeds, it provides a faster, more direct motion to the goal than using the online motion planner.

For this reason, it is deemed advantageous to use APF when there is no known obstacles between the two manipulators. Once an obstacle is detected in the intermediate region, the more complex online motion planner is invoked to ensure potential local minima issues are avoided.

5.4.4 Modification 4: Faster termination by combining manipulators' paths

In most cases, due to obstacles in their paths, it is likely that both manipulators may not reach the mutual goal at the same time. This means that one manipulator will reach the goal and must wait for the other to arrive. This reduces the efficiency of the solution.

Within the online motion planner, a subgoal selection algorithm is used to plot a path of subgoals from the end-effector's current position to the desired goal. Thus each manipulator has a set of subgoals $(SG_1, ...SG_N)$ which define the path of its motion, from the end-effector's current position (EE), around any obstacles towards the mutual goal position (MG). This path is defined by these points; $[EE \rightarrow SG_1 \rightarrow ... \rightarrow SG_N \rightarrow MG]$. Once the manipulator travels this path and reaches the mutual goal position, its motion

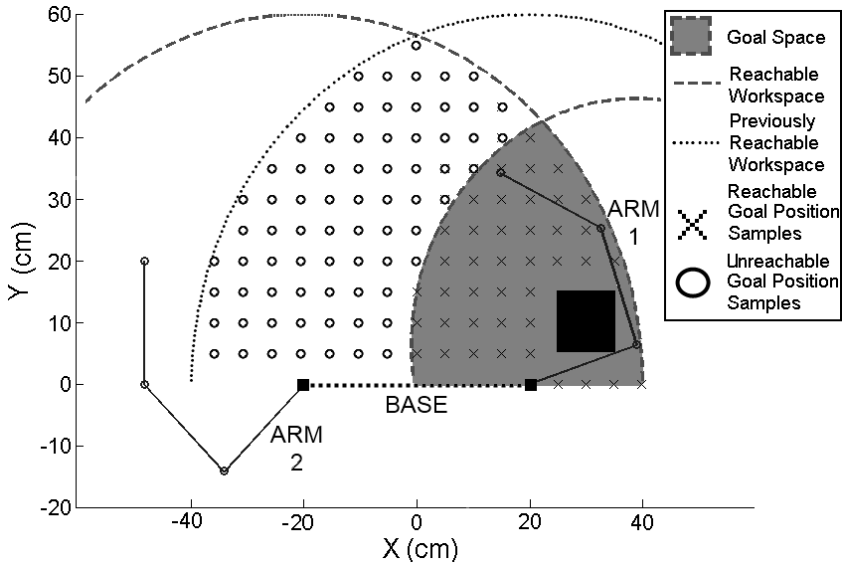


Figure 16: Reduced goal space

terminates. However the goal of this algorithm is to have the end-effectors meet each other. The chosen mutual goal only exists to facilitate this intercept. For this reason, the desired path of the manipulator should not stop when it reaches the mutual goal, it should continue on in an attempt to intercept the end-effector of the other manipulator.

Algorithm 2 Combining paths of manipulators into a single cooperative path

Existing Algorithm :

Path of ARM 1 = $[EE1 \rightarrow SG1_1 \rightarrow SG1_2 \rightarrow ... \rightarrow SG1_M \rightarrow MG]$

Path of ARM 2 = $[EE2 \rightarrow SG2_1 \rightarrow SG2_2 \rightarrow ... \rightarrow SG2_N \rightarrow MG]$

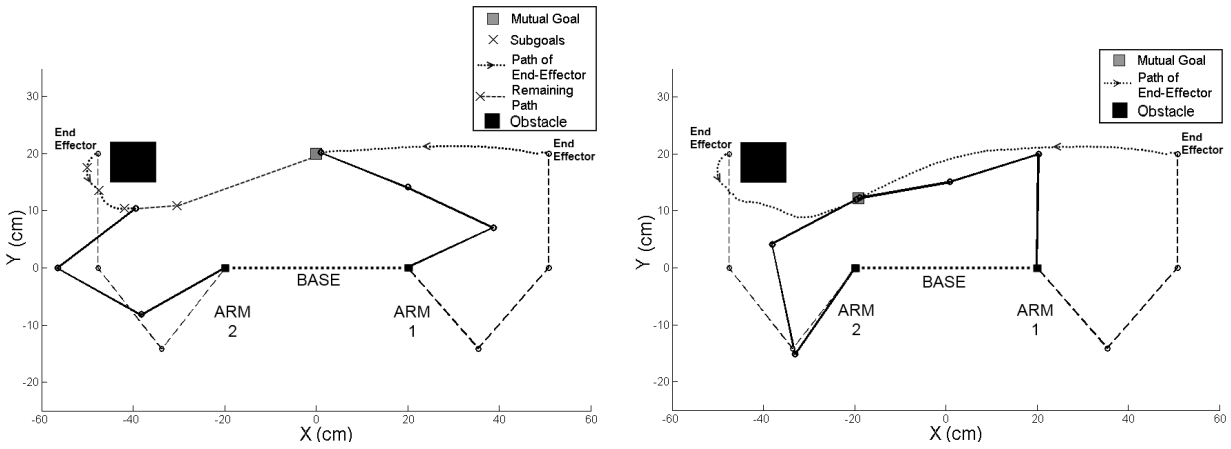
Modification :

Path of ARM 1 = $[EE1 \rightarrow SG1_1 \rightarrow SG1_2 \rightarrow ... \rightarrow SG1_M \rightarrow MG \rightarrow SG2_N \rightarrow SG2_{N-1} \rightarrow ... \rightarrow SG2_1 \rightarrow EE2]$

Path of ARM 2 = $[EE2 \rightarrow SG2_1 \rightarrow SG2_2 \rightarrow ... \rightarrow SG2_N \rightarrow MG \rightarrow SG1_M \rightarrow SG1_{M-1} \rightarrow ... \rightarrow SG1_1 \rightarrow EE1]$

In the existing algorithm two individual motions are described; a path from the end-effector of Arm 1 to the mutual goal and a path from the end-effector of Arm 2 to the mutual goal. However by following the logic expressed here, an alternate approach is proposed to redefine this motion. Instead of two decoupled paths, from each end-effector to the mutual goal, the two paths are combined into a single cooperative path between the two end-effectors. Now, if one manipulator reaches the mutual goal before the other, its motion does not terminate, instead it will continue down the cooperative path to meet the other manipulator, as is shown in Figure 17.

This modification is outlined by Algorithm 2. Consider if Arm 1 completes its path of subgoals, $[EE1 \rightarrow$



(a) Arm 1 waiting for Arm 2 to reach the mutual goal

(b) Arm 1 continues along combined path to meet Arm 2

Figure 17: Combining the subgoal paths

$SG1_1 \rightarrow SG1_2 \rightarrow ... \rightarrow SG1_M \rightarrow MG]$, before Arm 2 reaches the mutual goal. Using this improved method, Arm 1 does not terminate at the mutual goal (MG). Instead it will continue its motion down the path of subgoals of Arm 2, $[MG \rightarrow SG2_N \rightarrow SG2_{N-1} \rightarrow ... \rightarrow SG2_1 \rightarrow EE2]$, in order to meet the end-effector of Arm 2. This approach ensures that the manipulators never become static but always remain moving towards each other, leading to a faster termination at a mutual goal.

5.4.5 Integration of modifications into the existing algorithm

To incorporate the aforementioned performance enhancing modifications, some alterations must be made to the execution of the existing algorithm. Figure 18 gives an overview of this enhanced process. The goal position sampling is run at the beginning as before, however it uses a lower goal configuration sampling rate to compute a mutual goal position faster. Should this not yield a valid goal position, a higher sampling rate is used. The data retrieved from the sensors is now used to update the goal space as well as computing the repulsive fields and updating the obstacle maps. The other two changes are the combining of the manipulators' subgoal paths and the reversion to pure-APF motion planning when the mutual workspace is free of obstacles.

5.5 Monte Carlo Analysis

In this section, three approaches have been proposed to this decoupled motion planning task of attracting the manipulators' end-effectors to each other, so that they meet at a mutual goal position. These methods

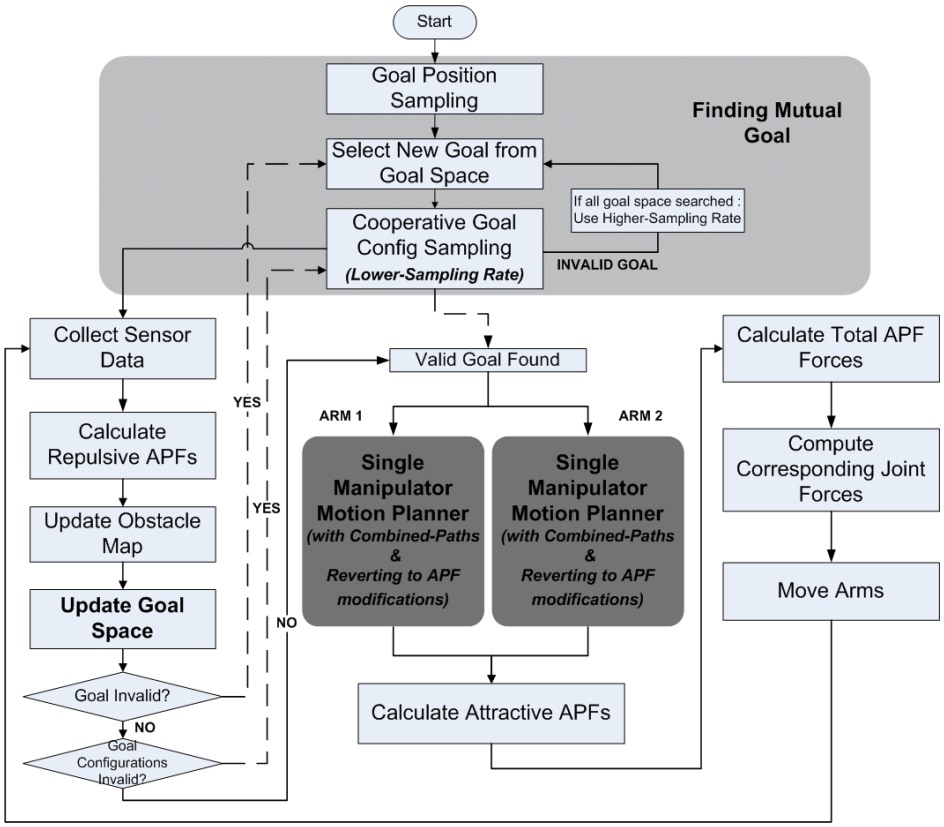


Figure 18: Online Motion Planner with performance enhancements

include;

- the standard APF approach, outlined in Section 5.2.
- the proposed online motion planner for reaching a mutual goal, outlined in Section 5.3.
- the proposed online motion planner with performance enhancing modifications, outlined in Section 5.4.

Monte Carlo simulations were used to compare the performance of these different approaches. Each algorithm was tested in 2000 varied scenarios with the same conditions as mentioned in Section 4.4. The process of goal position determination uses a step size of 5 mm for goal space sampling and 0.2 rad and 0.5 rad for goal configuration sampling for the higher and lower sampling rates, respectively.

The algorithms were tested for completeness, overall runtime and the time required to find a new goal position. To see how each performance enhancing modification affected the ability of the algorithm, numerous versions of the online motion planning algorithm were tested. Starting with the assessment of the online motion planner developed in Section 5.3, each performance enhancing modification was individually and run

Algorithm	Success Rate	Collisions	Local Minima
APF	87.8%	4.2%	8%
Online Motion Planner			
No Performance Enchancements	89.2%	4.6%	6.2%
+ Lower Sampling Rate	90.9%	4.2%	4.9%
+ Updating Goal Space	91.5%	4.2%	4.3%
+ Safely Revert to APF	94.7%	4%	1.3%
+ Combine Paths	95.2%	4%	0.8%

Table 4: Outcomes of the Monte Carlo Simulations

on the Monte Carlo simulation scenarios to judge its influence on the performance of the algorithm. This continued until all the enhancements had been added and the fully enhanced algorithm was obtained.

Table 4 shows the results from the tested scenarios. The APF approach has a high success rate in spite of its well-documented issues with local minima. There are two main reasons for this. Firstly, if one manipulator becomes stuck in a local minimum, it is not detrimental to the success of this dual-arm motion planning problem. The other manipulator can still move towards it and find a solution. Another reason is the lack of definite goal position for this motion planning problem. The goal position of the attractive APF changes with the motion of the manipulators, as described in Section 5.2. If one manipulator falls into a local minimum, it is possible that the motion of the free manipulator will change the desired mutual goal position and consequently, alter the attractive force on the trapped manipulator. This can cause the manipulator to be freed from the local minimum, allowing it to resume its motion towards the other manipulator. The flexibility of the mutual goal position means that, unless both manipulators are in a local minimum simultaneously, there is a good chance the APF approach can achieve a solution to this specific dual-arm motion planning problem.

It is demonstrated however that, the online motion planner is still more robust than the standard APF approach, with the final fully-enhanced online motion planner obtaining a success rate of 95.2% compared to 87.8% for the standard APF approach. This is due to the ability of the online motion planner to resolve local minima issues such as the collinear and reacharound local minima issues depicted in Figure 11a and Figure 11b. It is noted that as each modification is added, there is a slight increase in the success rate of the algorithm, especially when the subgoal paths are combined and there is a reverting to APF in safe conditions. This is mainly down to these modifications creating shorter, more natural paths avoiding difficult configurations for the manipulator.

Along with the robustness of the system, speed was also considered to be an important criteria for this proposed online motion planning method. The average processing time of each algorithm was recorded,

Algorithm	Success Rate	Average runtime	Average Processing Time
APF	87.8%	4.6500s	0.0917s
Online Motion Planner			
No Performance Enhancements	89.2%	11.2060s	0.1343s
+ Lower Sampling Rate	90.9%	9.7494s	0.1292s
+ Updating Goal Space	91.5%	9.6147s	0.1283s
+ Safely Revert to APF	94.7%	9.3994s	0.1246s
+ Combine Paths	95.2%	9.1451s	0.1210s

Table 5: Average processing time of each algorithm

Algorithm	Min Time	Average Time	Max Time
No Enhancements	0.446s	1.255s	15.245s
+ Lower Sampling Rate	0.110s	0.297s	3.820s
+ Updating Goal Space	0.108s	0.328s	1.692s

Table 6: Time to find a new valid goal position

with the results given in Table 5. The standard APF approach has a very fast processing time as it uses only real-time information in its calculations and thus requires much less computation. The online motion planner, proposed in Section 5.3, uses a higher-level of intelligence when computing the manipulator's paths. This makes it more robust than the standard APF approach but the processing time suffers due to the great computation needed. The computationally burdensome cooperative goal finding process is the main culprit for the slower processing time. Table 6 shows the average time this goal finding process took to find a valid goal position. The original online motion planner without enhancements took an average of 1.255 seconds every time it was required to compute a cooperative goal position and in the worst case scenario the manipulators remained static for 15.245 seconds before a goal position was found. This is clearly unacceptable for online implementation.

The performance enhancements were proposed to reduce the computational expense of this approach. Choosing a lower sampling rate was shown to find goal positions faster and thus greatly reduced the processing time of the algorithm. Also in Table 6, the effect of continually updating the goal space can be seen. Although it produced a slightly slower average time to find a mutual goal, the worst case scenario was greatly improved. This means that the motion of the manipulators is never greatly interrupted when the algorithm needs to recalibrate and find a new valid goal position. This claim is supported by observing the processing time of the algorithms in Table 5. Over the tested scenarios, using a lower sampling rate and continually updating the goal space produced a quicker average runtime than using a lower sampling rate only. The final two modifications; reverting to APF where applicable and combining the subgoal paths, helped produce shorter paths and thus faster calculations which further lowered the average runtime of the simulations, as can be

seen in Table 5.

Overall the proposed online motion planner is shown to successfully solve this type of dual-arm motion planning problem for a vast percentage of scenarios. Although it is not as fast as the standard APF, it is more robust, while the performance enhancements made to the algorithm ensure fast computation times, that are adequate for online implementation.

6 Concluding Remarks

The work presented in this paper improves on the existing manipulator motion planning techniques, for the deployment of autonomous manipulators in unknown environments. Specifically, two challenging motion planning tasks for the dual-arm system have been addressed; the execution of two individual and independent tasks; and the motion planning problem of reaching a cooperative goal both in a shared and unknown workspace.

A local motion planning solution based on standard APFs has been identified for each task. Although successful for a portion of the tested simulations, it is clear that APF motion planning alone is not enough to provide a robust solution to these dual-arm local motion planning tasks. Instead a dual-arm online motion planner consisting of goal configuration sampling, subgoal selection and APF motion planning has been proposed. This method improves on the robustness of the APF-approach for both types of decoupled motion planning tasks. The incorporation of an intelligent task planner makes it more efficient than the standard APF approach at completing individual dual-arm tasks. The proposed motion planner also maintains fast computations for both tasks, which is necessary for any online motion planning problem.

References

J. Agirrebeitia, R. Avilés, I.F. de Bustos, and G. Ajuria. A new APF strategy for path planning in environments with obstacles. *Mechanism and Machine Theory*, 40(6):645–658, 2005.

T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. ARMAR-III: An integrated humanoid platform for sensory-motor control. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 169–175, Genoa, Italy, Dec 2006.

J. Bohren, R.B. Rusu, E.G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, Wim

Meeussen, and S. Holzer. Towards autonomous robotic butlers: Lessons learned with the PR2. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5568–5575, 2011.

H. Bolandi and A. Farhad Ehaei. Trajectory planning of two cooperative mobile manipulators under closed-chain and differential constraints. *International Journal of Innovative Computing, Information and Control*, 8:1077–1102, Feb 2012.

S.J. Buckley. Fast motion planning for multiple moving robots. In *Proceedings. IEEE International Conference on Robotics and Automation*, pages 322–326 vol.1, Scottsdale, AZ, May 1989.

S Byrne, W Naeem, and R.S Ferguson. Efficient local sampling for motion planning of robotic manipulator. In *Joint Proceedings of the 13th Annual TAROS Conference and the 15th Annual FIRA RoboWorld Congress*, pages 164–175, Bristol, UK, August 2012.

S Byrne, W Naeem, and R.S Ferguson. An intelligent configuration-sampling based local motion planner for robotic manipulators. In *RoMoCo 2013, 9th International Workshop on Robot Motion and Control*, Wasowo, Poland, July 2013.

Jen-Hui Chuang, Chien-Chou Lin, and Tsun-Hou Chou. An alternate priority planning algorithm for dual-arm systems. In *IEEE International Symposium on Industrial Electronics*, volume 4, pages 3084–3089, Montreal, Quebec, Jul 2006.

Petar Curkovic and Bojan Jerbic. Dual-arm robot motion planning based on cooperative coevolution. In *Emerging Trends in Technological Innovation*, volume 314 of *IFIP Advances in Information and Communication Technology*, pages 169–178. Springer Berlin Heidelberg, 2010.

Yanqiong Fei, Ding Fuqiang, and Zhao Xifang. Collision-free motion planning of dual-arm reconfigurable robots. *Robotics and Computer-Integrated Manufacturing*, 20(4):351 – 357, 2004.

S.S. Ge and Y.J. Cui. New potential functions for mobile robot path planning. *IEEE Transactions on Robotics and Automation*, 16(5):615–620, Oct 2000.

M. Gharbi, J. Cortes, and T. Simeon. A sampling-based path planner for dual-arm manipulation. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 383–388, Xian, China, July 2008.

Q. Jia and X. Wang. An improved potential field method for path planning. In *Control and Decision Conference (CCDC)*, pages 2265–2270, Xuzhou, China, May 2010.

O Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5:90–98, April 1986.

J.-O. Kim and P.K. Khosla. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3):338–349, June 1992.

Y. Koga and J.-C. Latombe. Experiments in dual-arm manipulation planning. In *Proceedings. IEEE International Conference on Robotics and Automation*, pages 2238–2245 vol.3, Nice, France, May 1992.

Y. Koren and J. Borenstein. Potential field methods and their inherent limitations for mobile robot navigation. In *Proceedings. IEEE International Conference on Robotics and Automation*, pages 1398–1404, Sacramento, CA, April 1991.

S.M. LaValle and S.A. Hutchinson. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14(6):912–925, 1998.

J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2308–2315, Anchorage, AK, May 2010.

C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hirschmuller, S. Kielhofer, R. Konietschke, M. Suppa, T. Wimbock, F. Zacharias, and G. Hirzinger. A humanoid two-arm system for dexterous manipulation. In *6th IEEE-RAS International Conference on Humanoid Robots*, pages 276–283, Genoa, Italy, Dec 2006.

Chan-Hun Park, Kyoung-Taik Park, and Dae-Gab Gweon. Development of industrial dual arm robot for precision assembly of mechanical parts for automobiles. In *International Joint Conference SICE-ICASE*, pages 3059–3062, Busan, Korea, Oct 2006.

MinGyu Park and MinCheol Lee. A new technique to escape local minimum in artificial potential field based path planning. *KSME International Journal*, 17(12):1876–1885, 2003.

R.V. Patel and F. Shadpey. *Control of Redundant Robot Manipulators: Theory and Experiments*. Springer, 2005.

Yue Qiu, BiBo Guo, Lijun Xue, Bin Liang, and Cheng Li. On-orbit servicing to geo satellite using dual arm free-flying space robot. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2451–2456, Guilin, China, Dec 2009.

Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2010.

Yi-Chih Tsai and Han-Pang Huang. Motion planning of a dual-arm mobile robot in the configuration-time space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, pages 2458–2463, St Louis, MO, Oct 2009.

Nikolaus Vahrenkamp, Dmitry Berenson, Tamim Asfour, James Kuffner, and Rüdiger Dillmann. Humanoid motion planning for dual-arm manipulation and re-grasping tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, pages 2464–2470, St Louis, MO, Oct 2009.

Wenfu Xu, Yu Liu, and Yangsheng Xu. The coordinated motion planning of a dual-arm space robot for target capturing. *Robotica*, 30:755–771, Aug 2012.

Jae-Sang You, Dong-Hyung Kim, Sung-Jin Lim, Sung-Pil Kang, Ji Yeong Lee, and Chang-Soo Han. Development of manipulation planning algorithm for a dual-arm robot assembly task. In *Automation Science and Engineering (CASE), 2012 IEEE International Conference on*, pages 1061–1066, Seoul, Korea, Aug 2012.

Jiaxin Zhai, Weixin Yan, Zhuang Fu, and Yanzheng Zhao. Kinematic analysis of a dual-arm humanoid cooking robot. In *International Conference on Mechatronics and Automation (ICMA)*, pages 249–254, Chengdu, China, May 2012.

Qidan Zhu, Yongjie Yan, and Zhuoyi Xing. Robot path planning based on artificial potential field approach with simulated annealing. In *Sixth International Conference on Intelligent Systems Design and Applications*, volume 2, pages 622–627, Jinan, China, Oct 2006.

Yi Zhu, Tao Zhang, and Jingyan Song. An improved wall following method for escaping from local minimum in artificial potential field based path planning. In *Decision and Control, 2009 held jointly with the 28th Chinese Control Conference. Proceedings of the 48th IEEE Conference on*, pages 6017 –6022, Shanghai, China, Dec 2009.