Human-Aware Motion Planning for Safe Human-Robot Collaboration

by

Vidyasagar Rajendran

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2020

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

With the rapid adoption of robotic systems in our daily lives, robots must operate in the presence of humans in ways that improve safety and productivity. Currently, in industrial settings, human safety is ensured through physically separating the robotic system from the human. However, this greatly decreases the set of shared human-robot tasks that can be accomplished and also reduces human-robot team fluency. In recent years, robots with improved sensing capabilities have been introduced and the feasibility of humans and robots co-existing in shared spaces has become a topic of interest.

This thesis proposes a human-aware motion planning approach building on RRT-Connect, dubbed Human-Aware RRT-Connect, that plans in the presence of humans. The planner considers a composite cost function that includes human separation distance and visibility costs to ensure the robot maintains a safety distance during motion while being as visible as possible to the human. A danger criterion cost considering two mutually dependent factors, human-robot center of mass distance and robot inertia, is also introduced into the cost formulation to ensure human safety during planning. A simulation study is conducted to demonstrate the planner performance. For the simulation study, the proposed Human-Aware RRT-Connect planner is evaluated against RRT-Connect through a set of problem scenarios that vary in environment and task complexity. Several human-robot configurations are tested in a shared workspace involving a simulated Franka Emika Panda arm and human model.

Through the problem scenarios, it is shown that the Human-Aware RRT-Connect planner, paired with the developed HRI costs, performs better than the baseline RRT-Connect planner with respect to a set of quantitative metrics. The paths generated by the Human-Aware RRT-Connect planner maintain larger separation distances from the human, are more visible and also safer due to the minimization of the danger criterion. It is also shown that the proposed HRI cost formulation outperforms formulations from previous work when tested with the Human-Aware RRT-Connect planner.

Acknowledgements

Firstly, I would like to thank Dana Kulić for her support, mentorship and guidance during my master's. She has been an inspirational supervisor and her insightful suggestions, ideas and feedback have helped me grow as a researcher over the past two years.

Thank you to William Melek for being my co-supervisor and assisting me with opportunities to showcase some of my research at the RoboHub. Your input on my thesis was also greatly appreciated.

Thanks to my readers, Stephen Smith and Katja Mombaur, for reviewing and providing valuable feedback on this thesis.

I would like to extend a special thanks to Alexander Werner for spending countless hours debugging Linux, Docker and ROS issues with me during my master's. I've learned a lot about the practical side of working with robotic systems from you.

Thank you to Wesley for all the help during the master's, whether it was debugging ROS nodes or doing class projects together, it was great working with you.

Thanks to Sanjeev Bedi and Chris Rennick for allowing me to join the IDEAs clinic team and work on a very interesting educational autonomous car platform. I hope our collaborations will continue in the future.

Many thanks to Robert Wagner, Brandon J. DeHart and Marie Charbonneau for their assistance at the RoboHub and for all the lunches we had together.

Thank you to Pamela Carreño-Medrano for her advice, guidance, support and writing edits. It was a pleasure working with you and I'm excited to continue our collaborations.

Many thanks to all the professors and students that I've interacted with during my master's. I've learned so much from all of you and I'm grateful to have crossed paths. Special thanks to the members of the Adaptive Systems Lab for their help and assistance during my degree.

Thank you to Stack Overflow and ROS Answers for helping me through many late night debugging sessions.

Finally, I would like to thank my family and friends for encouraging me along during this journey.

Dedication

I would like to dedicate this thesis to my parents, Raj and Shanthi, for encouraging me to pursue graduate studies and for supporting me to the fullest along the way.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the fast-growing demand for robotic systems capable of interacting with humans in a safe, seamless, and intuitive manner, the development and integration of these systems in interactive and collaborative tasks has seen an extensive research effort in recent years, e.g., [5], [38]. Although interest in human-robot interaction has grown steadily, robots can still cause discomfort for humans through their appearance, embodiment, posture, and other attributes [43].

While factory automation has grown over the years, many manufacturing processes still require human workers. For assembly tasks that require the dexterity of humans, robots generally place parts onto specialized jigs or conveyors which the human then works on. In these scenarios, the robots are physically separated from the human to prevent collisions or incidents that may cause injury [57]. While this satisfies the objective of human safety, it generally reduces productivity and fluidity between the human and robot. To truly harness the productivity increase of robots and humans working together, minimal limitations should be imposed on the task environment and robot motion. Figure 1.1 shows an example scenario where a human and robot are interacting in a shared workspace to assemble a wooden toolbox. The robot is able to sense and pick up parts to handover to the human while also assisting when the human needs help to hold the wooden parts in place. The human is able to focus on tasks that require more dexterity while the robot focuses on the more tedious (but useful) tasks of picking and holding parts in place.

In contexts that require close-proximity human-robot interaction, such as the one shown in Figure 1.1, two objectives are usually required to be met: the safety and comfort of the human and the efficient completion of the task. In the work presented by Lasota et al. [44], results from a close-proximity human-robot collaboration user study showed that simply preventing collisions as they are about to occur can lead to inefficient human-robot interaction while also negatively impacting perceived safety and comfort for the human. The experiment involved human participants working on a collaborative task with a robot arm operating in two modes: a standard mode where the robot determined the shortest path to its goal and employed a pre-collision safety system that reduced velocity or stopped its motion based on separation distance, and an adaptive mode where the robot used human-aware motion planning to avoid portions of the shared workspace where it expected the human to be. The authors found that the human-aware motion planner lead to better perceived safety and comfort as assessed through questionnaires, while also improving team fluency assessed through various performance metrics. These results motivate the need for human-aware motion planning algorithms that can augment low-level control strategies that are focused on collision prevention and impact reduction.

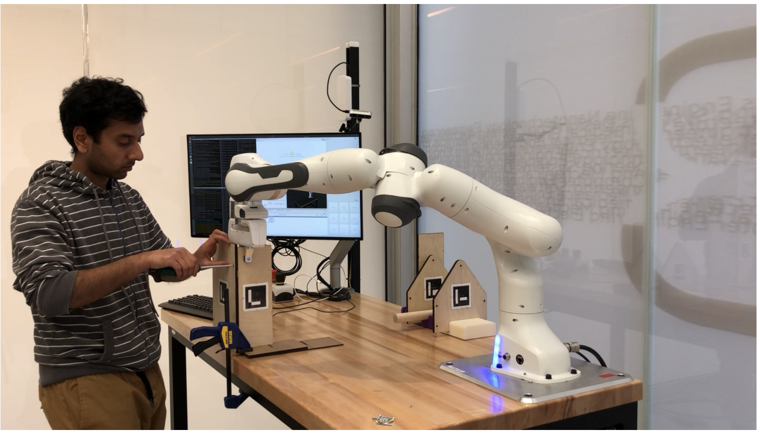


Figure 1.1: Human and robot interacting in a shared collaborative assembly workspace.

1.1 Problem Definition

Following [48], [36], [10], and [15], we formally define the motion planning problems to be addressed in this thesis.

The motion planning problem relies on abstracting the workspace of a robotic system into a *configuration space*. The *configuration space* or *C-space* (C), is the set of all possible configurations of the robot. A *configuration* $q \in C$ of a robot is a minimal set of variables that specifies the position and orientation of each rigid body composing the robot.

The subset of C inducing collisions with obstacles in the workspace is denoted C_{obs}. Assuming that $C \setminus C_{obs}$ is an open set, we denote the *obstacle-free space* as $C_{free} = cl(C \setminus C_{obs})$, where $cl(\cdot)$ denotes the closure of a set.

For a given start configuration, $q_{init} \in C_{free}$, and goal configuration, $q_{goal} \in C_{free}$, the path planning problem can be defined as a triplet $(C_{free}, q_{init}, q_{goal})$. A path over the *configuration space* is a continuous function $\sigma : [0, 1] \to C$ and it is said to be collision-free if for all $\tau \in [0, 1]$, $\sigma(\tau) \in C_{free}$. Let Σ denote the set of all paths, and Σ_{free} denote the set of all collision-free paths. The *feasible path planning problem* is defined as:

Problem 1. ***Feasible path planning problem.*** *Given a path planning problem* $(C_{free}, q_{init}, q_{goal})$*, find a feasible path* $\sigma(\tau) \in C_{free}$ *such that* $\sigma(0) = q_{init}$ *and* $\sigma(1) = q_{goal}$*, if one exists, or report failure otherwise.*

For planning on a a continuous-cost space, we define a cost-function $c : C \to R_{\geq 0}$ such that a positive real value is assigned to all configurations in C. Then the *cost-space path planning problem* is:

Problem 2. ***Cost-space path planning problem.*** *Given a path planning problem defined as the quadruplet* $(C_{free}, q_{init}, q_{goal}, c)$*, solve the feasible path planning problem while accounting for the cost function c during exploration of the C-space.*

To be specific, methods aimed at solving the *cost-space path planning problem* perform a rejection sampling of configurations in the *C-space* by imposing specific cost constraints evaluating each configuration on its cost alone, or on the cost variation to move locally between configurations.

It should be noted that, unlike the *optimal path planning problem* which aims to find the *minimum* cost path in Σ_{free}, the *cost-space path planning problem* only finds a feasible low cost path.

1.2 Thesis Contributions

In this thesis, we propose an HRI cost formulation for planning with articulated robot arms in the presence of humans. The work builds on the formulations in [63] and [50] and integrates the danger criterion formulation from [41] as a separate cost function term. A modified RRT-Connect planner, dubbed Human-Aware RRT-Connect, plans using the proposed HRI cost formulation. The modified planner is compared to the baseline RRT-Connect planner [34] in a human-robot collaboration simulation study. From the simulation results, we show that the proposed HRI cost formulation and Human-Aware RRT-Connect planner outperform the baseline planner for metrics such as minimum separation distance from the human, average separation distance from the human, path visibility and average robot inertia among others. The proposed HRI cost formulation outperforms formulations from previous work when tested with the Human-Aware RRT-Connect planner.

1.3 Thesis Organization

In Chapter 2, related work on motion planning in high dimensional spaces and motion planning in the presence of humans is reviewed. In Chapter 3, the description of the proposed human-aware motion planning method is detailed. Results from simulation experiments of the human-aware motion planning method in a shared human-robot workspace are described in Chapter 4. Conclusions and future work are discussed in Chapter 5.

Chapter 2

Literature Review

First, we review general techniques for planning in high dimensional spaces. Then, we move on to previous work in motion planning for human environments.

2.1 Sampling Based Planning Methods

Sampling based planning methods avoid constructing an explicit representation of the configuration space and instead, rely on probing the *C-space* with a sampling scheme [47]. This class of planning methods have become popular due to their speed and simplicity, especially in high-dimensional configuration spaces. Where classical grid-search planners suffer from the curse of dimensionality, i.e. they become computationally intractable with the increase of the configuration space dimension [53], sampling-based methods have been shown to solve planning problems with over 1000 dimensions [60]. Sampling-based planning methods are *probabilistically complete* meaning if a path exists, it is guaranteed to be found as the number of samples tends to infinity.

In this section we cover two of the most popular sampling-based planning algorithms, Probabilistic Road Maps and Rapidly Exploring Random Trees.

2.1.1 Probabilistic Road Maps

Probabilistic Roadmaps (PRMs) [37] were introduced to overcome the *curse of dimensionality* that exists in classic grid search techniques [49]. The PRM method is split into a *learning phase* and *query phase*. The *learning phase* constructs the probabilistic roadmap by repeatedly sampling random free configurations of the robot and connecting these configurations using a fast *local planner*. The roadmap is stored as an undirected graph where nodes comprise of valid configurations in C_{free} and edges represent collision-free paths between configurations. In the *query phase*, a *query* asks for a valid path between a start and goal configuration. First, the start and goal configurations are connected to the roadmap, followed by a graph search to find a feasible path in the roadmap. PRMs are "multiple-query" solvers in that the same roadmap can be used to solve multiple motion planning problems in the same static workspace.

2.1.2 Rapidly Exploring Random Trees

The Rapidly Exploring Random Tree (RRT) planner, originally published in [46], is a popular single-query motion planner that can solve problem 1, the feasible path planning problem. Similar to PRMs, they avoid an explicit construction of C_{free} by randomly sampling the configuration space. However, the approach is simplified in that no cycles are added to the resulting graph. Starting from q_{init}, the RRT method builds a tree T on the *C-space*. At each iteration, a configuration q_{rand} is sampled from C and an extension by step size ϵ is attempted from the nearest-neighbor in the tree, q_{near}, towards q_{rand}. If the extension succeeds, i.e. there were no collisions, a node q_{new} is added to $\mathcal{T}$ and connected by an edge to q_{near}. This process is repeated until a stop condition, such as the reaching of the goal configuration, q_{goal} is reached. The procedure is sketched in Algorithm 1. Figure 2.1 visualizes the EXTEND operation showing how a new node is added into the tree, $\mathcal{T}$.

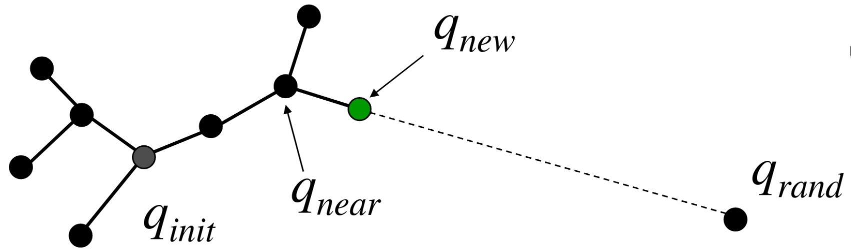


Figure 2.1: The EXTEND operation during the RRT expansion [12].

An extension to the original RRT planner was presented in [34] where two trees are grown, one from the start configuration and one from the goal configuration. Both trees follow the original RRT planner's EXTEND procedure but also include a *Connect* procedure that tries to grow each tree as far as possible toward q_{rand}, thereby connecting the

Algorithm 1 RRT

```
Input: Start and goal configurations
Output: Feasible path or Failure
Parameters: $\epsilon$ is the step size for a motion
$\mathcal{T}$.Init($q_{init}$)
for $i = 1$ to $K$ do
    $q_{rand} \leftarrow$ Rand($\mathcal{C}$)
    $q_{near} \leftarrow$ Nearest($q_{rand}, \mathcal{T}$)
    EXTEND($\mathcal{T}, q_{near}, q_{rand}$)
Return FAILED
function EXTEND($\mathcal{T}, q_{near}, q_{rand}$)
    if NEW_CONFIG($q, q_{near}, q_{new}$) then
        if DISTANCE($q, q_{new}$) $\leq \epsilon$ then
            Return Reached;
        else
            Return Advanced;
    Return Trapped;
function NEW_CONFIG($q, q_{near}, q_{new}$)
    $q_{new}$ $\leftarrow$UNIT_V($q, q_{near}$) $* \epsilon$
    if OBSTACLE_FREE($q_{new}$) then
        Return $True$;
    Return $False$
```

two trees together. This simple greedy heuristic achieves a speedup over the original RRT while maintaining the properties of probabilistic completeness and a uniform exploration of C_{free}.

To explore cost spaces, in [22], the authors propose a threshold based RRT-Connect planner for rough-terrain navigation. In the Extend procedure, the cost of q_{new} is checked and only accepted into the search tree if it is below a cost threshold. This threshold is initialized to a low value and increased based on the number of planner iterations that have passed. A drawback of the approach is the non-decreasing cost threshold which degrades the planner to RRT-Connect in cost topologies containing more than one local maxima along the trajectory. Transition-based RRT (T-RRT), another work aimed at planning on cost spaces, integrates a transition test that favours exploration of low cost configurations in C [32]. The transition test, based on stochastic optimization methods, accepts or rejects the move from q_{near} to q_{new} based on their respective costs. This guides the planner to follow valleys and saddle points of the cost-space in order to compute low-cost paths.

While the above variants improve different aspects of the original RRT, they do not provide any guarantees on finding the optimal minimum cost path. RRT* was introduced in [36] and was shown to be an *asymptotically optimal* planner, i.e. as the running time of the planner approaches infinity, the cost of the path approaches the optimal value. However, RRT* is known to converge slowly to the optimal path, especially in high-dimensional cost spaces [17].

2.2 Trajectory Optimization Methods

Another class of planners developed recently are trajectory optimization methods. These planners differ from sampling based planners in that they use continuous optimization techniques to find high quality paths. These methods generally define a path representation and a cost function and have to be initialized with an initial path (e.g., a straight line interpolation from the starting configuration to the goal configuration or the output path of a sampling based planner). At each iteration, the gradient of the current path cost is used to determine a new path of lower cost. This process is repeated until the path cost cannot be updated any further (generally a local minimum of the formulated cost function). These methods also generally assume fixed time steps between each waypoint in the path and thus are called *trajectory optimization methods* (as opposed to path optimization methods). Some examples of trajectory optimization planners are CHOMP (Covariant Hamiltonian Optimization for Motion Planning) [56], STOMP (Stochastic Trajectory Optimization for Motion Planning) [35] and TrajOpt [59].

2.3 Motion Planning in the Presence of Humans

Sampling based planning methods have, until recently, focused on producing feasible paths without considering path quality. This has lead to planners that produce paths quickly at the cost of being less than optimal. In many contexts, such as human-robot interaction, producing high quality paths with respect to some quality criterion is preferred. In [62], it is noted that applying and extending classical motion planning techniques to the problem of human-robot interaction was first done in [64], work that was published in the year 2007. Since then, the field has grown significantly as summarised in recent reviews [40], [43].

2.3.1 Mobile Robot Motion Planning in Human Environments

Although the focus of this thesis is on planning in the presence of humans for articulated robot arms, the literature on mobile robot motion planning in the presence of humans is relevant and informative. In this line of research, many works introduced cost formulations to make robot paths safer, more comfortable and more human-like. Some of these formulations have also been extended to planning for articulated robot arms.

In [7], Bennewitz et al. presented a robot motion strategy that was modelled after humans. The mobile robot tries to predict the actions of the people in an environment and avoids areas where humans are predicted to be. Data recorded with laser rangefinders are clustered using an expectation minimization algorithm and this data is used to derive a hidden Markov model that estimates the future positions of people in the environment. From this, the probabilistic belief about potential trajectories of persons is incorporated into an A* algorithm to determine the minimum-cost path (that would avoid human areas) in the three-dimensional configuration time-space of the robot. The results showed that the approach can reliably learn motion patterns and can be used to improve navigation behaviour for mobile robots.

In [64], a human aware mobile robot motion planner that integrated a static human's accessibility, their vision field and their preferences in terms of relative human-robot placement was introduced. A "safety criterion" is formulated and tries to maintain a separation distance between the robot and the human. The formulation takes into account different human-robot interaction scenarios and also reasons about a human's preferences when they are sitting or standing. A "visibility criterion" aims to keep the robot in the human's field of view. Finally, a "hidden zones" criterion is introduced to better formulate costs for robot positions that are occluded from the human's field of view by obstacles. Similar formulations are introduced and extended in [63] to produce robot motions that clearly show the robot's intention. One drawback of these works is that the different cost functions are defined on Cartesian grids and the planners employed are based on classic grid search techniques. This approach may be sufficient in the absence of strong workspace constraints but may fail in cluttered environments with many obstacles. Moreover the computation time for high dimensional configuration spaces will likely become intractable as the dimension increases.

In the domain of autonomous vehicles, Morales et al. studied safe motion planning for vehicles with human passengers inside [51]. The work developed the Human-Comfortable Path Planner (HCoPP) system for autonomous passenger vehicles with the aim to improve the feeling of comfort for human passengers. This was achieved by augmenting the shortest-path constraint with constraints related to relevant environmental features. A three layer costmap is integrated into the approach to balance the shortest path constraint along with the path comfort constraints. Through a user study, the results showed that the HCoPP system produced paths that were perceived to be more comfortable to humans.

Costmap formulations first introduced for mobile robot navigation have also been adapted to articulated robot motion planning in the presence of humans.

2.3.2 Articulated Robot Motion Planning in Human Environments

In this section we review the literature related to human-aware and safe planning for robot arms.

In the planner proposed by Kulić et al., a danger criterion based on robot inertia and relative distance between the human and robot center of mass is formulated [41]. The proposed planner uses a two stage approach with the first stage searching for robot configurations that minimize the danger criterion while the second stage seeks the goal. The results showed that the formulated danger criterion, paired with the planning approach, was able to generate safe, feasible paths for interaction. Minimizing the danger criterion during planning reduces the chance of collision by distancing the robot from the human. Moreover, for unanticipated collisions, the risk of injury is reduced because the planner tries to ensure that the robot stays in low inertia configurations when moving from the start to the goal.

Lacevic et al. propose a safe RRT-Connect planner which modifies the way the start and goal trees are expanded [42]. They first formulate a *cumulative static danger field* which reasons about the distance from a workspace point to points of interest on the robot. This formulation is used when selecting nearest neighbors in the trees for expansion. Instead of selecting neighbors based solely on a distance metric, the *cumulative static danger field* is also minimized leading to expansion from safer regions of the configuration space. The planner is implemented for a 6-DOF robotic arm in two static simulation environments with multiple cylinders as obstacles. The planner is able to produce paths that remain far from obstacles. The authors test their algorithm 1000 times for the same environment and start and goal configurations showing that the average *cumulative static danger field* is minimized effectively.

In [50], Mainprice et al. extend the work from [64] and [63] and formulate similar criteria such as human field of view, safety distance and arm comfort while employing the T-RRT planner [32] to generate safe motion plans in cluttered environments for a handover task. The work was later integrated into a control architecture that combined a supervisory attentional system with a human-aware manipulation planner [9]. Experiments were run with 5 subjects in different human-robot manipulation tasks such as give, receive, pick and place. Users generally perceived the interactions as safe, reliable and natural.

In the work by Vahrenkamp et. al. [72], an approach for determining suitable locations for human-robot interaction tasks is formulated. The method introduces a task specific Interaction Workspace as a representation of the workspace accessible by both the human and robot. Several quality measures for both the human and robot are considered during the construction of the Interaction Workspace such as joint space travel, workspace travel, visibility, safety distance and effort (energy consumption of arm postures). An exemplar hand-over task where a robot hands tools to a human and vice versa is described and implemented. The results showed that the computation time and size of the Interaction Workspace was inversely proportional to the distance between the human and robot.

Hayne et al. approach the problem of planning in the presence of humans from a different perspective. The authors propose cost function formulations that avoid the workspace previously occupied by humans, and increase consistency of the robot motion so that movements are as predictable as possible for the human [28]. These formulations are used in the TrajOpt planner for a specific human-robot shared workspace task. Results from simulations and physical experiments show that the best task success rates occur when using either cost function term individually or both cost function terms together in comparison to a baseline method. A similar approach is formulated in [77] where an occupancy cost map and signed distance field map are used to aid a STOMP planner for a shared human-robot workspace task.

While most of the works discussed here propose offline planners, Faroni et al. propose a framework based on a model predictive control approach where trajectory velocities are modified online to slow down task execution and the redundancy of the robot system is exploited to maximize the distance from the human operator [24]. The approach generates a path using a high-level planner (e.g. STOMP) and then modifies it online. The predictive horizon is 0.5s and the human position is assumed to be constant during this time. The results show that the robot is able to successfully slow down and deform its trajectory when the distance between the robotic system and the human is less than a threshold value.

In [20], Dragan et al. investigate how robot motion can convey intent and model the problem using mathematical definitions of legibility and predictability of robot motion. In their work, they found that legible robot motions are often quite different from predictable motions making predictability and legibility contradictory properties. They evaluated the validity of their mathematical models through a user study where participants predicted which goal a robot or human was reaching toward in a set of videos. Trajectories that were rated as more legible by the participants were generally considered to be more legible as evaluated by the mathematical formulation developed in the work, but trajectories with a high mathematical predictability were not always rated as more predictable by humans. The authors explained that this could be because participants had a wide variety of expectations of how the robot arm would move and thus emphasized the importance of considering human expectation when planning robot motions. In later work [21], Dragan et al. also incorporate the notion of legibility into a trajectory optimization planner and show that there must be a compromise between legibility and predictability so that the generated robot motion does not become too surprising or unpredictable for the human.

2.4 Summary

This chapter provided an overview of sampling-based motion planning methods, trajectory optimization planners and motion planning in the presence of humans. Most works implement cost functions that reason on the human presence and then employ planners to plan using those cost functions. Early work in the field of human-aware motion planning defined Cartesian grids and used classic grid search techniques to plan on them. Later work extended this approach to sampling based planners.

Most of the works presented in this chapter are formulated for specific human-robot interaction scenarios such as handovers or close-proximity motion planning and present general path quality criteria such as integral cost or mechanical work. While these quality criteria show that the algorithms work as intended for the costs that they minimize, more insight about the algorithms can be gathered from metrics such as average separation distance, trajectory visibility and others. Moreover, many of the works do not show large scale tests of their formulations in a variety of different human-robot shared workspace scenarios. This thesis aims to test the proposed algorithm in various environments and shows planner usage for both close-proximity motion planning and handovers. We report general path quality criteria and metrics such as average separation distance, robot inertia and path visibility among others.

Chapter 3

Human-Aware Motion Planning Formulation

In this chapter the proposed human-aware motion planning method is detailed. A composite cost function is proposed that considers the human and the robot in the shared workspace. First, a distance cost which aims to keep the closest point on the human away from certain locations on the robot is introduced. Next, a cost on the human's visibility of the robot is introduced which tries to ensure that the robot is within the human's field of view during motion. Finally, a danger criterion cost is introduced that reasons on the level of danger imposed by the robot on the human through relative center of mass distances and robot inertia. These separate cost terms are combined in a weighted sum and can be evaluated for a single robot configuration. The cost function is then used within a sampling based planner that plans the robot's motion by checking whether new configuration samples are collision free and within certain cost bounds. An overview of the approach is given in Figure 3.1.

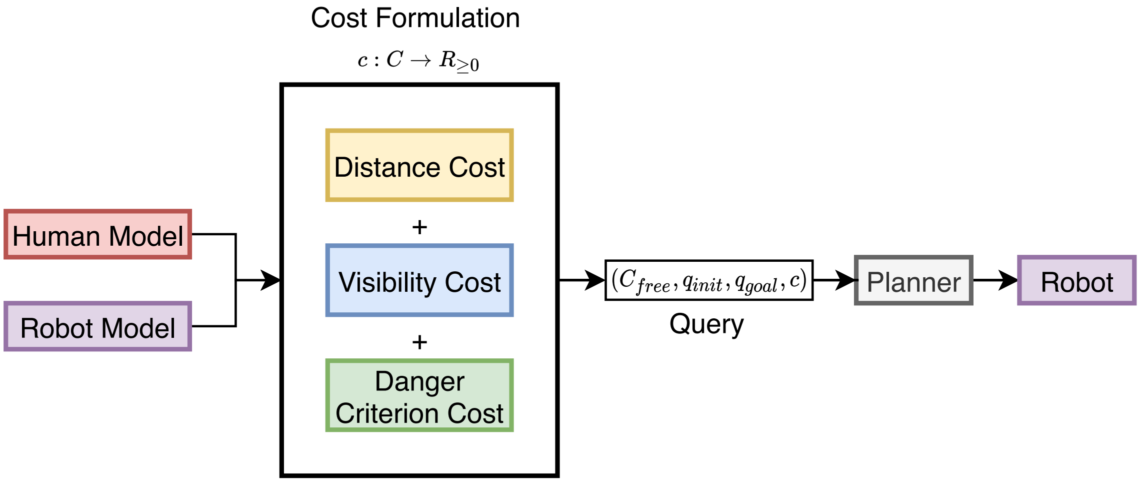


Figure 3.1: The overall human-aware motion planning approach used in this thesis.

3.1 Human-Aware Cost Formulation

In this section, the various cost function terms are described in detail.

A 32-DOF human model was adapted from [31] to represent the human in the shared human-robot workspace as shown in Figure 3.2. Using this model, various cost function terms are formulated based on the human's position, the human's head orientation and the robot's configuration.

We denote H as the human model configuration. H has 38 terms in its configuration vector, 32 for the joint degrees of freedom and 6 describing the position of the human model in the $World$ frame. We denote the vector $[i, j, k]$ as a voxel in the workspace.

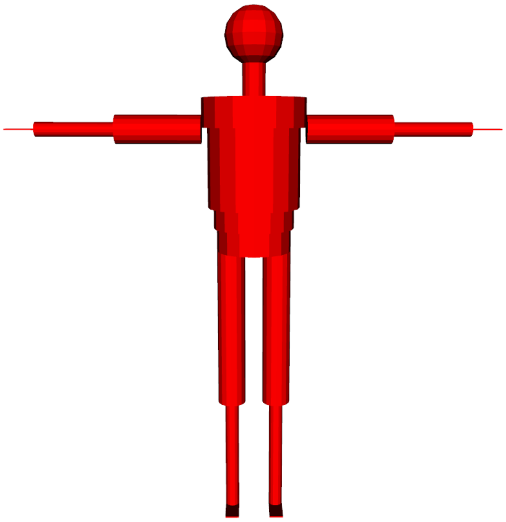


Figure 3.2: 32-DOF Human Model.

3.1.1 Distance Cost

Similar to the formulation in [50], a distance cost, $c_{dist} \in [0, \infty]$, is introduced. This cost function tries to ensure the safety of interaction by maximizing the separation distance between the robot and human, until some maximum threshold where the cost becomes null. The approach in [50] only considers an approximate bounding volume of the human body without considering arm geometry. In this work, a Euclidean distance transform on the entire human model (including all limbs) is calculated. Figure 3.3 shows an example of a human distance costmap.

The distance cost function is implemented as a repulsive potential function and is given as:

$$c_{dist}(H, x) = \begin{cases} 0 & EDT(H, x) \geq d_{max} \\ \gamma \left(\frac{1}{EDT(H,x)} - \frac{1}{d_{max}} \right)^2 & otherwise \end{cases}, \tag{3.1}$$

where $x \in [i, j, k]$, EDT is the Euclidean Distance Transform, d_{max} is the maximum extent of the Euclidean Distance Transform and γ is a scaling constant used to scale c_{dist} such that the cost is one when the distance between the human and the robot is d_{min}, the minimum allowable separation distance, and zero when the distance between the human and robot is d_{max} or larger. Upon collision between the robot and the human, the cost evaluates to ∞. γ is given as:

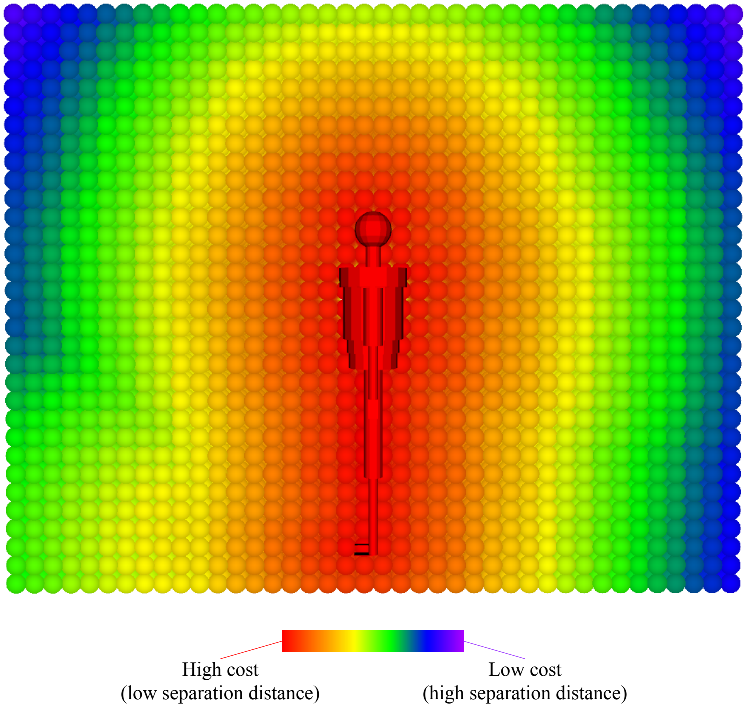


Figure 3.3: A cross section of the human distance costmap defined on the workspace with orange indicating high cost and violet indicating low cost.

$$\gamma = \left(\frac{d_{min} \cdot d_{max}}{d_{min} - d_{max}} \right)^2 . \tag{3.2}$$

$EDT(H, x)$ provides the closest distance to the human from point x. The human model is first voxelized before a *dynamic brushfire algorithm* computes the shortest Euclidean distance to each voxel. The distance values can be updated incrementally if the human moves to a different position. The implementation is further detailed in [45], and is directly used in this work.

3.1.2 Visibility Cost

The visibility cost, $c_{vis} \in [0, 1]$ is based on the angle $\psi \in [0, \pi]$, between the human's current head orientation, and the line between point $p \in [i, j, k]$ in the workspace and the head.

To calculate ψ, we assign a head frame denoted $Head$ to the human model as shown in Figure 3.4. Let T_{WH} be the 4x4 homogenous transformation matrix from the $World$ frame to the $Head$ frame and let h denote the translational component of the transformation. R is the 3x3 rotation matrix of the transformation. T_{WH} is given as:

$$T_{WH} = \begin{bmatrix} R & h \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & h_1 \\ r_{21} & r_{22} & r_{23} & h_2 \\ r_{31} & r_{32} & r_{33} & h_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.3}$$

We model the human's direction of gaze as the $x - axis$ of the head frame as shown in Figure 3.4. From this, we take $\vec{g} = [r_{11}, r_{21}, r_{31}]$, the first column of the 3x3 rotation matrix in T_{WH}. If we let $\vec{x}$ be the vector from the point p to the head frame, i.e. $\vec{x} = [i - h_1, j - h_2, k - h_3]$, we can calculate ψ as the arccosine of the dot product between the normalized vectors $\vec{g}$ and $\vec{x}$:

$$\psi = \arccos\left(\frac{\vec{g}}{\|\vec{g}\|} \cdot \frac{\vec{x}}{\|\vec{x}\|}\right). \tag{3.4}$$

From the formulation in [76], human attention is more attracted to social and visual cues in a small central area known as the effective field of view (eFOV) shown in Figure 3.6. Features that are collected within this region induce higher levels of attention from the human. Studies conducted in [69] and [73], showed that humans have a tendency to look towards the center of an image rather than the periphery, regardless of the contents of the image. Hence, the final visibility cost function is normalized to a value between 0 and 1 and then squared to give smaller angles near the center of the human's field of view, lower costs. c_{vis} is given as:

$$c_{vis}(H, x) = \left(\frac{\psi}{\pi}\right)^2. \tag{3.5}$$

Since the eFOV is a cone with radius 15°, raising the cost to a power of 2 has the effect of making c_{vis} roughly 0 for angles ψ within the eFOV. This is plotted in Figure 3.7.

The costmap is visualized in Figure 3.5 where the direction of gaze is denoted by a red arrow pointing out from the human head. Violet and blue spheres indicate points of high visibility and orange spheres denote points of low visibility.

When used during planning, the visibility cost has the effect of producing paths that are as visible as possible for the human while moving to the goal configuration.

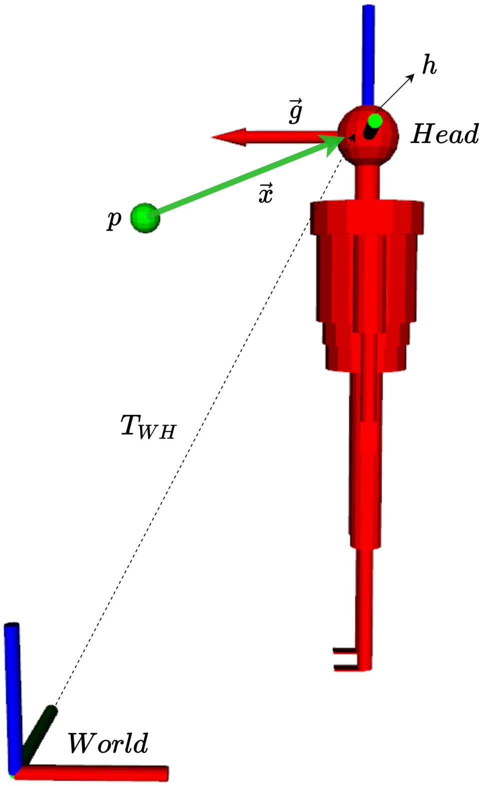


Figure 3.4: The two vectors $\vec{g}$ and $\vec{x}$ are used to calculate the angle ψ from the point p. T_{WH} denotes the homogeneous transformation matrix from the $World$ frame to the $Head$ frame and h denotes the location of the head in the $World$ frame.

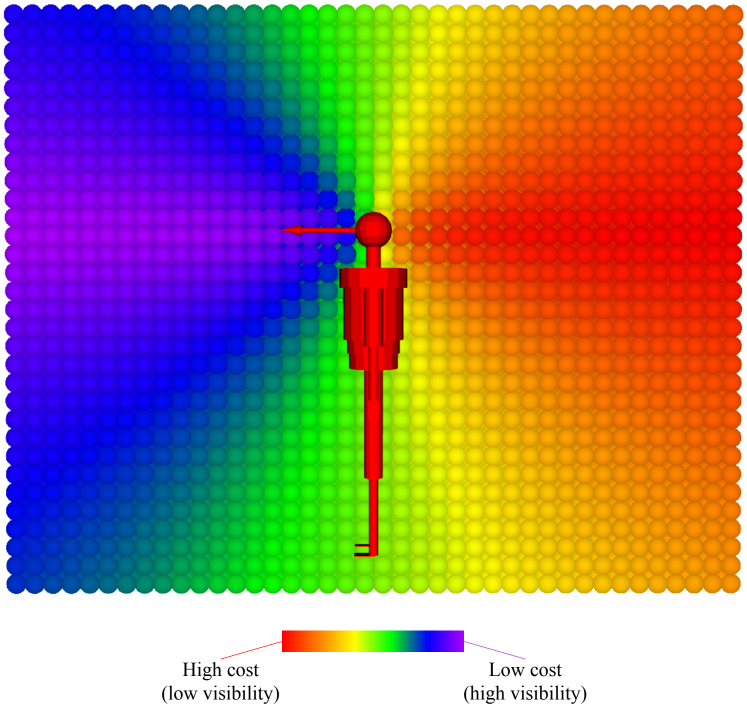


Figure 3.5: A cross section of the human visibility costmap defined with orange indicating high cost (low visibility) and violet indicating low cost (high visibility).

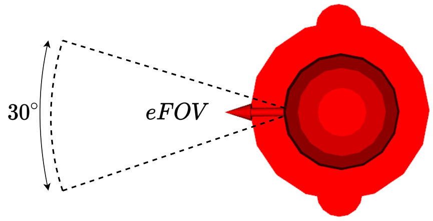


Figure 3.6: The human's effective field of view (eFOV).

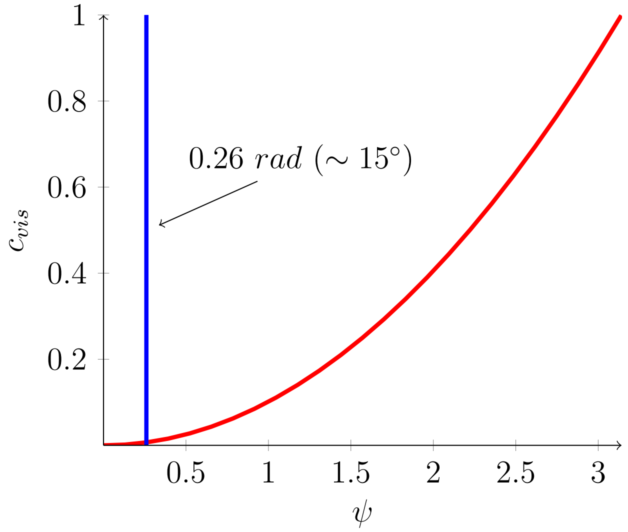
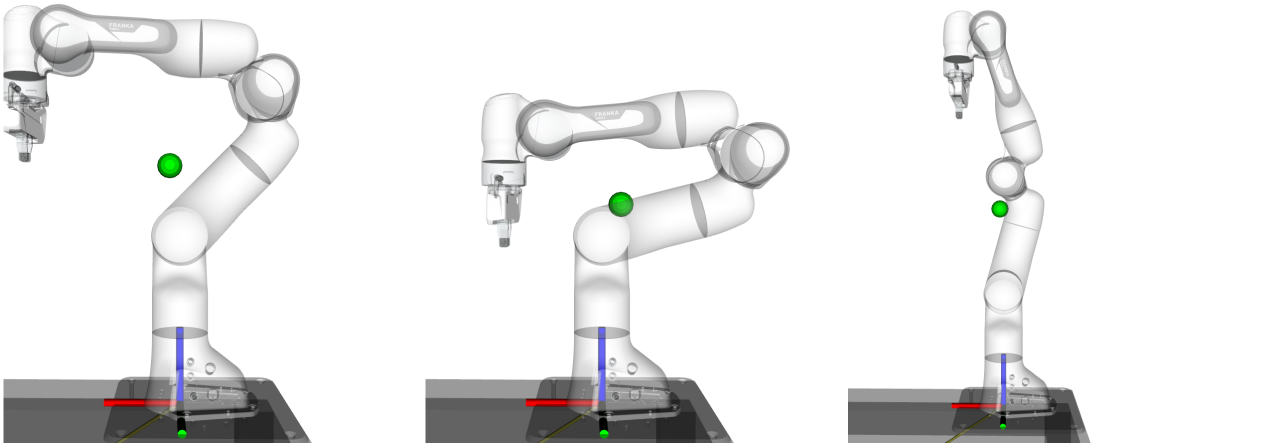


Figure 3.7: The cost c_{vis} is plotted against the angle ψ. The cost for angles within the eFOV are close to zero as shown by the vertical blue line.

3.1.3 Danger Criterion Cost

To improve human safety, $c_{dc} \in [0, 1]$ is introduced to bias the search towards robot configurations that minimize a danger criterion. To accomplish this, a formulation similar to the product based danger criterion from [41] was used. The criterion is based on the robot inertia and relative distance between the human and the robot center of mass.

The inertia of a general articulated body is a 3x3 tensor so a scalar value representing the robot inertia for a configuration q is required. Following [41], since the robot's inertia may be distributed in more than one plane, we take the highest eigenvalue of the inertia tensor, the largest *principal moment of inertia*, as the scalar measure. We denote this value as I_s. Figure 3.8 shows three configurations of a robot arm with different scalar inertia values.



(a) Home configuration (b) Low inertia configuration (c) High inertia configuration

Figure 3.8: Examples of different arm configurations showing the center of mass with a green sphere relative to the arm's base frame. The scalar inertia values for each configuration are: a) $I_s = 2.088$, b) $I_s = 1.652$, c) $I_s = 2.872$. The size of the green sphere bears no significance.

The inertia criterion is given as:

$$c_{inertia}(I_s) = \left(\frac{I_s}{I_{max}}\right)^4, \tag{3.6}$$

where I_s is the scalar moment of inertia value about the center of mass and I_{max} is the maximum safe inertia of the robot. As noted in [41], I_{max} can be set based on the largest force magnitude that does not cause pain and the maximum acceleration of the robot. The cost is raised to the fourth power in order to give higher scalar inertia values higher costs relative to lower scalar inertia values.

For the center of mass distance criterion between the human and robot the following formulation is used:

$$c_{com_{dist}}(H, d_{cm}) = \begin{cases} 0 & d_{cm} \geq d_{max_{com}} \\ k\left(\frac{1}{d_{cm}} - \frac{1}{d_{max_{com}}}\right)^2 & otherwise \end{cases}, \tag{3.7}$$

where d_{cm} is the center of mass distance between the human and the robot and $d_{max_{com}}$ is the maximum center of mass distance. k is a scaling constant used to scale the center of mass potential function such that the cost is zero when the center of mass distance between the human and robot is large (greater than $d_{max_{com}}$) and is one when the center of mass distance between the human and robot is $d_{min_{com}}$. k is calculated as:

$$k = \left(\frac{d_{min_{com}} \cdot d_{max_{com}}}{d_{min_{com}} - d_{max_{com}}}\right)^2. \tag{3.8}$$

The final product-based danger criterion is computed using equations 3.6 and 3.7 as:

$$c_{dc}(H, q) = c_{inertia}(I_s) \cdot c_{com_{dist}}(d_{cm}). \tag{3.9}$$

3.2 Final Cost Function Formulation

The full cost formulation, combining equations 3.1, 3.5 and 3.9 is given as:

$$c(H, q) = \max_j w_{dist} c_{dist}(H, FK(q, p_j)) + \max_j w_{vis} c_{vis}(H, FK(q, p_j)) + w_{dc} c_{dc}(H, q) \tag{3.10}$$

with FK being the forward kinematics function of the robot, and p_j indicating the j^{th} point of interest. The points of interest can be any of the 7 joints on the robot. For example, when planning, if it is important for the end effector, wrist and elbow of the robot to avoid the human, these can be specified as points of interest. The maximum cost out of the j points of interest is used when more than one point of interest is specified.

w_{dist}, w_{vis} and w_{dc} are the three weighting parameters for the cost function terms. Depending on the robot behaviour desired, these can be set accordingly.

As a note, c_{dist} and $c_{com_{dist}}$ are both distance based costs with the difference being that the former is a cost for the closest distance between specific points on the robot and the entire human while the latter is a cost for the center of mass distances. Depending on the planner employed and the human-robot interaction context, the weights for these respective costs can be set to attain different planned paths. For example, if maintaining a strict clearance between the robot and the human is of priority, w_{dist} should be set higher than w_{dc}. If minimizing inertia and center of mass distance is more important than maintaining a minimum separation distance from the human (e.g., for a handover), then w_{dc} should be set larger than w_{dist}. A weight sensitivity analysis is presented in Section 4.5 where the choice of weights is investigated in more detail.

3.3 Human-Aware RRT-Connect Planner

As the cost function formulation presented in Section 3.2 returns a cost for a single robot configuration, a sampling-based planner is ideal to plan a geometric path. This work uses a modified version of RRT-Connect [34].

Algorithm 2 details the Human-Aware RRT-Connect planner. The function HA-RRT _CONNECT, starting at line 1, is the main planner that is modelled after the original RRT-Connect algorithm. A cost thresholding approach similar to [22] is introduced with the main difference being that the threshold is adaptively tuned (similar to [32]) instead of being a non-decreasing parameter.

The planner maintains two trees, $\mathcal{T}_{\mathcal{A}}$ and $\mathcal{T}_{\mathcal{B}}$ and grows them over each iteration until they become connected and a path from the start to the goal configuration is found. In each planning iteration, one of the two trees is extended and then an attempt is made to connect the nearest vertex of the other tree to the newly generated vertex. If no connection is made, the trees are swapped to reverse their roles and allow both trees to explore C_{free}.

In the original RRT-Connect algorithm, a distance metric is used to find the nearest configuration, q_{near}, to extend the search tree. In Human-Aware RRT-Connect, following a similar approach to [42], q_{near} minimizes $\rho(q, q_{old}) + \alpha COST(q_{old})$ that captures both the distance and cost assessment of the potential nearest neighbour node, q_{old}. In this work, ρ is the L^2-norm metric function (*Euclidean distance*).

Compared to RRT-Connect, a cost threshold, c_{thres}, is introduced, which imposes an additional criterion for the acceptance of a configuration into any of the trees along with collision checking. Line 3 of Algorithm 2 shows the initialization of the parameter c_{thres}, passed through EXTEND to NEW_CONFIG. NEW_CONFIG extends the search tree by a single step ϵ and produces the configuration q_{new}. q_{new} must be obstacle free and below c_{thres} as described in line 22 of Algorithm 2. Additionally, if this criterion is satisfied, we check if the cost of q_{new} is less than it's nearest neighbor, q_{near} and accept it if so. Otherwise, if q_{new} is not of lesser cost than q_{near}, it is accepted into the tree with a probability of $\eta \in [0, 1]$. This is done to ensure that the trees continue exploring C_{free} instead of getting trapped at local cost minimas.

Line 15 shows the algorithm termination criterion when a path is found. If the distance between the new configuration that was added to the current search tree is less than ϵ away from the other tree then the two trees are connected and the generated path is returned. If after K iterations, no path is found, the planner terminates and returns failure.

Algorithm 2 Human-Aware RRT-Connect Planner

```
Input: Planning problem $(C_{free}, q_{init}, q_{goal}, c)$
Output: Feasible low cost path or Failure
Parameters: $\epsilon$ is the step size for a motion
function HA-RRT_CONNECT($q_{init}, q_{goal}$)
    $\mathcal{T}_{\mathcal{A}}$.Init($q_{init}$), $\mathcal{T}_{\mathcal{B}}$.Init($q_{goal}$)
    $c_{thres} \leftarrow c_{init}$                                   ▷ cost threshold
    for $i = 1$ to planner_iterations do
        $q_{rand} \leftarrow$ Rand($\mathcal{C}$)
        if EXTEND($\mathcal{T}_{\mathcal{A}}, q_{rand}, c_{thres}$) == $Trapped$ then
            if CONNECT($\mathcal{T}_{\mathcal{B}}, q_{new}, c_{thres}$) == $Reached$ then
                Return PATH($\mathcal{T}_{\mathcal{A}}, \mathcal{T}_{\mathcal{B}}$)
        SWAP($\mathcal{T}_{\mathcal{A}}, \mathcal{T}_{\mathcal{B}}$)
        $\psi \leftarrow$ UPDATE_THRES($c_{thres}$)                   ▷ update cost threshold
    Return FAILED
function EXTEND($\mathcal{T}, q, c_{thres}$)
    $q_{near} \leftarrow \arg\min_{q_{old}\in\mathcal{T}}\{\rho(q, q_{old}) + \alpha COST(q_{old})\}$
    if NEW_CONFIG($q, q_{near}, c_{thres}, q_{new}$) then
        if $\rho(q, q_{new}) \leq \epsilon$ then
            Return Reached;
        else
            Return Advanced;
    Return Trapped;
function NEW_CONFIG($q, q_{near}, c_{thres}, q_{new}$)
    $q_{new} \leftarrow$UNIT_V($q, q_{near}$) $* \epsilon$
    if OBSTACLE_FREE($q_{new}$) & COST($q_{new}$) $< c_{thres}$ then
        if COST($q_{new}$) $<$ COST($q_{near}$) then
            Return $True$;
        else if rand01() $< \eta$ then
            Return $True$;
    Return $False$
function CONNECT($\mathcal{T}, q, c_{thres}$)
    repeat
        $S \leftarrow$ EXTEND($q, \mathcal{T}, c_{thres}$)
    until not $S$ == Advanced
    Return $S$;
```

Algorithm 3 Cost Threshold Update

```
Parameters: $c_{rate}$ is the rate at which to increase or decrease the threshold, $n_{success_{max}}$ is
    the maximum number of successful additions to the tree, and $n_{fail_{max}}$ is the maximum
    number of unsuccessful tree additions
procedure UPDATE_THRESHOLD($c_{thres}$)
    if $n_{success} > n_{success_{max}}$ then
        $c_{thres}$ −= $c_{rate}$
        $n_{success} = 0$
    if $n_{fail} > n_{fail_{max}}$ then
        $c_{thres}$ += $c_{rate}$
        $n_{fail} = 0$
```

This planner inherits the advantages of RRT-like planners such as the exploratory strength which biases the planning towards large Voronoi regions of the configuration space. In addition, the cost-based heuristic biases the tree expansion to low cost areas of the configuration space.

The c_{thres} parameter is updated in Algorithm 3 using a simple heuristic. If $n_{success_{max}}$ configurations have been accepted into either search tree, c_{thres} is decreased. If $n_{fail_{max}}$ consecutive configurations have been rejected from either search tree, c_{thres} is increased. This limits the trees from adding too many high cost configurations while also allowing for the necessary exploration of the configuration space in order to find a path in a short amount of time.

The c_{init} parameter can be set to a low starting value (e.g. 0) to bias the search to low cost configurations. Through Algorithm 3, over a number of iterations the cost will update and start expanding the tree from low cost regions of C_{free}. However, the parameter can also be set to the starting configuration cost to speed up the Human-Aware RRT-Connect algorithm. This allows the robot to start exploring the free space around the starting configuration sooner than if c_{thres} was set to an arbitrary value. Consider a scenario where the human and robot start very close to each other. If c_{thres} was always initialized to a standard value, it could take many iterations for the cost threshold to be increased before being able to move away from the starting configuration. Moreover, in this scenario, if the human is already close to the robot it can be assumed that they may be more comfortable with higher cost trajectories (e.g. ones that have closer minimum separation distances to the human). Thus, setting the cost threshold to the starting configuration cost is acceptable.

3.4 Path Post-processing

The geometric path produced by Algorithm 2 is post-processed to improve path quality and smoothness. A final step is employed to time parametrize the path for execution on the robot.

3.4.1 Random Cost Shortcutting

First a shortcutting method is applied where for a number of iterations, random configuration pairs along the path are selected and a collision free, lower cost, connection is attempted to be made between them. The method is detailed in Algorithm 4.

For shortcutting, the same cost formulation as described in Section 3.2 is used to calculate the cost of configurations along the shortcut path. The shortcut segment replaces the original segment only if the maximum cost out of all configurations along the newly created path segment is less than the maximum cost of the configurations along the original path segment. This is done (as opposed to averaging the cost along the path segment) to ensure that path quality is not adversely effected in the shortcutting phase. As an example, if we consider the distance cost, the average cost along the path could be lower than the original path segment along a shortcut path but, the path could still get very close to the human if a few configurations have large costs. Considering the maximum cost along the segment ensures that all configurations along the shortcut maintain good separation distances.

Algorithm 4 Random Cost Shortcutting

```
Input: Path $P$, number of iterations smoothing_iterations
Output: Path $P$ with shortcutting applied
function RANDOM_SHORTCUTTING($P$)
    for $i = 1$ to smoothing_iterations do
        $(q_1, q_2) \leftarrow P.getTwoRandomConfigs()$
        $shortcut \leftarrow getInterpolatedSegment(q_1, q_2)$
        if $isValid(shortcut)$ then
            if $maxCost(shortcut) \leq maxCost(P.getPathSegment(q_1, q_2))$ then
                $P.replacePathSegment(shortcut, q_1, q_2)$
    Return $P$
```

3.4.2 Low-pass Filtering of Joint Positions

Additionally, since RRT-based planners tend to produce jagged paths, especially when travelling around or close to obstacles, low-pass filtering is applied to smooth the path. After filtering the path, collisions are not checked as the filtering will only remove mild oscillations along the path without altering the overall motion. An example of the low-pass filtering is shown in Figure 3.9 for a single joint of the Panda arm. The blue line indicates the unfiltered positions for the fourth joint of the Panda arm along a path produced by the Human-Aware RRT-Connect planner and the green line is the filtered path.

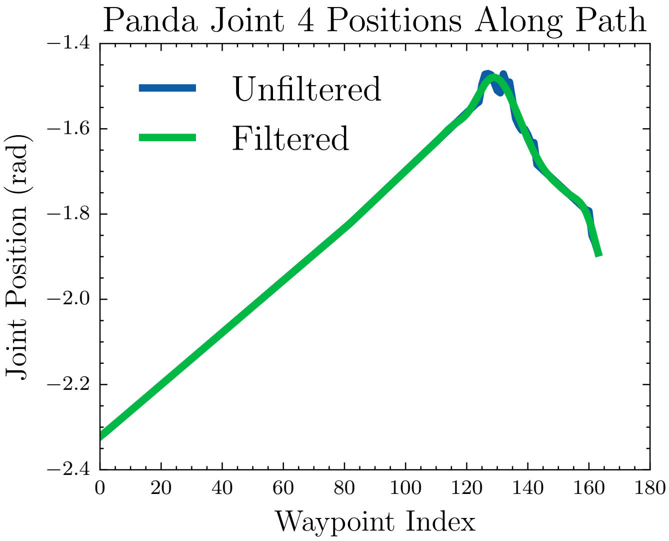


Figure 3.9: Example of low-pass filtering applied to Panda Joint 4 along a path.

3.4.3 Time Parameterization

After shortcutting and low-pass filtering, the path, $\{q_{init}, q_2, q_3, ..., q_{goal}\}$, is time parameterized with velocity and acceleration data using MoveIt's implementation of the Iterative Parabolic Time Parameterization (IPTP) algorithm [3].

Chapter 4

Simulation Experiments

In this chapter, we detail the stimulation experiments run to assess the quality of the proposed Human-Aware RRT-Connect planner proposed in Chapter 3. We compare Human-Aware RRT-Connect to RRT-Connect, the standard implementation detailed in [34], using a set of problem scenarios involving a simulated Franka Emika Panda Arm and a human in a shared human-robot workspace. We also present an analysis of the cost function terms and justify the use of robot points of interest. We end this chapter by comparing Human-Aware RRT-Connect against a variant of T-RRT, Connect T-RRT.

In Section 4.1 the various problem scenarios used to gauge path quality and gather planner metrics are detailed. Section 4.2 explains the implementation details and the planner parameters used for all tests (unless otherwise specified). In Section 4.3, the metrics and path quality criteria are described in detail. In Section 4.4, Human-Aware RRT-Connect is compared against RRT-Connect and the results are presented and analyzed. Section 4.5 presents a sensitivity analysis of the cost function formulation by varying the weights of the cost function terms and the use of points of interest.

4.1 Problem Scenarios

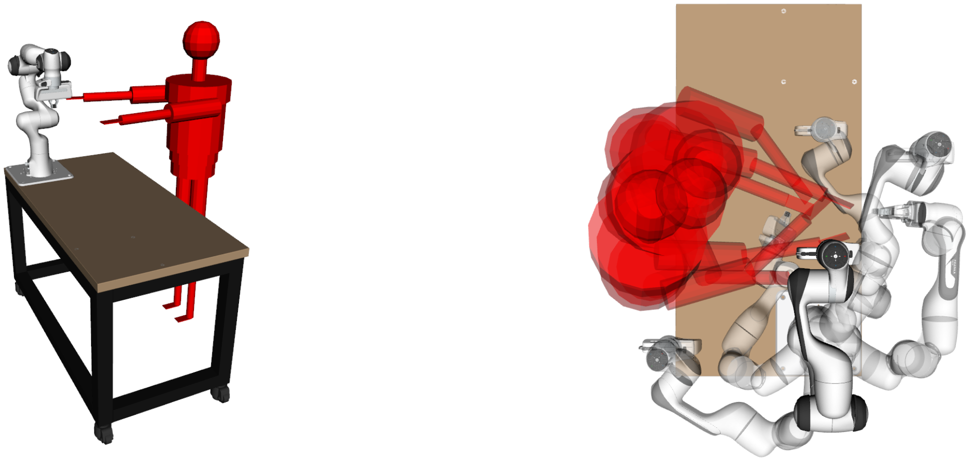
This section details the different testing scenarios used to assess the quality of the proposed Human-Aware RRT-Connect planner. The scenarios are differentiated based on the level of environment and task complexity. The three scenarios presented are described in Table 4.1. The following subsections give more details.

Problem Scenario	Environment	Robot Task
Simple	Human and robot in a shared workspace with no obstacles in the planning scene.	Plan to various joint space goals in the presence of the human.
Cluttered	Human and robot in a shared workspace with various obstacles in the planning scene.	Plan to various joint space goals in the presence of the human.
Cluttered Handover	Human and robot in a shared workspace with various obstacles in the planning scene.	Handover an object to the human.

Table 4.1: The three problem scenarios used to gauge planner quality.

4.1.1 Problem Scenario 1: Simple Environment

In this scenario, the Franka Emika Panda arm is fixed to a table with the human placed in front of the table with outstretched arms. The robot generates plans from its home configuration to various goal configurations that require it to plan around the human. The environment and scenario are shown in Figure 4.1a. The set of start and goal configurations for the robot are listed in Table A.1 in Appendix A. The starting configuration was fixed to q_{init} while the goal configuration was varied over five different joint space configurations, q_{goal_1} through q_{goal_5}. Similarly, a set of three configurations were chosen for the human, listed in Table A.2. The configurations were selected such that the robot would have to plan around the human to reach its goal configuration. The configurations can be seen in Figure 4.1b. These sets of configurations are iterated over 1000 trials when generating the planner results.



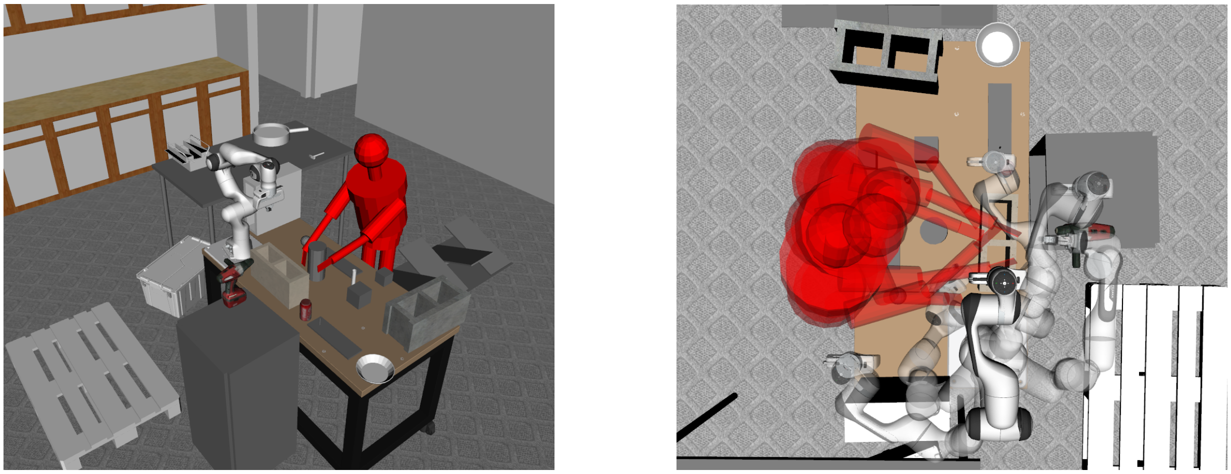
(a) Simple scenario environment.

(b) Simple scenario top view with human and robot configurations shown.

Figure 4.1: The simple human-robot environment scenario showing the Franka Emika Panda mounted on a table with the human statically positioned in the workspace.

4.1.2 Problem Scenario 2: Cluttered Environment

In this scenario, the Franka Emika Panda arm is mounted on a table with the human positioned over the table similar to Problem 1. However, in this case, the environment is cluttered with objects. This scenario aims to test the planner in a more challenging environment where various obstacles are in the robot's planning scene along with the human. The RRT-Connect planner treats all objects and the human as obstacles while the Human-Aware RRT-Connect planner uses the cost function formulation to produce human-aware paths but merely avoids all other obstacles. The environment and scenario are shown in Figure 4.2. The same set of configurations for both the human and the robot were reused from problem scenario 1, with a slight modification made to one of the robot configurations to avoid collision with a static obstacle in the environment. These configurations are detailed in Appendix A and similar to problem scenario 1, are iterated over 1000 trials to generate the planner results.



(a) Cluttered scenario environment.

(b) Cluttered scenario top view with human and robot configurations shown.

Figure 4.2: The cluttered human-robot environment scenario showing the Franka Emika Panda mounted on a table with the human statically positioned in the workspace. Various static obstacles are placed in the environment to simulate a rich HRI scenario.

4.1.3 Problem Scenario 3: Handover in a Cluttered Environment

This scenario uses the same environment as depicted in Figure 4.2a with the task being to handover the drill to the human. The robot's starting configuration is $q_{init_{handover}}$ and its goal configuration is q_{goal_6}, listed in Table A.1. The robot starts with the drill in its end-effector and then approaches the human's left hand to perform the handover. The human's configuration is listed in Table A.2. This planning problem, $(C_{free}, q_{init_{handover}}, q_{goal_6}, c)$, is iterated over 100 trials comparing Human-Aware RRT-Connect to RRT-Connect.

4.2 Implementation Details

In all problem scenarios, the Human-Aware RRT-Connect planner utilizes the cost-function formulation from Section 3.2 to plan in the presence of the human while the RRT-Connect planner only considers the human as an obstacle. The human is assumed to be static during the planning horizon and execution of the robot trajectory.

The human and robot kinematics and dynamics were defined using the Unified Robot Description Format (URDF) and the planners were implemented in Python and utilize

ROS [67] and MoveIt [1]. It should be noted that MoveIt is only used for its Planning Scene interface, and for its forward kinematics service to compute link poses in the world frame. The OctoMap library [30] is used to voxelize the human and compute the euclidean distance transform (implementation from [45]) for the distance cost updates and for collision checking with the human model. The human model voxelization is shown in Figure 4.6.

The results shown in this section have been obtained from simulations run on a computer with an AMD Ryzen 5 2600 CPU running at 3.4 GHz with 16GB of RAM. All planner implementations are single-threaded.

In this work, the planners produce paths of the form $\{q_{init}, q_1, q_2, ..., q_{goal}\} \in C_{free}$ and in a final post-processing step are time parameterized with velocity and acceleration data. The final parameterized trajectory is spline interpolated by a `joint_trajectory_controller` [71] before execution on the (simulated) robot.

Table 4.2 details the various planner parameters common to both Human-Aware RRT-Connect and RRT-Connect. The maximum distance the robot can travel in the joint space without collision checking is denoted as ϵ. The maximum number of iterations that the planner can use to find a plan is denoted `planner_iterations`. If a plan is not found within the number of iterations specified by this parameter, the planner returns failure. The `smoothing_iterations` parameter specifies the number of iterations that Algorithm 4 runs for. It is set to the number of configurations in the path produced by the planner, n.

Parameter	Parameter Value
ϵ	0.02
`planner_iterations`	10000
`smoothing_iterations`	n

Table 4.2: Common parameters for both planners that stay constant throughout all tests.

Table 4.3 shows the parameters used in the cost function formulation presented in Chapter 3. The parameter d_{max} is set to the maximum extent of the Euclidean distance transform which is 2.5m in our implementation. The parameter $d_{max_{com}}$ is similarly set to 2.5m while $d_{min_{com}}$ is set to 0.8m. Finally, to bias the planner to select configurations further away from the human, d_{min} is set to 0.1m. These parameters are held constant throughout all tests.

Parameter	Parameter Value
I_{max}	3.0
d_{max}	2.5
d_{min}	0.1
$d_{min_{com}}$	0.8
$d_{max_{com}}$	2.5

Table 4.3: Parameters for the cost function formulation.

Table 4.4 shows the Human-Aware RRT-Connect planner weights used for the cost function and the planner specific parameters. To bias the planner to find paths that maintain a separation distance of d_{min}, the w_{dist} weighting term for the distance cost was set slightly higher than the visibility and danger criterion costs. The cost function weights are changed in Section 4.5 to illustrate how the parameters affect the planner but for all other tests the values in Table 4.4 are used.

The term used to weight the configuration cost in the nearest neighbour search, α, was set to 1.8 empirically. Setting α to larger values generally leads to safer paths that take longer to plan for. An analysis varying the α parameter and examining various path metrics is shown in Appendix B.

The expansion probability, η, from the NEW_CONFIG function in Algorithm 2 is set to 0.3 empirically. Setting η higher than 0.1 results in faster expansion of the search trees but lower quality paths. Setting η to a small value (e.g., below 0.1), leads to long search times but higher quality paths. An analysis varying the η value and examining various path metrics is shown in Appendix B.

The parameter $n_{fail_{max}}$ is set to 10 to allow enough time for the planner to find configurations of low cost around the current node being extended in the tree. With the strict cost thresholding implemented in the NEW_CONFIG function, the planner sometimes takes several iterations to accept a configuration into the current search tree. By setting $n_{fail_{max}}$ to a value of 10, a compromise between getting trapped at a local minimum of the cost space vs. exploring the surrounding regions of C_{free} is attained. The parameter $n_{success_{max}}$ is set to 2 so that the tree doesn't expand too many nodes at the current cost. The rationale is, if the tree can grow successfully at the current cost, since the acceptance criteria on line 23 of Algorithm 2 follows a downward slope in cost, we want to ensure we follow the slope down to a local minimum instead of accepting nodes at the same cost. The updating of $n_{fail_{max}}$ and $n_{success_{max}}$ after each planning iteration ensures that the algorithm moves out of local minima within a reasonable amount of time, finding a path if one exists.

The cost threshold, c_{thres} is initialized to 0.0 to bias the planner to find high-quality paths. The c_{rate} parameter is set to 0.01.

Parameter	Parameter Value
w_{dist}	0.4
w_{vis}	0.3
w_{dc}	0.3
$n_{success_{max}}$	2
$n_{fail_{max}}$	10
α	1.8
η	0.3
c_{init}	0.0
c_{rate}	0.01

Table 4.4: Human-Aware RRT-Connect weights and planner parameters.

As mentioned in Section 3.2, points of interest can be specified for the c_{dist} and c_{vis} cost function terms. Figure 4.3 shows the various Panda arm link frames with the points of interest used during testing highlighted in green. Since many of the frames overlap, only four points of interest were used: *panda_gripper_center* which is effectively the end-effector frame, *panda_link*7 which covers the wrist area of the arm, *panda_link*4 which is the elbow of the arm, and *panda_link*2 which is close to the base of the arm. These four frames were used for both c_{dist} and c_{vis}.

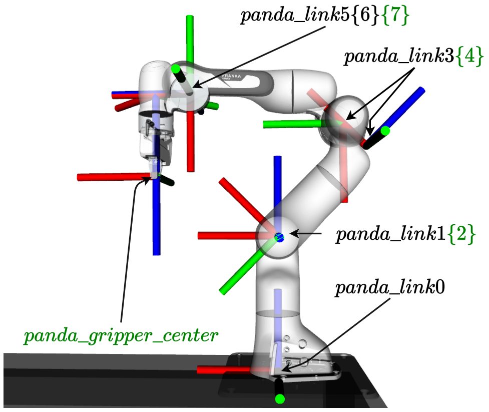


Figure 4.3: The Franka Emika Panda Arm link frames with the points of interest highlighted in green.

4.3 Human-Aware Planning Metrics

To assess planner path quality with regards to human-awareness and safety, a set of metrics were determined and defined. The metrics and their descriptions are detailed in Table 4.5 followed by descriptions of their formulations. As a note with regards to language, we use “separation distance” and “clearance” interchangeably. Similarly, “cone of gaze” and “effective field of view” are used interchangeably.

Metric	Metric Description
Minimum clearance	The minimum separation distance between the robot and the human along the path.
Average clearance	The average separation distance between the robot and the human along the path.
Path length	The end-effector path length of the robot trajectory in meters.
Path visibility	The ratio of the number of configurations that stay in the human’s eFOV to the total number of configurations along the path.
Planning time	The planner planning time.
Number of nodes	The number of nodes produced by the planner during exploration of C_{free}.
Average robot inertia	Average I_s along the robot path.
Mechanical work	Summation of positive cost variations along path.
Integral of the cost	Summation of configuration costs along path.

Table 4.5: Path metrics reported for each planner testing scenario.

As a general method to assess planner path quality, let $c_p := \Sigma_{free} \rightarrow R_{\geq 0}$ denote a path quality criterion where every path in Σ_{free} is assigned a positive real cost value. There are several ways to define c_p with the most common way being the *integral of the cost* along a path [15]. If we let n be the number of subdivisions of the path, we can define the constant step size $\delta = \frac{1}{n}$, and the discrete approximation of the integral of the cost can be defined as:

$$c_p(\sigma) = \frac{length(\sigma)}{n} \sum_{k=1}^{n} c\left(\sigma\left(\frac{k}{n}\right)\right). \tag{4.1}$$

In [32], it was shown that the concept of *mechanical work* can assess path quality better than integral cost. Mechanical work sums the positive cost variation between configurations on the path and can be interpreted as summing the "forces" acting against the motion [15]. If we use the constant step size $\delta = \frac{1}{n}$, the discrete approximation of the mechanical work of a path can be defined as:

$$c_p(\sigma) = \sum_{k=1}^{n} \max\left\{0,\ c\left(\sigma\left(\frac{k}{n}\right)\right) - c\left(\sigma\left(\frac{k-1}{n}\right)\right)\right\}. \tag{4.2}$$

Additionally, the mathematical formulations of some of the other metrics in Table 4.5 are given below. The trajectory length is defined as:

$$path_{len}(\sigma) = \sum_{i=1}^{n} ||FK(q_i) - FK(q_{i-1})||, \tag{4.3}$$

where we denote $FK(q_i)$ as the cartesian position of the robot end-effector at the robot configuration q_i and n as the number of configurations on the path σ.

The minimum clearance distance to the human is defined as:

$$path_{min_{clear}}(\sigma) = \min_j EDT(H, FK(q_i, p_j)), \tag{4.4}$$

where we denote $EDT(H, FK(q_i, p_j))$ as the nearest euclidean distance to collision from the robot to the human calculated by the Euclidean Distance Transform, EDT, at the configuration q_i for the points of interest p_j. The points of interest for this metric are $j \in \{panda\_link2, panda\_link4, panda\_link7, panda\_gripper\_center\}$ as depicted in Figure 4.3.

The average clearance distance to the human is defined as:

$$path_{avg_{clear}}(\sigma) = \frac{\sum_{i=0}^{n} \min_j EDT(H, FK(q_i, p_j))}{n} \tag{4.5}$$

The average robot inertia is defined as:

$$path_{inertia}(\sigma) = \frac{\sum_{i=0}^{n} I_s}{n} \tag{4.6}$$

4.4 Results

This section summarizes and analyzes the results gathered from each problem scenario described in Section 4.1.

4.4.1 Problem Scenario 1

The various metrics detailed in Table 4.5 are plotted in Figure 4.4 comparing Human-Aware RRT-Connect with RRT-Connect for the simple environment depicted in Figure 4.1.

The results show that the robot is able to maintain a larger minimum and average separation distance when planning with the Human-Aware RRT-Connect planner (Figure 4.4a and Figure 4.4b). The minimum separation distance for Human-Aware RRT-Connect is just under the d_{min} value of 0.1m. As a note, some robot goal configurations brought the robot closer than 0.1m (d_{min}) to the human. Other than these situations, the robot is able to adhere to the parameter constraints. The robot is able to maintain a large average separation distance of around 0.21m when executing paths using the Human-Aware RRT-Connect planner.

The robot is also more visible to the human, as shown in Figure 4.4c, when planning with Human-Aware RRT-Connect as opposed to RRT-Connect. Among the set of robot goal configurations chosen for this test, some are completely outside of the human's field of view, but the Human-Aware RRT-Connect planner biases the robot towards the effective field of view of the human before approaching its goal.

Integral cost and mechanical work are minimized when planning with Human-Aware RRT-Connect as shown in Figure 4.4h and Figure 4.4i. Although Human-Aware RRT-Connect produces longer paths than RRT-Connect (as shown in Figure 4.4d), the average integral cost of the paths is still smaller than that of RRT-Connect. Mechanical work is roughly five times lower for Human-Aware RRT-Connect as compared to RRT-Connect partly due to line 23 of Algorithm 2 that tries to accept configurations only if they are of lesser cost than their nearest neighbor.

From Figure 4.4e, a drawback of the Human-Aware RRT-Connect planner is the increase in planning time as compared to RRT-Connect. Since more nodes need to be generated to find configurations of acceptable cost that can form a path, Human-Aware RRT-Connect is in general slower than RRT-Connect.

The average robot inertia between the two planners is roughly the same with Human-Aware RRT-Connect being slightly lower. A larger difference is likely not seen due to the weighting priority put on w_{dist}. For some goal configurations, this has the effect of extending the arm out of compact low inertia configurations in order to maintain larger minimum separation distances.

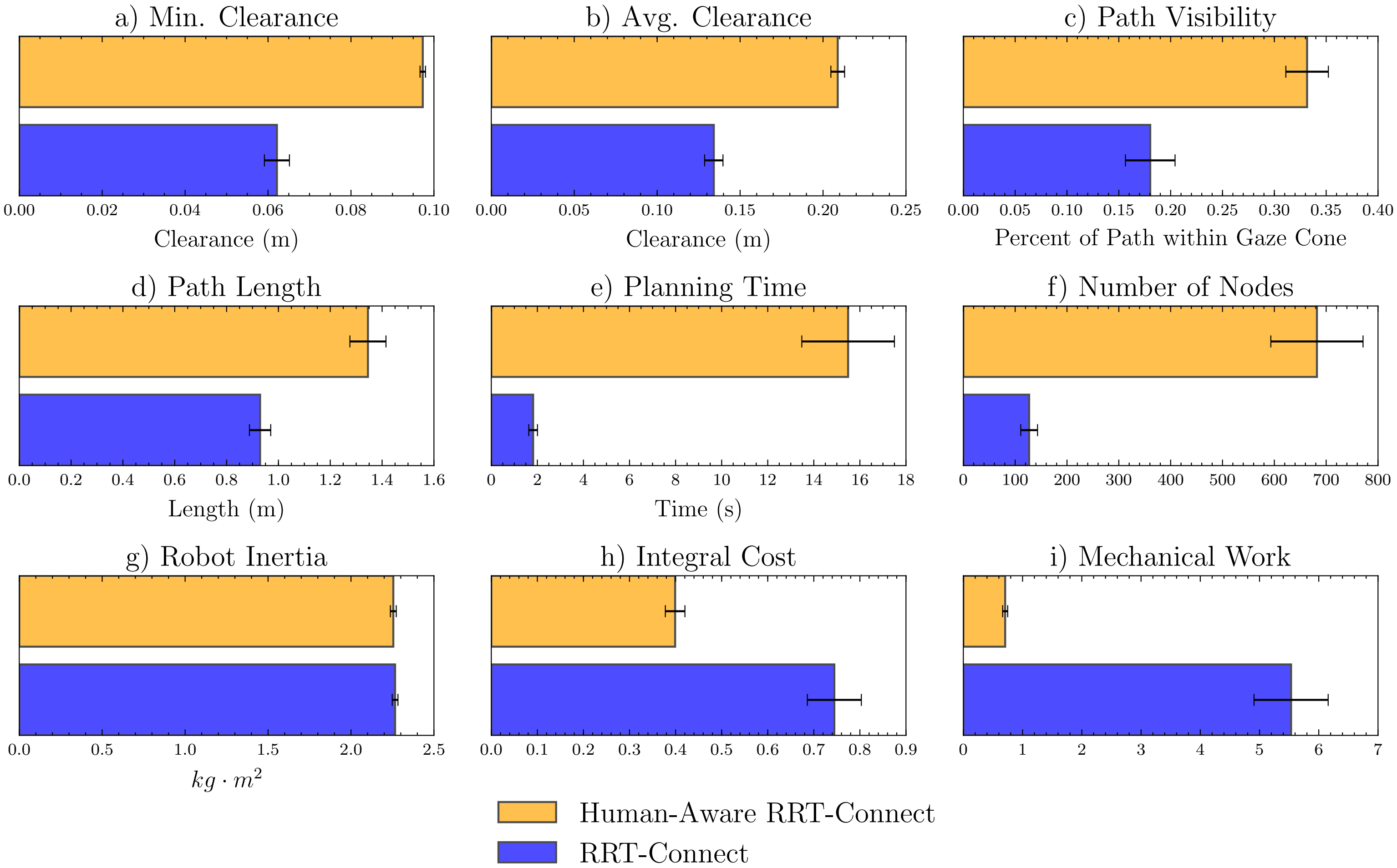


Figure 4.4: The nine metrics calculated for the two planners for problem scenario 1, the simple environment. Each graph shows the average and standard error of the mean (SEM) over 1000 trials.

4.4.2 Problem Scenario 2

In problem scenario 2, several objects are placed in and around the shared human-robot workspace. These range from objects that are represented as simple primitive shapes to more complex meshes as shown in Figure 4.2a. As noted in Section 4.1, problem scenario 2 used the same human configurations as problem scenario 1 and the same robot configurations as problem scenario 1 with slight modifications made to q_{goal_2} to avoid static obstacles in the environment.

The results for this scenario, shown in Figure 4.5, when comparing Human-Aware RRT-Connect to RRT-Connect are similar to the simple scenario. The Human-Aware RRT-Connect planner performs better than RRT-Connect in terms of minimum separation distance, average separation distance and path visibility. The integral cost and mechanical work of paths generated by Human-Aware RRT-Connect in this scenario are also not adversely affected by the clutter on the table. Planning time is slightly increased with a larger standard error of the mean.

From this problem scenario, we see that the additional clutter added into the environment has little effect on the overall performance of the Human-Aware RRT-Connect planner in terms of the cost function formulation used. The primary objective of the planner is to expand the start and goal trees from safer regions with respect to the human and the static objects simply reduce the size of C_{free}. While narrow passages due to static obstacles are not tested, some goal configurations present a narrow cost passage due to their proximity to static obstacles. Configuration q_{goal_1} reaches under the human arm and requires the robot to maintain a separation distance of around 0.11m which it is able to do even in the presence of obstacles such as the cinder block and cylinders.

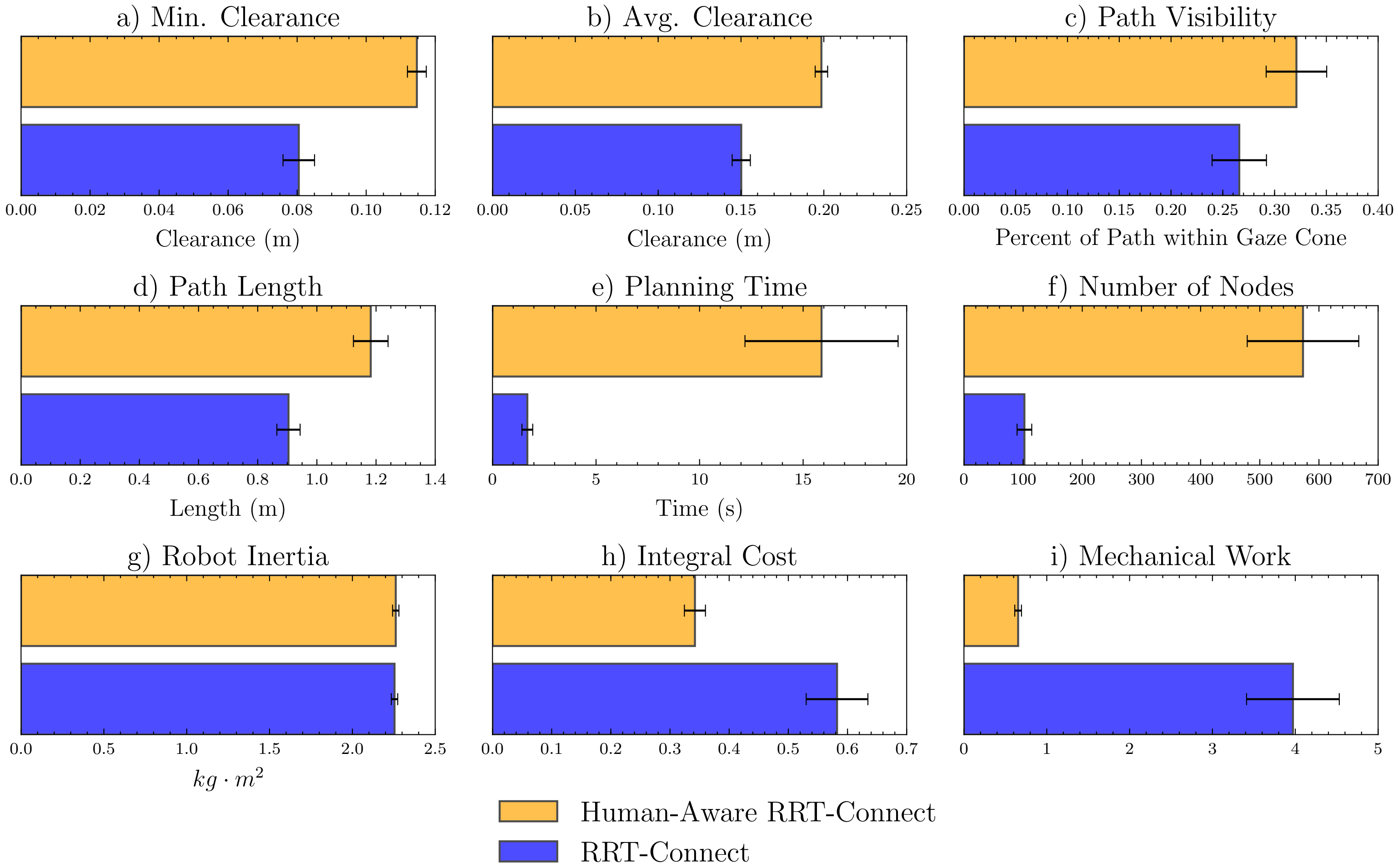


Figure 4.5: The nine metrics calculated for the two planners for problem scenario 2, the cluttered environment. Each graph shows the average and standard error of the mean (SEM) over 1000 trials.

4.4.3 Problem Scenario 3

For the cluttered handover, the environment is the same as depicted in Figure 4.2a and is represented through the MoveIt Planning Scene in Figure 4.6. The OctoMap voxelized human representation from the Planning Scene is also shown. The robot transfers the drill to the human's left hand.

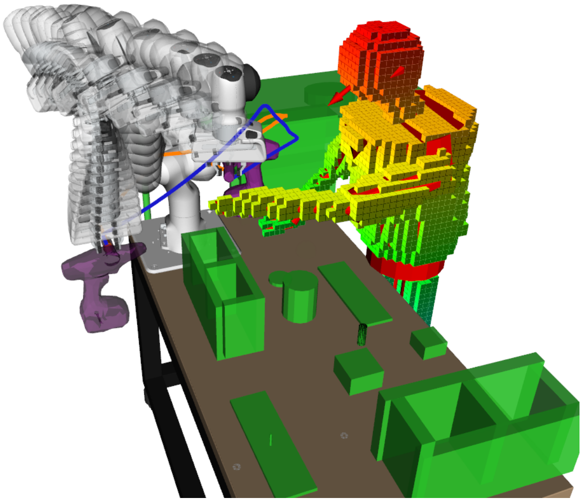


Figure 4.6: An example path generated by Human-Aware RRT-Connect for problem scenario 3, a cluttered handover. The blue path denotes RRT-Connect and the orange path denotes Human-Aware RRT-Connect. The robot path trail is shown for Human-Aware RRT-Connect.

For this particular start and goal configuration pair, the planning time for Human-Aware RRT-Connect is only around one second higher than RRT-Connect as shown in Figure 4.7e. Human-Aware RRT-Connect is able to maintain better performance for the minimum separation distance and average separation distance metrics while maintaining roughly the same path length as compared to RRT-Connect. The blue path denotes RRT-Connect and as can be seen, the path takes a very direct approach and brings the drill close to the human's torso. Human-Aware RRT-Connect first moves upwards from the start configuration and keeps the center of mass of the robot as far as possible from the human before approaching the handoff position. Overall, the results shown in Figure 4.7 show that the Human-Aware RRT-Connect planner can successfully plan a handover (where the robot moves quite close to the human) while minimizing the cost function formulation from Section 3.2.

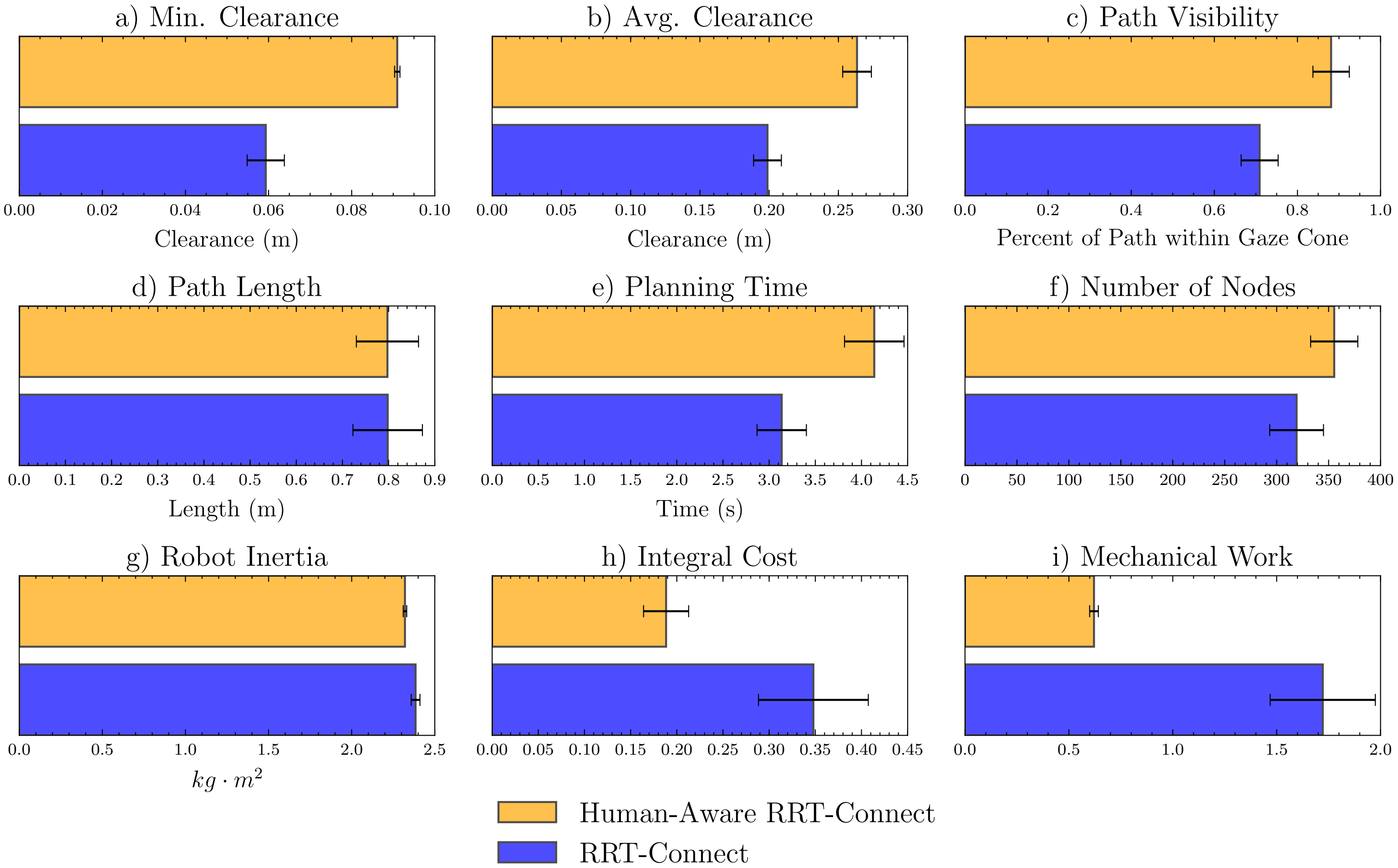


Figure 4.7: The nine metrics calculated for the two planners for problem scenario 3, the cluttered handover. Each graph shows the average and standard error of the mean (SEM) over 100 trials.

4.5 Human-Aware RRT-Connect Planner Sensitivity Analysis

To illustrate the effects of the planner weights, some exemplar paths and metrics are shown in this section with weights altered from those listed in Table 4.4. Additionally, the distance and visibility cost function formulations from this work are compared with the formulation presented in [50]. More specifically, in this work we consider several links on the robot arm as points of interest for the distance and visibility to the human while [50] only considers the end effector.

4.5.1 Planner Weights Sensitivity Analysis

In order to illustrate the effects of each cost function term, three weight sets were created with each giving priority to one of the cost function terms. The weights are shown in Table 4.6. We compare each weight set for the Human-Aware RRT-Connect planner with the baseline RRT-Connect. All tests were performed in the simple environment.

Parameter	Weight Set 1	Weight Set 2	Weight Set 3
w_{dist}	0.8	0.1	0.1
w_{vis}	0.1	0.8	0.1
w_{dc}	0.1	0.1	0.8

Table 4.6: Human-Aware RRT-Connect weight sets for the sensitivity analysis.

Weight Set 1

For this test, w_{dist} is set larger than w_{vis} and w_{dc} to explore the effects of the distance cost function term on the overall formulation. The human configuration was set to $Config.1$ listed in Table A.2 and the robot planned from q_{init} to q_{goal_1} listed in Table A.1. This test was run for 100 trials. Figure 4.8 shows an exemplar trial with the RRT-Connect path shown in blue and the Human-Aware RRT-Connect path shown in orange.

From Figure 4.9b we see that w_{dist} has the effect of quickly biasing the planner to low cost configurations (that achieve high separation distance from the human) even if the start and goal configurations are close to the human. The average separation distance maintained by the Human-Aware RRT-Connect planner is around 0.21m, well above that of RRT-Connect as shown in Figure 4.9a.

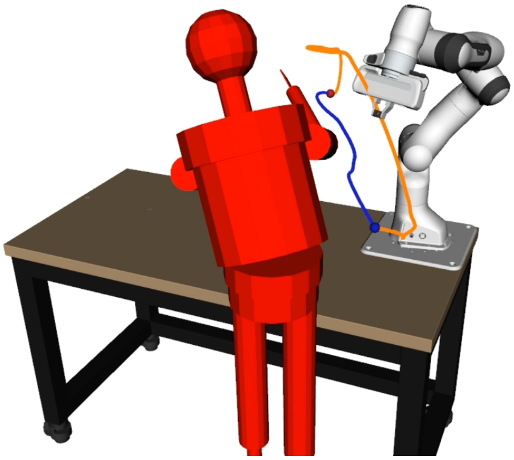
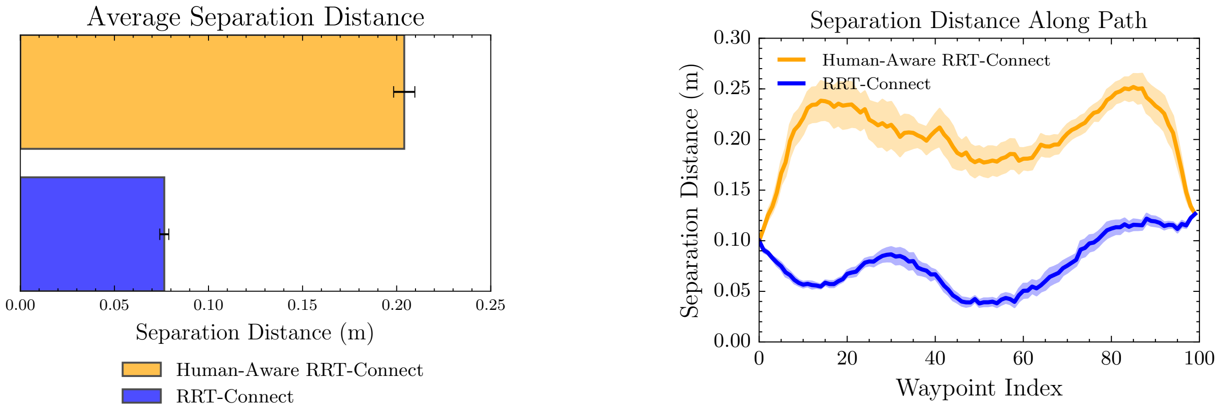


Figure 4.8: Example paths showing Human-Aware RRT-Connect (orange) and RRT-Connect (blue) for Weight Set 1 which has $w_{dist} >> w_{vis} + w_{dc}$.



(a)

(b)

Figure 4.9: The average separation distance and separation distance along the path for Weight Set 1 listed in Table 4.6. The error shown is the standard error of the mean (SEM).

Weight Set 2

Weight Set 2 sets w_{vis} larger than w_{dist} and w_{dc} in order to test the effect of the visibility cost on the overall cost formulation. The human configuration was set to $Config.3$ listed in Table A.2 and the robot planned from q_{init} to q_{goal_1} listed in Table A.1. This test was run for 100 trials and exemplar robot paths are shown in Figure 4.10.

From Figure 4.11b we see that the w_{vis} weight biases the planner to stay in the cone of gaze region (effective field of view) for a longer portion of the trajectory before moving toward the goal. RRT-Connect quickly exits the cone of gaze region whereas the Human-Aware RRT-Connect planner enters it deliberately from the starting configuration. From Figure 4.11a, we see that Human-Aware RRT-Connect is able to maintain around 40% of its path within the human's cone of gaze.

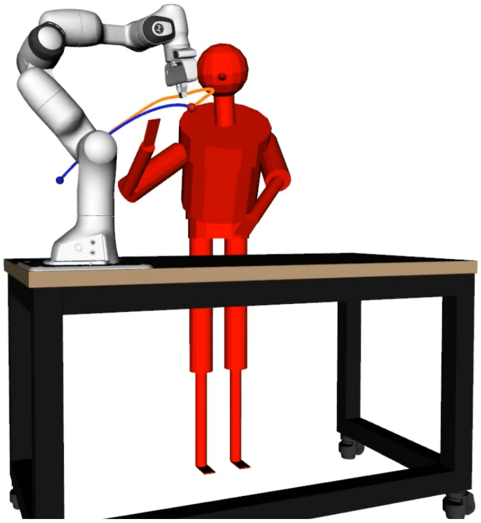
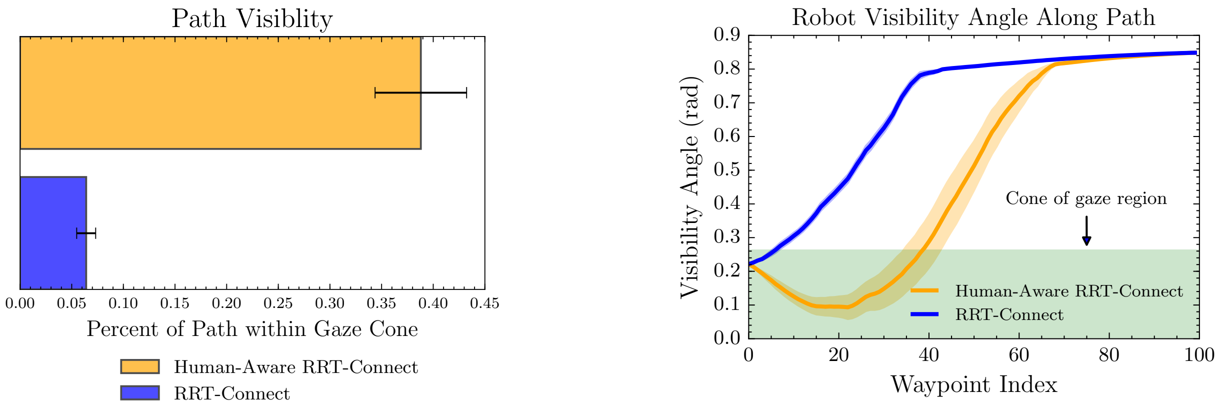


Figure 4.10: Example paths showing Human-Aware RRT-Connect (orange) and RRT-Connect (blue) for Weight Set 2 which has $w_{vis} >> w_{dist} + w_{dc}$.



(a)

(b)

Figure 4.11: The path visibility and visibility angle along the path for Weight Set 2 listed in Table 4.6. The error shown is the standard error of the mean (SEM).

Weight Set 3

Weight Set 3 sets w_{dc} larger than w_{dist} and w_{vis} in order to observe the effect of the danger criterion cost c_{dc}. The human configuration was set to $Config.1$ listed in Table A.2 and the robot planned from q_{init} to q_{goal_2} listed in Table A.1. This test was run for 100 trials and exemplar robot paths are shown in Figure 4.12. This configuration poses a challenging problem for the robot as it has to plan to a position underneath the robot's arm.

From Figure 4.13a we see that Human-Aware RRT-Connect reduces the average robot inertia and CoM distance effectively across the 100 trials. From Figure 4.13b, we see that the human-robot center of mass distance is larger for Human-Aware RRT-Connect along the path. Finally, Figure 4.13c shows that the robot inertia is effectively reduced along the path for Human-Aware RRT-Connect. The robot selects low inertia configurations in the middle portions of the path improving safety for the human in case of unexpected collisions.

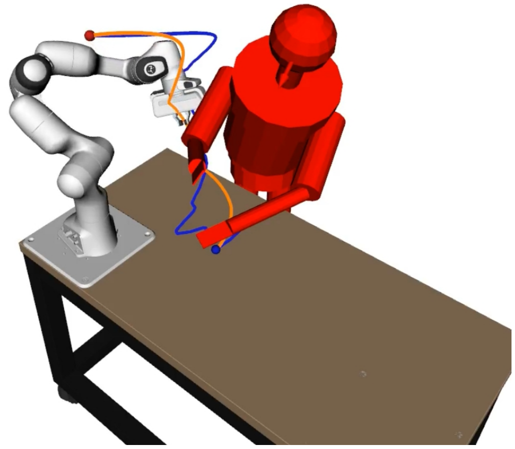
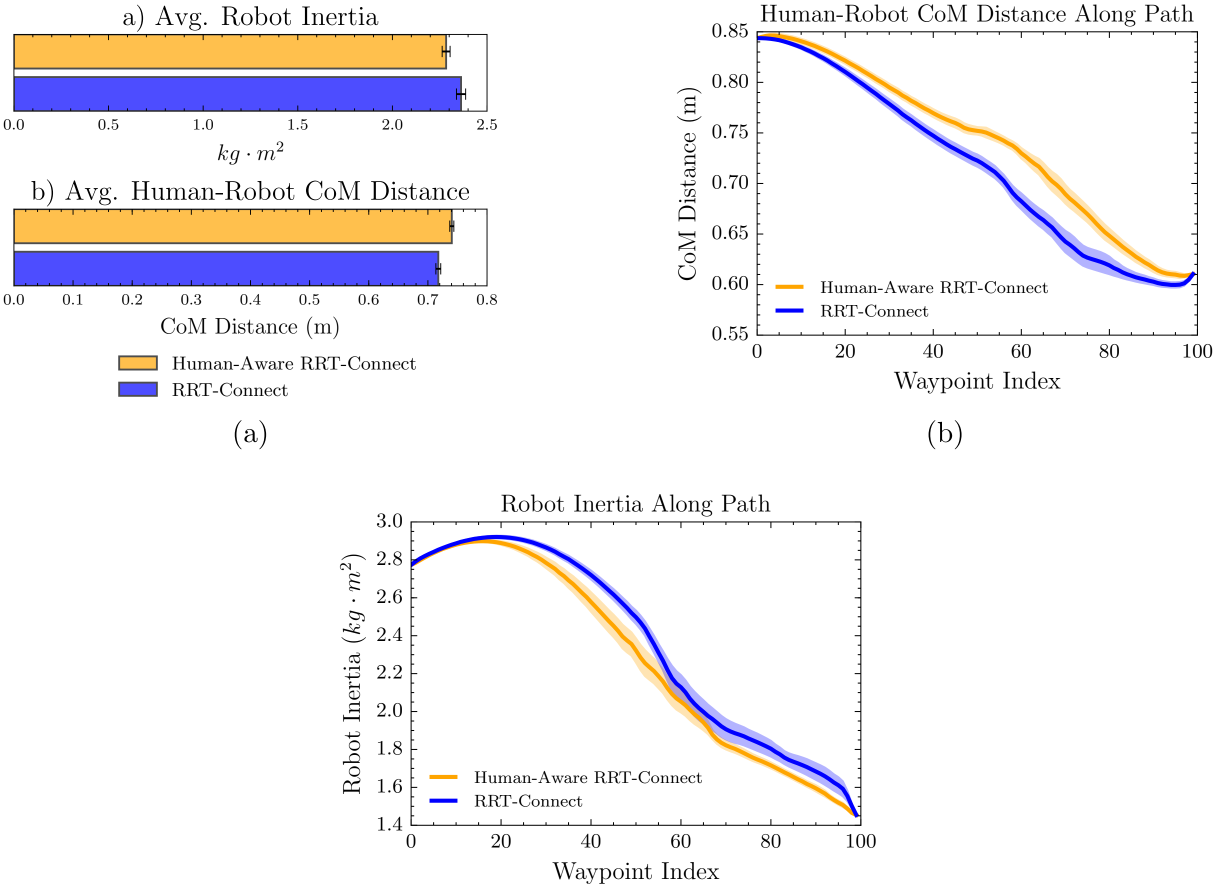


Figure 4.12: Example paths showing Human-Aware RRT-Connect (orange) and RRT-Connect (blue) for Weight Set 3 which has $w_{dc} >> w_{dist} + w_{vis}$.



(a)

(b)

(c)

Figure 4.13: The path visibility and visibility angle along the path for Weight Set 3 listed in Table 4.6. The error shown is the standard error of the mean (SEM).

4.5.2 Distance Cost Formulation Analysis

For the planning tests presented in this section, the distance cost function considers all joints of the robot arm as points of interest (POI) when computing the distance to human. In this subsection we show that this is more effective than only considering the end-effector distance to the human as was done in [50].

For this test, the robot started at q_{init} and planned to q_{goal_2} while the human was positioned using $Config.1$. These configurations are listed in Table A.1 and Table A.2. The weights and planner parameters were set to Weight Set 1 from Table 4.6 to bias the planner to search for configurations that maintain a large separation distance from the human. The test was run for 100 trials.

From Figure 4.15a we see that Human-Aware RRT-Connect using points of interest performs better than only using the end effector for the minimum separation distance, average separation distance and average path length metrics. The path length is reduced when using points of interest because the planner works to keep away different portions of the robot from the human such as the elbow or the wrist. For this specific pair of start and goal configurations, the effect of using points of interest is seen near the end of the path where the robot is able to keep its wrist away from the human's body. This can be seen visually in Figure 4.14 and quantitatively in Figure 4.15b where the separation distance along the path is plotted. Starting from roughly waypoint index 70, the Human-Aware RRT-Connect planner using points of interest maintains a larger separation distance from the human as compared to the Human-Aware RRT-Connect planner that only uses the end effector for the distance cost calculation.

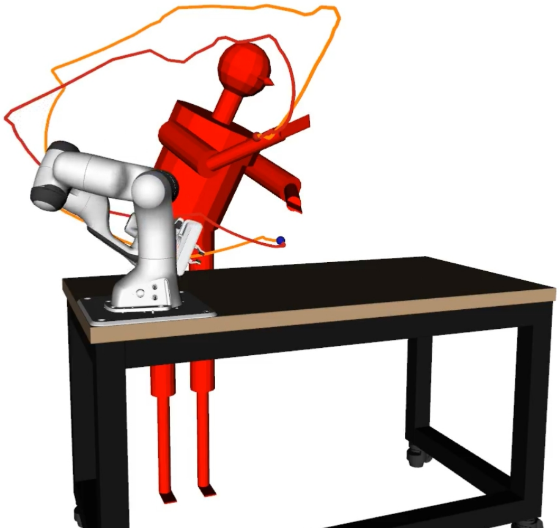
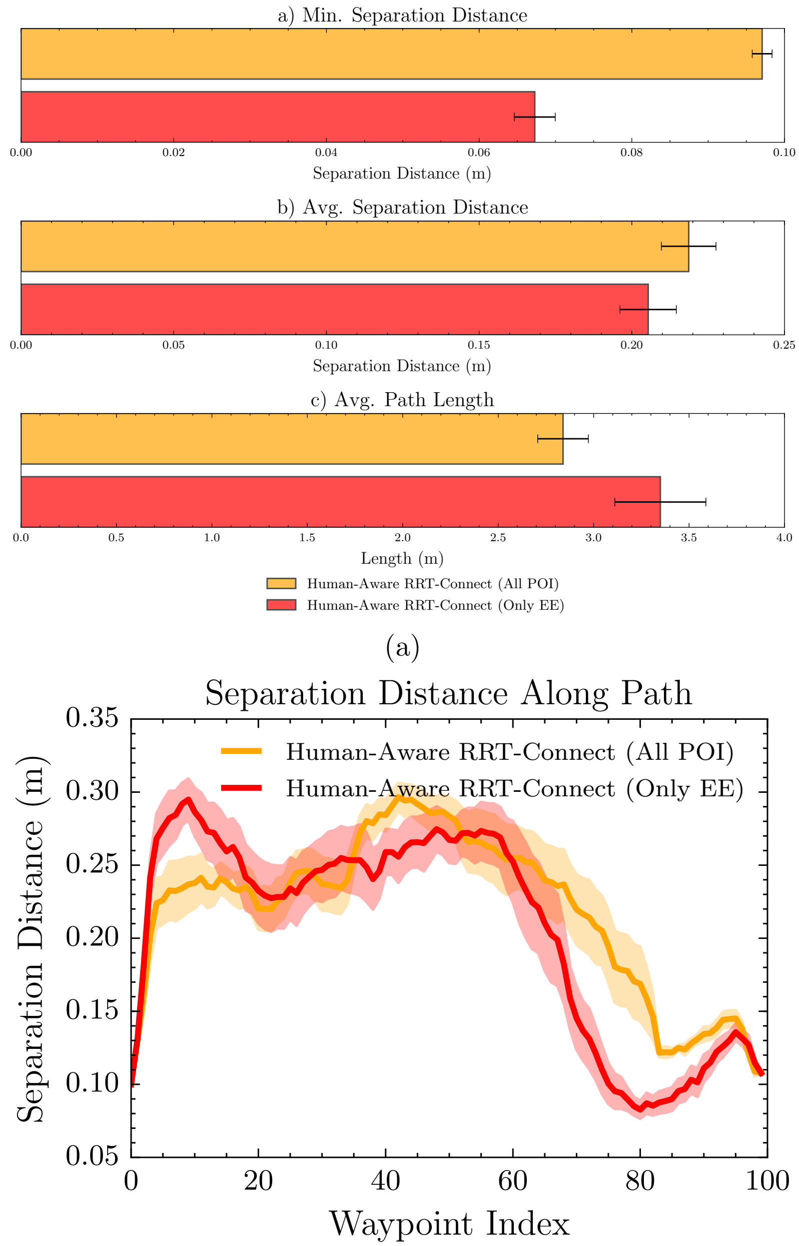


Figure 4.14: Example paths showing the use of points of interest (orange) vs. only using the end effector (red) for Weight Set 1.



(a)

(b)

Figure 4.15: The a) minimum separation distance, average separation distance and average path length and b) separation distance along the path when planning with Human-Aware RRT-Connect using points of interest (orange) vs. Human-Aware RRT-Connect using only the end effector (red). The error shown is the standard error of the mean (SEM).

4.5.3 Visibility Cost Formulation Analysis

We similarly analyze the use of points of interest against only using the end-effector for the visibility cost. For this test, the robot started at q_{init} and planned to q_{goal_1} while the human was positioned using $Config.3$. These configurations are listed in Table A.1 and Table A.2. The weights and planner parameters were set to Weight Set 2 from Table 4.6 to bias the planner to search for configurations that are more visible to the human. The test was run for 100 trials.

From Figure 4.17 and the exemplar paths shown in Figure 4.16, we see that the use of points of interest biases the planner to stay in the cone of gaze region longer than using only the end-effector. This start and goal configuration pair is a good example because the goal configuration is much outside the human's field of view.

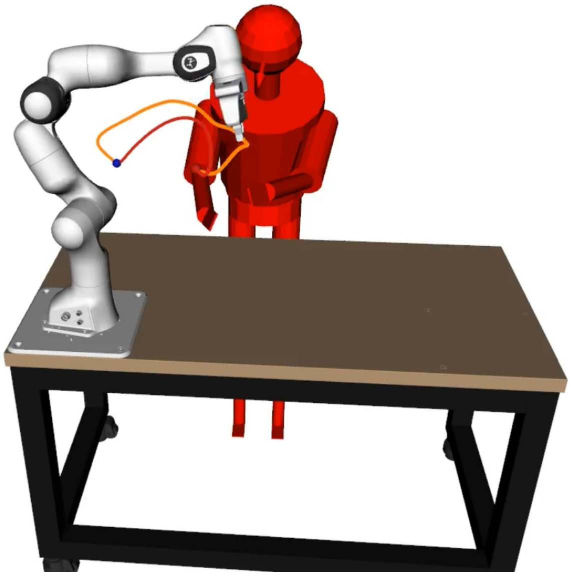
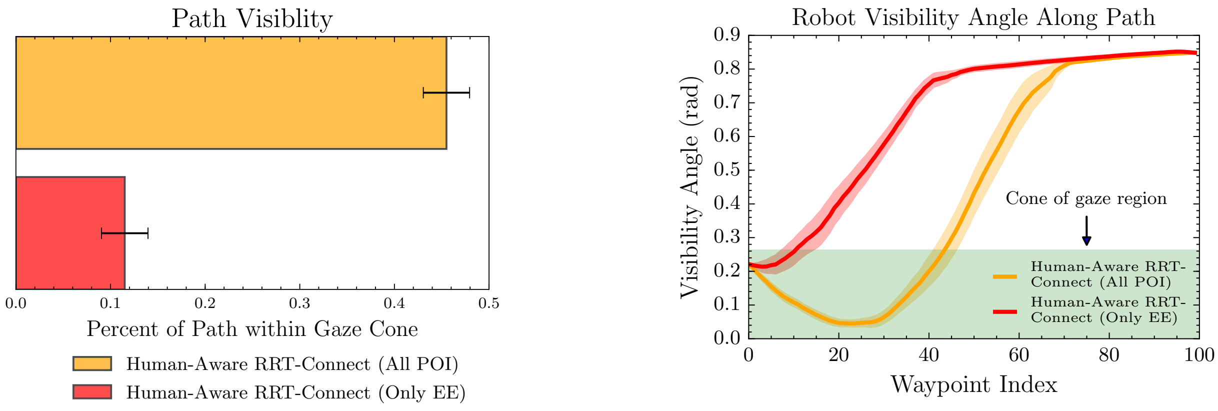


Figure 4.16: Example paths showing the use of points of interest (orange) vs. only using the end effector (red) for Weight Set 2.



(a) Path visibility.

(b) Visibility angle along path.

Figure 4.17: The path visibility and visibility angle along the path when planning with Human-Aware RRT-Connect using points of interest (orange) vs. Human-Aware RRT-Connect using only the end effector (red). The error shown is the standard error of the mean (SEM).

4.6 Comparison with Connect T-RRT

To make a more direct comparison with related work, we also evaluated Human-Aware RRT-Connect against a variant of T-RRT [32], the planner used in [50]. As a note, in [50], the cost function involved a distance cost, visibility cost and a human comfort cost. This thesis involved variants of the distance and visibility costs but the human comfort cost was not used since the cost function developed here was not specific to a handover task. As such, the comparison here is more directly between Human-Aware RRT-Connect and the T-RRT variant. This is a useful comparison since T-RRT (and its variants) have been shown to be able to plan using cost functions defined over many different configuration spaces [32].

For this comparison, we use the variant Connect T-RRT [16] to more closely match the Human-Aware RRT-Connect planner. The cost function formulation detailed in Section 3.2 was used within both planners. The original T-RRT algorithm, a planner introduced to solve general cost space planning problems, grew one tree rooted at the start configuration and differed from the original RRT algorithm by introducing a *transition test* that aimed to filter configurations based on stochastic optimization techniques. Connect T-RRT uses the same techniques as the original T-RRT algorithm but grows two trees, one from the starting configuration and one from the goal configuration, and tries to connect the trees by iterating the tree extension and transition test as long as there are no collisions and the transition test passes. This planner variant is detailed in Appendix C, Algorithm 5.

As noted in [16], the $T_{rate} \in (0, 1]$ parameter determines the trade-off between low computation time and high path quality. A value not too small, such as 0.1, leads to a greedy search while a value such as 0.01, leads to higher quality paths. Through empirical testing, we found that a value of 0.01 produces high quality paths at the expense of very long planning times. As such, to trade-off between path quality and planning time, we used a value of 0.1 for T_{rate}. From the results presented later in this section, we see that this value produces high quality paths within reasonable planning times.

The Connect T-RRT planner parameters are described in Table 4.7. The temperature, T, is initialized to a small value T_{init}. The Connect T-RRT planner used the same cost function parameters and weights as described in Table 4.3 and Table 4.4. Human-Aware RRT-Connect also used all the same parameters as were detailed in Table 4.3 and Table 4.4.

Parameter	Parameter Value
ϵ	0.02
c_{max}	0.9
T_{init}	$1e^{-6}$
T_{rate}	0.1

Table 4.7: Connect T-RRT planner parameters.

When comparing Connect T-RRT to Human-Aware RRT-Connect we used problem scenario 1, detailed in Subsection 4.1.1, where a simple environment was instantiated, and 3 different human configurations and 5 different robot configurations were cycled through for 1000 iterations. Figure 4.18 shows the results for this test.

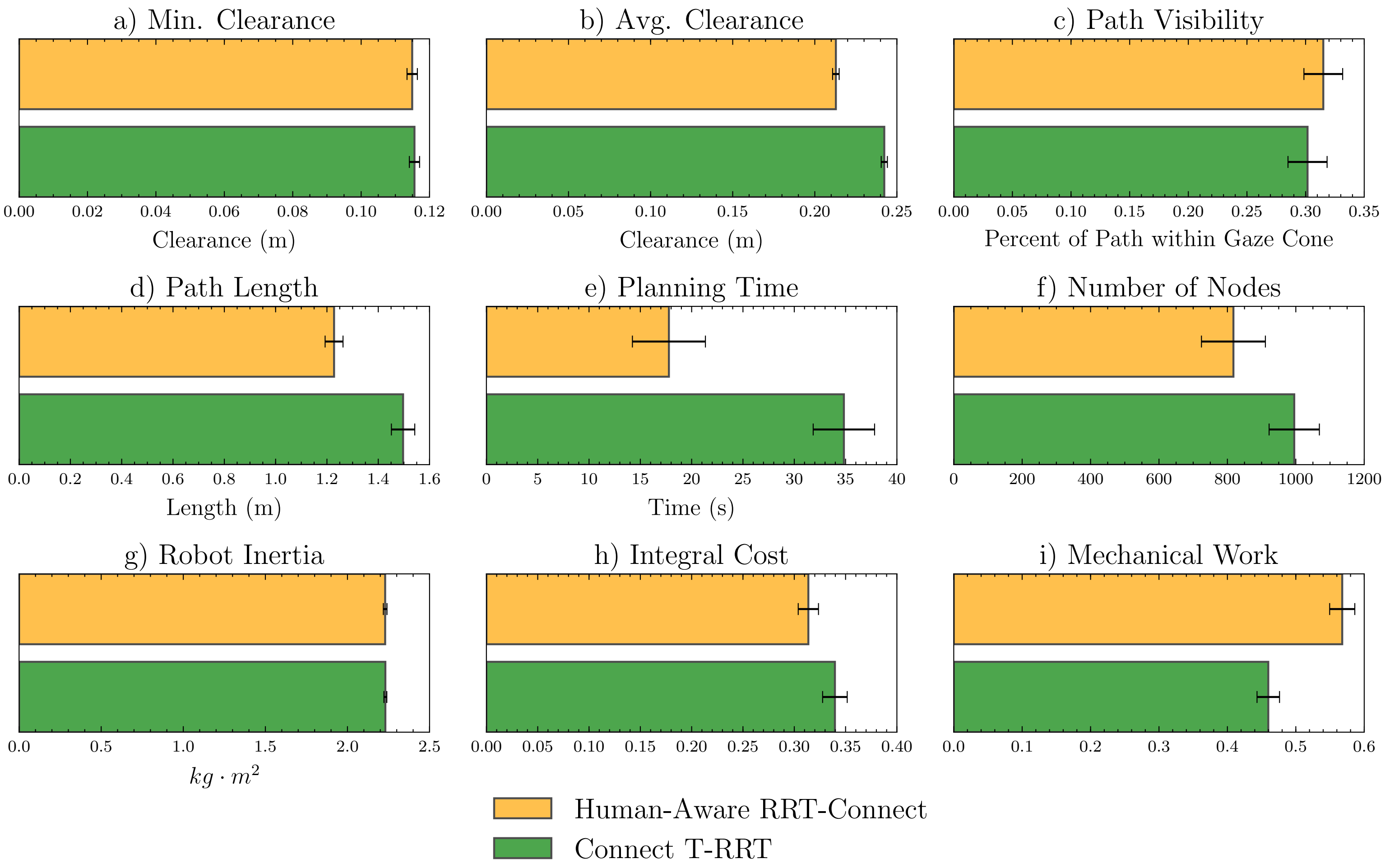


Figure 4.18: The nine metrics calculated to compare Connect T-RRT against Human-Aware RRT-Connect using problem scenario 1. Each graph shows the average and standard error of the mean (SEM) over 1000 trials.

From the results presented in Figure 4.18 we can see that both Connect T-RRT and Human-Aware RRT-Connect produce paths of similar quality with Connect T-RRT having a slightly lower mechanical work (better path quality) as shown in Figure 4.18i. Path visibility is also similar with Human-Aware RRT-Connect having a slightly higher average visibility. The average separation distance for Connect T-RRT is around 3cm higher than Human-Aware RRT-Connect while the minimum separation distance is roughly the same among the two planners.

From the reported metrics, there are a few major differences. First, the average path length for Human-Aware RRT-Connect is around 25cm shorter than Connect T-RRT. This is noteworthy since the shorter path length did not have too much of a negative impact on other metrics with relation to Connect T-RRT. Second, the average planning time for Human-Aware RRT-Connect is roughly half of what is reported for Connect T-RRT. This may be because the planner parameters used for Human-Aware RRT-Connect lead to a more greedy search (and thus results in a slightly lower path quality than Connect T-RRT) but this result shows promise in that high path qualities can be achieved at lower planning times when using Human-Aware RRT-Connect.

As a note, these results are presented for a specific setting of the T_{rate} parameter and a variant of the original T-RRT algorithm. While efforts were made to optimize the planner for a fair comparison with Human-Aware RRT-Connect, further investigation may be needed to more closely match the various parameters in Human-Aware RRT-Connect. Overall, Human-Aware RRT-Connect could be a viable alternative to Connect T-RRT (and other T-RRT variants) where high quality paths are required at reasonable planning times and shorter path lengths.

Chapter 5

Conclusions

5.1 Insights from Implementation

This thesis introduced an HRI cost formulation and sampling-based planner to plan in the presence of humans. The cost formulation included a distance cost, visibility cost and danger criterion cost. The distance cost ensured that the robot maintained a safe distance from the human during planning. The visibility cost aimed to bias the robot toward the human's effective field of view. The danger criterion cost was based on the robot inertia and the relative human-robot centre of mass distance. These individual cost terms were combined in a weighted sum. We proposed and implemented the Human-Aware RRT-Connect planner which plans in the presence of a human using the proposed cost formulation. The planner is based on the original RRT-Connect algorithm with modifications made to the way the search trees are expanded (taking inspiration from [42] and [32]). Configurations are accepted or rejected based on a cost threshold and a mechanism to update this threshold is outlined.

The proposed Human-Aware RRT-Connect planner produced high quality paths in terms of general path quality criteria (i.e., integral cost and mechanical work) and in terms of metrics related to human-robot collaboration such as minimum separation distance, average separation distance, visibility and robot inertia. We also showed that the use of points of interest on the robot performed better than solely using the end-effector while planning with Human-Aware RRT-Connect using the proposed cost formulation. Human-Aware RRT-Connect was compared with Connect T-RRT and it was found that Human-Aware RRT-Connect had faster average planning times with similar path qualities.

5.2 Limitations

While Human-Aware RRT-Connect produces high quality paths with respect to the HRI cost formulation presented in Section 3.2, there are several limitations of the proposed human-aware planning approach. One limitation is that there are no guarantees that the planner will find the optimal solution since the algorithm returns after finding an initial path that connects the start and goal trees. Planning time is another limitation of the current approach. Since, on average, planning takes several seconds, usage in a real time system would not be possible. Moreover, this work assumed the human was static during the planning horizon and trajectory execution. This is generally not the case in real world applications.

Motion consistency is also an issue. Ideally, when queried with the same planning problem, $(C_{free}, q_{init}, q_{goal}, c)$, the planner should return the same path. However, this is not the case with sampling-based planners. In a human-robot shared workspace, this could be an important criteria for human comfort and trust in the robot.

Finally, the planner requires several parameters to be set and while some guidelines have been given in this thesis, a more in depth study into the interaction between them should be investigated.

5.3 Future Work

To make motions more consistent, approaches such as the one used in [28] can be employed where previously occupied workspace voxels are used to inform the planner on which areas of the workspace to avoid and which areas to reuse. Approaches involving motion databases and Experience Based Motion Planners [14] can be used to save and recall previously planned trajectories. In this approach, a fast planner can be used to repair stored paths online based on the human's current position in the workspace.

In this work we extended RRT-Connect to develop the Human-Aware RRT-Connect planner. However, even though RRT-Connect is known to produce paths quickly, imposing cost constraints during sampling slows it down significantly. One alternate approach to speed up the nearest-neighbor search shown in line 13 of Algorithm 2 is to maintain a k-d tree of all the search tree nodes and then use it to find the nearest neighbor of lowest cost within a radius of the returned nearest neighbor. While this is not equivalent to the current implementation, the trade off between speed and path quality can be investigated.

The weights of the cost function formulation in Section 3.2 were set empirically based on the desired planning behaviour. In order to set the weights more optimally based on the context and desired behaviour, Inverse Optimal Control can be used. However, since the cost formulation is a min-max problem (for the distance and visibility cost), the non-differentiability of the max function presents some difficulty in finding an optimal solution. A technique such as the one presented in [23] can be used to formulate a smooth approximation of the problem to find an optimal solution.

An important item to address for future work is the planner's use in a real world task. The techniques presented in this thesis can ensure safety and robot human-awareness during motion planning, but whether this translates to increased productivity and efficiency for the human-robot team should be investigated. A study similar to [44] can be conducted with the proposed planner to observe its effect in a real human-robot shared workspace task. Taking on this endeavour will require integration work to accurately capture the human pose, center of mass and gaze direction. The metrics reported in this thesis are useful in assessing the planner path quality but additional metrics relevant to evaluating fluency [29] in human-robot collaboration should be reported when running a user study with the proposed planner.

References

[1] Moveit. http://moveit.ros.org. Accessed: 2020-03-23.

[2] rosbag. http://wiki.ros.org/rosbag. Accessed: 2020-03-23.

[3] Time parameterization. http://docs.ros.org/kinetic/api/moveit_tutorials/html/doc/time_parameterization/time_parameterization_tutorial.html. Accessed: 2020-06-30.

[4] Daphne Aeraiz-Bekkis, Gowrishankar Ganesh, Eiichi Yoshida, and Natsuki Yamanobe. Robot movement uncertainty determines human discomfort in co-worker scenarios. pages 59–66, 04 2020.

[5] Arash Ajoudani, Andrea Maria Zanchettin, Serena Ivaldi, Alin Albu-Schäffer, Kazuhiro Kosuge, and Oussama Khatib. Progress and prospects of the human–robot collaboration. *Autonomous Robots*, 42(5):957–975, 2018.

[6] T. Arai, R. Kato, and M. Fujita. Assessment of operator stress induced by robot collaboration in assembly. *CIRP Annals*, 59(1):5 – 8, 2010.

[7] Maren Bennewitz, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. Learning motion patterns of people for compliant robot motion. *The International Journal of Robotics Research*, 24(1):31–48, 2005.

[8] Dmitry Berenson, Pieter Abbeel, and Kenneth Y. Goldberg. A robot path planning framework that learns from experience. *2012 IEEE International Conference on Robotics and Automation*, pages 3671–3678, 2012.

[9] Xavier Broquère, Alberto Finzi, Jim Mainprice, Silvia Rossi, Daniel Sidobre, and Mariacarla Staffa. An attentional approach to human–robot interactive manipulation. *International Journal of Social Robotics*, 6:533–553, 2014.

[10] F. Bullo and S. L. Smith. *Lectures on Robotic Planning and Kinematics*. 2019.

[11] John F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA, USA, 1988.

[12] Howie Choset. Robotic motion planning: Rrts.

[13] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.

[14] David Coleman, Ioan Alexandru Sucan, Mark Moll, Kei Okada, and Nikolaus Correll. Experience-based planning with sparse roadmap spanners. *CoRR*, abs/1410.1950, 2014.

[15] D. Devaurs, T. Siméon, and J. Cortés. Optimal path planning in complex cost spaces with sampling-based algorithms. *IEEE Transactions on Automation Science and Engineering*, 13(2):415–424, 2016.

[16] Didier Devaurs. *Extensions of Sampling-based Approaches to Path Planning in Complex Cost Spaces: Applications to Robotics and Structural Biology*. PhD thesis, 10 2014.

[17] Didier Devaurs, Thierry Siméon, and Juan Cortés. Enhancing the transition-based rrt to deal with complex cost spaces. 05 2013.

[18] E. W. Dijkstra. A note on two problems in connexion with graphs. *NUMERISCHE MATHEMATIK*, 1(1):269–271, 1959.

[19] A. D. Dragan, S. Bauman, J. Forlizzi, and S. S. Srinivasa. Effects of robot motion on human-robot collaboration. In *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 51–58, 2015.

[20] A. D. Dragan, K. C. T. Lee, and S. S. Srinivasa. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 301–308, 2013.

[21] Anca Dragan and Siddhartha Srinivasa. Generating legible motion. 06 2013.

[22] A. Ettlin and H. Bleuler. Rough-terrain robot motion planning based on obstacleness. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6, 2006.

[23] S.-C. Fang and Soon-Yi Wu. Solving min-max problems and linear semi-infinite programs. *Computers & Mathematics with Applications*, 32(6):87–93, September 1996.

[24] M. Faroni, M. Beschi, and N. Pedrocchi. An mpc framework for online motion planning in human-robot collaborative tasks. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1555–1558, 2019.

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

[26] Manuel Fernandez Carmona, Tejas Parekh, and Marc Hanheide. Making the case for human-aware navigation in warehouses. In Kaspar Althoefer, Jelizaveta Konstantinova, and Ketao Zhang, editors, *Towards Autonomous Robotic Systems*, pages 449–453, Cham, 2019. Springer International Publishing.

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

[28] Rafi Hayne, Ruikun Luo, and Dmitry Berenson. Considering avoidance and consistency in motion planning for human-robot manipulation in a shared workspace. In Danica Kragic, Antonio Bicchi, and Alessandro De Luca, editors, *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pages 3948–3954. IEEE, 2016.

[29] G. Hoffman. Evaluating fluency in human–robot collaboration. *IEEE Transactions on Human-Machine Systems*, 49(3):209–218, 2019.

[30] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. Software available at http://octomap.github.com.

[31] Flowers Laboratory Inria. human_moveit_config. https://github.com/baxter-flowers/human_moveit_config, 2019.

[32] L. Jaillet, J. Cortés, and T. Siméon. Sampling-based path planning on configuration-space costmaps. *IEEE Transactions on Robotics*, 26(4):635–646, 2010.

[33] L. Jaillet and T. Simeon. A prm-based motion planner for dynamically changing environments. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 2, pages 1606–1611 vol.2, 2004.

[34] James J. Kuffner Jr. and Steven M. Lavalle. Rrt-connect: An efficient approach to single-query path planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 995–1001, 2000.

[35] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. Stomp: Stochastic trajectory optimization for motion planning. pages 4569–4574, 05 2011.

[36] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30:846 – 894, 2011.

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

[38] Arshia Khan and Yumna Anwar. Robots in healthcare: A survey. In *Science and Information Conference*, pages 280–292. Springer, 2019.

[39] K. Kittiampon and J. E. Sneckenberger. A safety control system for a robotic workstation. In *1985 American Control Conference*, pages 1463–1465, 1985.

[40] Thibault Kruse, Amit Kumar Pandey, Rachid Alami, and Alexandra Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726 – 1743, 2013.

[41] Dana Kulic and Elizabeth Croft. Safe planning for human-robot interaction. *J. Field Robotics*, 22:383–396, 01 2005.

[42] Bakir Lacevic, Paolo Rocco, and Morten Strandberg. Safe motion planning for articulated robots using rrts. *2011 XXIII International Symposium on Information, Communication and Automation Technologies*, pages 1–7, 2011.

[43] P. A. Lasota, T. Song, and J. A. Shah. *A Survey of Methods for Safe Human-Robot Interaction.* 2017.

[44] Przemyslaw A. Lasota and Julie A. Shah. Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration. *Human Factors*, 57(1):21–33, 2015. PMID: 25790568.

[45] Boris Lau, Christoph Sprunk, and Wolfram Burgard. Efficient grid-based spatial representations for robot navigation in dynamic environments. *Robotics and Autonomous Systems (RAS)*, 2012. Accepted for publication. Software available at http://octomap.sf.net/.

[46] Steven M. LaValle. Rapidly-exploring random trees : a new tool for path planning. 1998.

[47] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, USA, 2006.

[48] Steven M. LaValle. Motion planning : The essentials. 2011.

[49] Steven M. LaValle, Michael S. Branicky, and Stephen R. Lindemann. On the relationship between classical grid search and probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):673–692, 2004.

[50] Jim Mainprice, Emrah Akin Sisbot, Leonard Jaillet, Juan Cortés, Rachid Alami, and Thierry Siméon. Planning human-aware motions using a sampling-based costmap planner. In *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*, pages 5012–5017. IEEE, 2011.

[51] Y. Morales, A. Watanabe, F. Ferreri, J. Even, T. Ikeda, K. Shinozawa, T. Miyashita, and N. Hagita. Including human factors for planning comfortable paths. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6153–6159, 2015.

[52] Calvin K.L. Or, Vincent G. Duffy, and Chui Chui Cheung. Perception of safe robot idle time in virtual reality and real industrial environments. *International Journal of Industrial Ergonomics*, 39(5):807 – 812, 2009.

[53] Luka Petrovic. Motion planning in high-dimensional spaces. *CoRR*, abs/1806.07457, 2018.

[54] Luka Petrovic. Motion planning in high-dimensional spaces. *CoRR*, abs/1806.07457, 2018.

[55] Tobias Puetz. Enabling robots to achieve new levels of factory automation, 2017.

[56] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494, 2009.

[57] S. Robla, Victor Becerra, J. LLata, Esther Gonzalez-Sarabia, Carlos Torre-Ferrero, and J. Pérez-Oria. Working together: A review on safe human-robot collaboration in industrial environments. *IEEE Access*, PP:1–1, 11 2017.

[58] T. Sasaki and H. Hashimoto. Human observation based mobile robot navigation in intelligent space. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1044–1049, 2006.

[59] John Schulman, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. 06 2013.

[60] A. Shkolnik and R. Tedrake. Path planning in 1000+ dimensions using a task-space voronoi bias. In *2009 IEEE International Conference on Robotics and Automation*, pages 2061–2067, 2009.

[61] Manshi Shukla and Amar Nath Shukla. Growth of robotics industry early in 21st century. *International Journal of Computational Engineering Research (IJCER) Vol. 2, Issue 5*, pages 1554–1558, 2012.

[62] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.

[63] E. A. Sisbot, L. F. Marin, and R. Alami. Spatial reasoning for human robot interaction. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2281–2287, 2007.

[64] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883, 2007.

[65] Emrah Sisbot, Luis Marin-Urias, Xavier Broquère, Daniel Sidobre, and Rachid Alami. Synthesizing robot motions adapted to human presence - a planning and control framework for safe and socially acceptable robot motions. *I. J. Social Robotics*, 2:329–343, 09 2010.

[66] Emrah Akin Sisbot. *Towards human-aware robot motions*. Theses, Université Paul Sabatier - Toulouse III, November 2008.

[67] Stanford Artificial Intelligence Laboratory et al. Robotic operating system.

[68] M. Stolle, H. Tappeiner, J. Chestnutt, and C. G. Atkeson. Transfer of policies based on trajectory libraries. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2981–2986, 2007.

[69] Benjamin W. Tatler. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision*, 7(14):4–4, 11 2007.

[70] Andrea Thomaz, Guy Hoffman, and Maya Cakmak. Computational human-robot interaction. *Foundations and Trends® in Robotics*, 4(2-3):105–223, 2016.

[71] Adolfo Rodriguez Tsouroukdissian. Wiki.

[72] N. Vahrenkamp, H. Arnst, M. Wächter, D. Schiebener, P. Sotiropoulos, M. Kowalik, and T. Asfour. Workspace analysis for planning human-robot interaction tasks. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1298–1303, 2016.

[73] Benjamin Vincent, Roland Baddeley, Alessia Correani, Tom Troscianko, and Ute Leonards. Do we look at lights? using mixture modelling to distinguish between low- and high-level factors in natural image viewing. *Visual Cognition - VIS COGN*, 17:856–879, 08 2009.

[74] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[75] Weitian Wang, Yi Chen, Rui Li, and Yunyi Jia. Learning and comfort in human–robot interaction: A review. *Applied Sciences*, 9:5152, 11 2019.

[76] A. Zaraki, D. Mazzei, M. Giuliani, and D. De Rossi. Designing and evaluating a social gaze-control system for a humanoid robot. *IEEE Transactions on Human-Machine Systems*, 44(2):157–168, 2014.

[77] Xuan Zhao and Jia Pan. Considering human behavior in motion planning for smooth human-robot collaboration in close proximity. *CoRR*, abs/1807.07749, 2018.

Appendices

Appendix A

Human and Robot Configurations

This section provides the robot and human configurations used for the different problem scenarios in Chapter 4.

A.1 Configurations Used in the Problem Scenarios

Configuration	$q1$	$q2$	$q3$	$q4$	$q5$	$q6$	$q7$
q_{init}	0	-0.78	0.0	-2.36	0	1.57	0.78
$q_{init_{handover}}$	-0.81	-0.11	0.09	-2.32	0.013	2.21	0.06
q_{goal_1}	-0.12	-1.09	1.39	-2.11	1.07	1.67	1.67
q_{goal_2}	-0.94	-1.62 (+0.75)	2.14	-2.80 (-0.2)	0.55	3.41	0.26
q_{goal_3}	-0.32	0.44	-0.16	-1.00	0.07	1.43	0.33
q_{goal_4}	-1.98	0.95	1.50	-2.03	0.63	1.54	-2.66
q_{goal_5}	2.37	-1.76	-1.4	-1.32	-1.87	1.58	1.30
q_{goal_6} (handover)	0.52	-0.31	-0.41	-1.89	-0.12	1.59	-1.88

Table A.1: The start and goal configurations for the robot used to test the planners in problem scenarios 1, 2, and 3. For q_{goal_2}, the values in brackets indicate the changes made to the configuration for problem scenario 2 (to avoid a collision with an obstacle on the table). q_{goal_1} through q_{goal_5} are used for problem scenarios 1 and 2 while q_{goal_6} is used for problem scenario 3.

Joint	$Config.1$	$Config.2$	$Config.3$	$Handover$
spine_0	-0.38	0	0.22	0.02
spine_1	0.26	0.48	0.64	0.17
spine_2	0.36	-0.10	-0.09	0.0
neck_0	0	0	0.02	0.0
neck_1	0	0	-0.34	0.36
neck_2	0	0	-0.36	0.13
right_shoulder_0	1.08	0.89	0.12	0.82
right_shoulder_1	1.56	1.67	0.52	0.24
right_shoulder_2	0.16	0.19	1.08	-0.58
right_elbow_0	0.89	0.44	1.63	1.08
right_wrist_0	0	0.3	0.25	0.18
right_wrist_1	0	0	-0.31	-0.66
right_wrist_2	0	0	-0.25	-1.29
left_shoulder_0	0	1.09	0.16	0.0
left_shoulder_1	-1.52	-1.76	-1.82	-1.52
left_shoulder_2	0	0.02	0.20	-0.22
left_elbow_0	-0.71	-0.59	-0.53	-0.55
left_wrist_0	0	0	0	0.07
left_wrist_1	0	0	0	0.14
left_wrist_2	0	0	0	-1.51

Table A.2: The joint values for the three human configurations used in the Problem Scenarios. The set of hip and leg configurations are not listed as they were all set to their respective zero configurations for both problem scenarios.

Appendix B

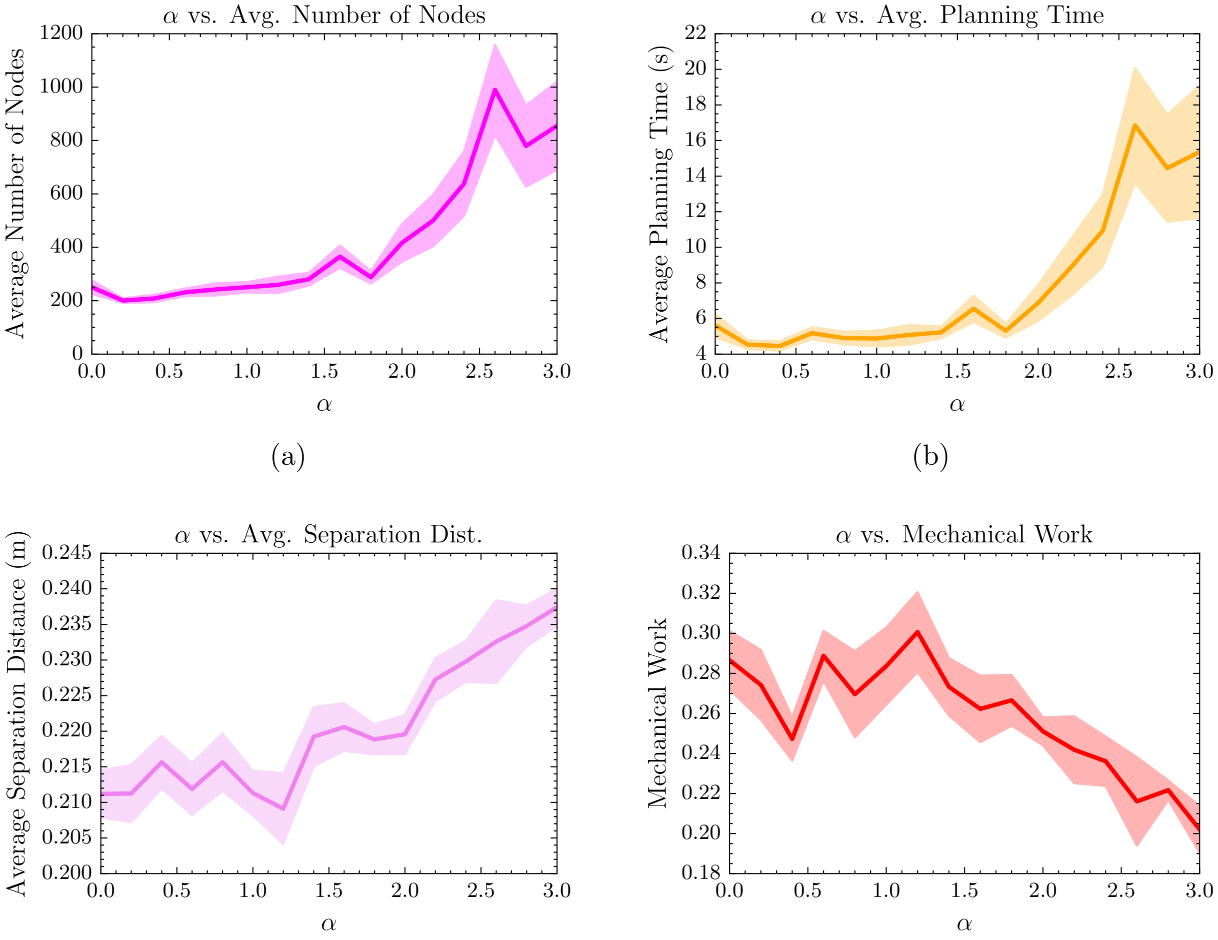
Human-Aware RRT-Connect Sensitivity Analysis

This section aims to provide further analysis with regards to a few Human-Aware RRT-Connect planner parameters.

B.1 Nearest Neighbor Configuration Cost Weight

On line 13 of Algorithm 2, the nearest neighbor search aims to minimize the distance and cost of the nearest neighbor node (i.e., $\underset{q_{old} \in \mathcal{T}}{\arg\min}\{\rho(q, q_{old}) + \alpha COST(q_{old})\}$). In this section, we aim to vary α, the weighting term for the cost of the potential nearest neighbor. To test the parameter, we use the simple environment depicted in Figure 4.1a, put the human in $Config.3$ listed in Table A.2 and let the robot plan from q_{init} to q_{goal_1} listed in Table A.1. The parameter α is varied from 0.0 to 3.0 in increments of 0.2 and 100 trials are generated for each value.

From Figure B.1, we see that larger α yields safer paths due to the decreasing nature of the mechanical work metric shown in Figure B.1d and the increase in average separation distance shown in Figure B.1c. However, for these gains in path quality, the number of nodes added to the search tree and the planning time increases significantly past α values of 1.8 as shown in Figure B.1a and Figure B.1b. This is because, for large α, the expression $\arg\min\{\rho(q, q_{old}) + \alpha COST(q_{old})\}$ is more biased to select the lowest cost node in the search tree with little priority given to the distance metric. Since there is no mechanism to prune the tree based on the number of nearest neighbours for a node, the tree starts expanding near local minima instead of a more uniform search of the configuration space. To compromise, α was set to 1.8 in this thesis to ensure that the nearest-neighbour search took into account configuration costs (to increase path quality) but didn't require long planning times.



(a)

(b)

(c)

(d)

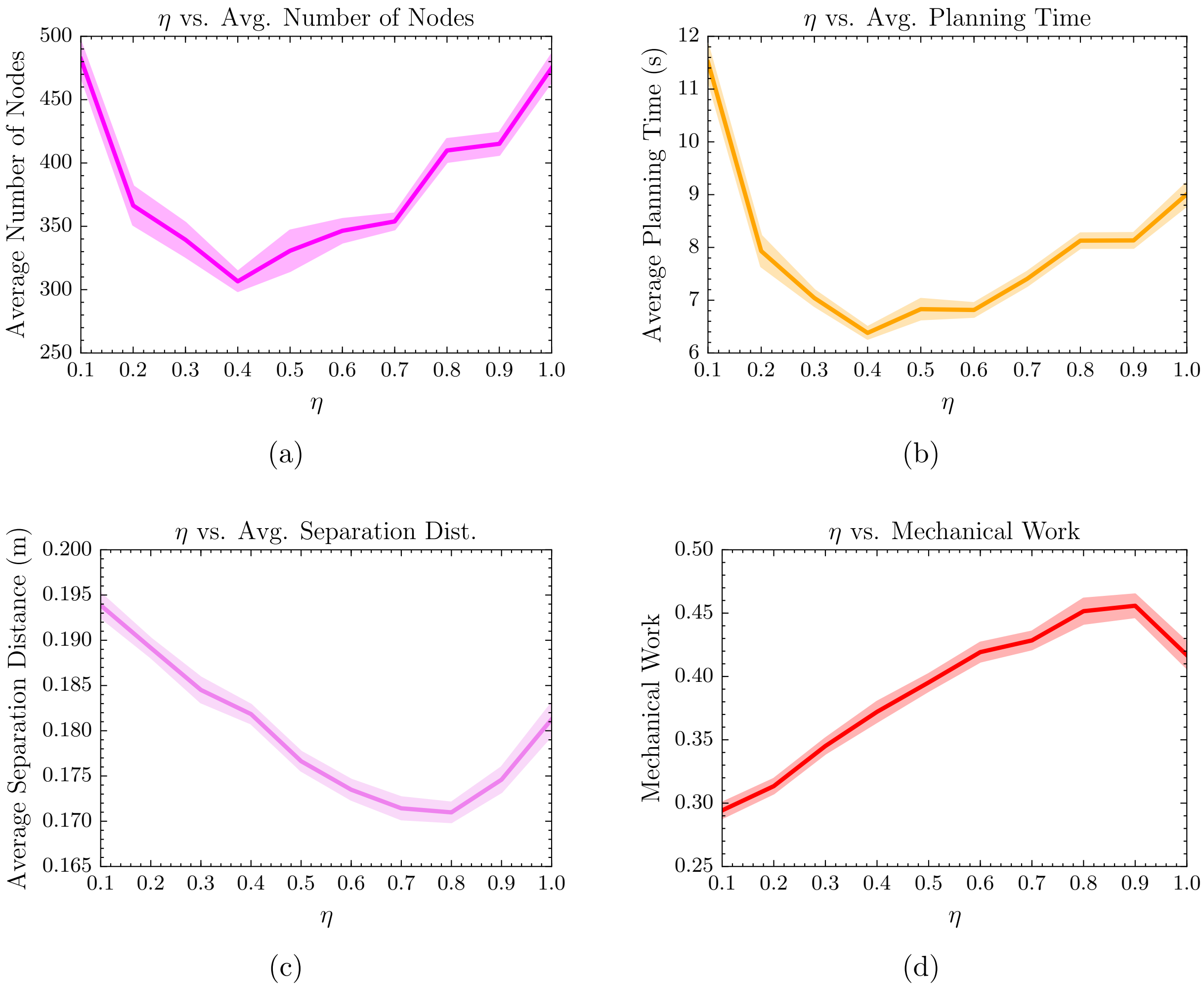
Figure B.1: Varying α against a) Average Number of Nodes, b) Average Planning Time, c) Average Separation Distance, and d) Mechanical Work for the Human-Aware RRT-Connect Planner.

B.2 Expansion Probability

On line 25 of Algorithm 2, a parameter η, the expansion probability, is introduced to allow the search tree to expand even if the new configuration, q_{new} is not of lesser cost than its nearest neighbor. This is done by accepting q_{new} with a probability of η, as long as its cost is at least less than the cost threshold c_{thres}. To test the parameter, we use the simple environment depicted in Figure 4.1a, put the human in $Config.3$ listed in Table A.2 and let the robot plan from q_{init} to q_{goal_1} listed in Table A.1. The parameter is varied from 0.1 to 1.0 in increments of 0.1 and 100 trials are generated for each value. As a note, an η value of 0.0 takes longer than 10000 iterations to solve the planning problem because the search trees are restricted to adding new configurations that are strictly of lesser cost than their nearest neighbors. For that reason, $\eta = 0.0$ is excluded from these results.

From the plots in Figure B.2d and Figure B.2c we can see that path quality is highest for lower values of η. This is because lower values reject configurations that are not of lesser cost than their nearest neighbors more often. As η is increased, path quality decreases but planning time improves. However, beyond an η value of 0.6, the planning time slightly increases. This may be because the planner is able to sample C_{free} with more lenient constraints (i.e., only collision checking and $COST(q_{new} < c_{thres})$ and thus adds more nodes into the search tree thereby increasing the nearest neighbor search time. This also motivates the need for lines 23-26 in Algorithm 2 and the η parameter as regularization terms that keep the number of expanded nodes within the search tree small, so that planning time is reduced.

From the presented analysis, an η value of 0.3 was settled on as a compromise between planning time, number of nodes generated and path quality.



(a)

(b)

(c)

(d)

Figure B.2: Varying η against a) Average Number of Nodes, b) Average Planning Time, c) Average Separation Distance, and d) Mechanical Work for the Human-Aware RRT-Connect Planner.

Appendix C

Connect T-RRT Algorithm

This section provides the details of the T-RRT variant used in Section 4.6, Connect T-RRT [16]. The algorithm and transition test are outlined below.

Algorithm 5 Connect T-RRT Planner

```
Input: Planning problem ($C_{free}, q_{init}, q_{goal}, c$)
Output: Feasible path or Failure
Parameters: $\epsilon$ is the step size for a motion
function CONNECT_T-RRT($q_{init}, q_{goal}$)
    $\mathcal{T}_{\mathcal{A}}$.Init($q_{init}$), $\mathcal{T}_{\mathcal{B}}$.Init($q_{goal}$)
    for $i = 1$ to planner_iterations do
        $q_{rand} \leftarrow$ Rand($\mathcal{C}$)
        if EXTEND($\mathcal{T}_{\mathcal{A}}, q_{rand}, c_{thres}$) == $Trapped$ then
            if CONNECT($\mathcal{T}_{\mathcal{B}}, q_{new}, c_{thres}$) == Reached then
                Return PATH($\mathcal{T}_{\mathcal{A}}, \mathcal{T}_{\mathcal{B}}$)
        SWAP($\mathcal{T}_{\mathcal{A}}, \mathcal{T}_{\mathcal{B}}$)
    Return FAILED
function EXTEND($\mathcal{T}, q, c_{thres}$)
    $q_{near} \leftarrow \arg\min_{q_{old} \in \mathcal{T}} \{\rho(q, q_{old})\}$
    if NEW_CONFIG($q, q_{near}, c_{thres}, q_{new}$) then
        if $\rho(q, q_{new}) \leq \epsilon$ then
            Return Reached;
        else
            Return Advanced;
    Return Trapped;
function NEW_CONFIG($q, q_{near}, c_{thres}, q_{new}$)
    $q_{new} \leftarrow$UNIT_V($q, q_{near}$) $* \epsilon$
    if OBSTACLE_FREE($q_{new}$) &
          TRANSITION_TEST($\mathcal{T}$, COST($q_{new}$), COST($q_{near}$)) then
        Return $True$;
    Return $False$
function CONNECT($\mathcal{T}, q, c_{thres}$)
    repeat
        $S \leftarrow$ EXTEND($q, \mathcal{T}, c_{thres}$)
    until not $S$ == Advanced
    Return $S$;
```

Algorithm 6 Transition Test

Input: Search tree $\mathcal{T}$, nearest neighbour configuration cost c_i, new configuration cost c_j

Output: $True$ if transition is accepted, $False$ if transition is rejected

Parameters: the cost threshold c_{max}, the current temperature T, the temperature increase rate T_{rate}

```
procedure TRANSITION_TEST($\mathcal{T}, c_i, c_j$)
    if $c_j > c_{max}$ then
        Return $False$
    else if $c_j < c_i$ then
        Return $True$
    else if $exp(-(c_j - c_i)/T) > 0.5$ then
        $T \leftarrow T/2^{(c_j - c_i)/(0.1 \cdot \mathcal{T}.costRange())}$
        Return $True$
    else
        $T \leftarrow T \cdot 2^{T_{rate}}$
        Return $False$
```