Motion Planning for Human-Robot Collaboration

Florian Beck, Andreas Kugi

Abstract

In this work, we address motion planning for robots in human-robot collaboration. An overview of important properties of a motion planning algorithm in terms of safety and human comfort is provided. In terms of comfort, we emphasize fluency, legibility, and human-like motion. Furthermore, existing planning algorithms are reviewed and contrasted in terms of these desired properties. Based on this review of the literature, a receding horizon trajectory optimization approach is proposed, and its main features are highlighted.

Keywords

Motion Planning, Receding Horizon, Human-Robot Collaboration, Safety, Comfort

1 Introduction

In recent years, there has been an increase in demand for robots capable of working in the proximity of humans or even collaborate with them. Possible applications range from collaborative tasks in industry, such as load sharing tasks or joint assembly, to service robots in domestic environments. Because traditional safety measures such as fences are no longer appropriate for these applications, novel concepts are required to enable safe collaboration. Aside from safety, collaborative tasks give rise to additional requirements in task orchestration and adaptable robot behavior based on observations of the environment. Furthermore, human comfort during the interaction is critical in establishing the robot as a trustworthy collaborator.

These requirements are typically handled by different layers in the automation architecture. First, a cognitive decision layer coordinates tasks between the human and the robot. This layer gives explicit goals to a motion planning layer, which are then executed by an underlying controller layer.

In this work, we focus on the motion planning layer while explicitly considering the interface to a suitable controller for task execution. In the first step, an overview of the requirements with respect to safety and human comfort in human-robot collaboration (HRC) will be provided. Second, existing planning algorithms proposed in the literature will be shortly reviewed given these requirements.

Based on this analysis, open issues towards a flexible motion planning approach for HRC are identified. A receding horizon trajectory optimization planner is proposed as a contribution to resolving these issues. For this, we take advantage of the possibility to formulate the requirements for safety and comfort during the interaction as objective functions and constraints for trajectory optimization.

https://doi.org/10.34727/2022/isbn.978-3-85448-052-5_6
This chapter is licensed under a Creative Commons Attribution-ShareAlike 4.0 International licence.

2 Collaborative Robots

Collaborative robotics applications require not only algorithmic solutions for the control, planning, and cognitive layers, but also suitable mechanical structures. In recent years, several collaborative robots, also referred to as cobots, have been developed and brought to market. Examples include robots by Universal Robots, the KUKA LBR iiwa, and Franka Emika's Panda. The latter two are based on technology developed at DLR [Hirzinger et al. 2002] focusing on lightweight, torque-controlled robots with elastic joints. The main advantage of the lightweight design, while maintaining a reasonable payload, is that it reduces the inertia of the robot links which directly contributes to reducing injuries upon impact. Another benefit of such lightweight collaborative robots is their ability to be mounted on mobile platforms, allowing for mobile manipulation. Furthermore, these robots feature seven degrees of freedom (DOF), which increases the manipulability through kinematic redundancy. The 7 DOF robot arms also mimic the human hand to some extent, allowing analogies in planning and control to be drawn between the human and the robot.

In our work, we use the KUKA LBR iiwa 14 R820 as a collaborative manipulator arm, which can also be mounted on DS Automotion's Sally, a differential drive mobile platform, shown in Fig. 1 as a reference platform.

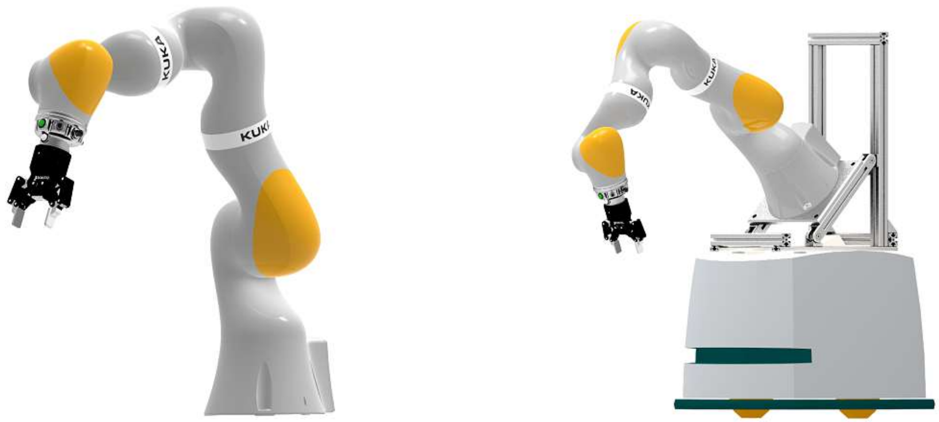

Figure 1 The KUKA LBR iiwa 14 R820 collaborative robot with a gripper (left) and KUKA LBR iiwa 14 R820 mounted on DS Automotion's SALLY, a differential drive mobile platform (right).

3 Motion Planning Requirements for HRC

Motion planning algorithms serve as a central component in the robot's automation architecture. In this section, the desired properties of motion planning algorithms with respect to human-robot collaboration, in particular safety and comfort, as well as existing approaches from the literature will be reviewed.

3.1 Safety

The most important criterion for enabling close collaboration between humans and robots is safety. In the context of industrial robots, safety is typically ensured by fencing or structural measures surrounding the robot, such that the robot moves only if no humans are in close proximity. This is of course incompatible with collaborative tasks. As a result, safety concepts are required to avoid collisions or to mitigate the consequences of impact in case of collisions. An overview of design criteria for safe human-robot interaction, both on the mechanical construction level and for the algorithm design, is given in [Alami et al. 2006]. In the following, we focus on algorithm design, assuming appropriate mechanical properties as described in Section 2.

In [Haddadin et al. 2017], the authors distinguish between two phases, namely pre-collision and post-collision. Motion planning is mainly concerned with the pre-collision phase. This means that collision-free trajectories must be planned while still meeting of task completion requirements. Typical collision avoidance approaches require a geometric representation of the robot's environment. Due to the computational complexity, objects are often approximated by convex shapes. Although this simplifies pairwise collision checks, the environment can still be non-convex if it contains multiple convex shapes. Using these convex representations, algorithms like the Gilbert-Johnson-Keerthi (GJK) algorithm [Gilbert et al. 1988; Cameron 1997] can be employed to check whether the robot is in collision with the environment in a given configuration. Another popular collision checking approach is the V-Clip Algorithm [Mirtich 1998]. The paper's performance comparison does not indicate a clear improvement, but rather depends on the specific application.

Collision checking is computationally expensive in motion planning in general because pairwise checks between obstacles and the robot, or parts of the robot, must be performed. Furthermore, depending on the planning algorithm this may have to be repeated several times. For safety, it is also important to consider that such collision checking approaches are only executed at discrete points in time. Collisions between two sampling points are theoretically possible depending on the time discretization intervals. There are two solutions to this problem. First, sampling density can be increased. This, however, comes at significant computational costs due to the increased number of samples required for representing a movement. Second, collision detection can be extended to a continuous-time approach. Examples for continuous collision checking can be found in [Schulman et al. 2014] and [Merkt et al. 2019]. Such collision detection approaches are also applicable in dynamic environments. However, the environment geometry must depend on time. Hence, a motion model of objects in the environment is required to predict their movement.

For motion planning approaches, the post-collision phase must be considered in addition to the pre-collision phase. The post-collision is typically treated in the underlying control layer. Detection of collision and appropriate reaction strategies are proposed in [De Luca et al. 2006] and [Haddadin et al. 2008] through torque measurements in the joints. These strategies are typically combined with impedance control [Ott 2008], enabling a compliant robot behavior. Such compliant behavior is often desired in the Cartesian space of the robot end-effector. To use such control laws, the motion planning algorithm must provide sufficiently smooth trajectories, i.e. at least two times continuously differentiable. Furthermore, due to the presence of the inverse Jacobian in the control law, Cartesian impedance control requires singularity-free trajectories. In this regard, it is critical not only to avoid singular configurations during planning, but also to include a sufficiently large safety margin around the singularities. This is because, in the proximity of singularities, small velocities in the task space can still result in large velocities in the joint space.

3.2 Natural Motion and Comfort

In addition to functional aspects of a planner, such as reaching a goal, feasibility of the trajectory, and the adherence to safety aspects according to Section 3.1, human comfort must be taken into account when planning a robot's motion. In general, it is difficult to rigorously define robot motion that is comfortable for humans. It is highly dependent on how a human perceives the situation and can vary greatly depending on the individual. Furthermore, it may be dependent on the robot's capabilities and design. Studies in human-robot interaction (HRI) try to identify such properties. Furthermore, it is desirable to formalize such properties to an extent such that they can be considered during planning. This was accomplished for certain criteria, which will be discussed in the following. An overview of social aspects and psychological factors for safety in HRI can be found in [Lasota et al. 2017].

One of the most discussed aspects regarding comfort is proxemics [Hall 1963], i.e. the notion of distance between humans or a human and a robot, respectively, during certain interactions. The influence of a separation distance between a robot and a human was for example investigated in [Arai et al. 2010; Koay et al. 2006; Kulić and Croft 2007]. In a collaborative setting, distance is frequently constrained by the task at hand. Aside from the desired end-effector goal, there are often additional DOF that can be used to determine the pose or movement of the robot in space, depending on the specific task. For a mobile manipulator, this includes the positioning of the vehicle itself concerning the end-effector goal and the human.

An important concept with respect to comfort is legibility, as for example discussed in [Lichtenthäler et al. 2012] to increase the perceived safety. Legibility is a measure of how well the robot can convey its intent. In the motion planning context, this means that movement has to be planned such that ambiguity is reduced making goals easily inferable by a human. In some cases, this can be achieved by certain exaggeration of the movement, for example moving in a circular arc toward an object. Of course, this type of exaggeration is not always achievable, especially if several target objects are located close to each other. In such a scenario, it depends on other factors, e.g. if the human can infer where the robot is moving next. This cannot be solved using motion planning alone. An optimization-based formulation of legibility can be found in [Dragan et al. 2013], which also gives a comparison to the notion of predictability. Predictable motion is defined as predicting how a motion will look like if the goal has already been determined. As a result, the inference direction is reversed. In this regard, predictable motion can differ from legible motion. Predictability or legibility is preferred depending on the collaborative task at hand. For example, if the task consists of a fixed, sequential process, a human already knows what the robot's goals are, and predictability is more important than legibility. Legible motion, on the other hand, is preferred when the task is ambiguous.

In [Hoffman 2019], an overview of methods to evaluate fluency in HRC is given. They provide a definition and a model for assessing fluency. Fluent collaboration occurs when a human and a robot achieve a high level of coordination, resulting in precisely timed, efficient sequences of action. In a user study, they discovered that human idle time, i.e. the human waiting for the robot, as well as the functional delay of the robot, has a significant influence on subjective fluency. Longer human idle time is perceived as increasing fluency, which was indicated by feedback from participants who thought the robot did a better job. Increasing the functional delay, on the other hand, has a negative impact on the sense of fluency. This can be directly related to the robot's time to action following the completion of the human's turn during the collaboration. The requirement of short functional delays implies that fast planning and replanning are essential properties of motion planning algorithms. An example of fluency for robot-human handovers is given in [Cakmak et al. 2011] considering the functional delay. They propose that conveying intent is a major factor in fluency. If the robot does not make its intentions to hand over an object clear, functional delays increase and the sense of fluency decreases during the interaction. This demonstrates that not only fast planning is required, but approach directions and timing must also be considered for comfortable interactions. Further examples of the importance of approach directions during handovers are given in [Koay et al. 2007] and [Sisbot and Alami 2012]. Human motion and action prediction are extremely useful for reducing such functional delays and increasing fluency. There is a substantial body of literature on human motion prediction in terms of long-term prediction, i.e. full reaching motions, see, e.g., [Luo et al. 2018], as well as short-term predictions obtained by tracking algorithms. Both are important for motion planning. Short-term predictions primarily improve the observations, resulting in more accurate estimates of the goals and dynamic obstacles in the environment. Long-term prediction, on the other hand, can be used to estimate human intentions and thus, influence fluency directly. Prediction combined with rapid replanning results in both reactive and anticipatory action [Hoffman and Breazeal 2007].

Depending on the mechanical structure of the robot, also human-like motions can be planned. Anthropomorphic robot arms, for example, such as the KUKA LBR iiwa, mimic the structure of a human arm with seven DOF. Optimal control theory was used to analyze human reaching motions in relation to the hand pose, see, e.g., [Flash and Hogan 1985] and [Todorov and Jordan 2002]. The results show that hand movement minimizes jerk, leading to smooth motions with bell-shaped velocity profiles. These findings provide explicit criteria that, in principle, can be applied to robotic motion planning. Maximizing the smoothness of the trajectories is somehow contradictory to minimizing the time, i.e. time optimality, which is commonly desired in industrial processes to maximize throughput. Fast robot movements, however, are perceived as less safe when interacting with humans [Arai et al. 2010]. This implies that the smoothness of robot motion is extremely important in HRC. Another important aspect is motion planning in the task space, i.e. Cartesian end-effector coordinates because most existing motion planning algorithms are designed in the joint space. In the case of a redundant robot, the nullspace motion must also be considered. The nullspace motion typically determines the robot's elbow movement, which is strongly dependent on the robot structure and can only be determined on a very limited basis by HRI.

4 Motion Planning Algorithms

In this section, we give an overview of existing motion planning algorithms in the literature while also assessing their capabilities with respect to the criteria identified in Section 3. Because of its importance in robot autonomy, motion planning has received a lot of attention in robotics research. The corresponding algorithms can be categorized into planning for static and dynamic environments. While we are primarily interested in real-time planning in dynamic environments, algorithms proposed for static environments are frequently used as the foundation for developing real-time capable methods for dynamic environments.

In static environments, sampling-based methods received a lot of attention. Their primary benefit is that obstacles do not need to be explicitly modeled in the configuration space. A collision detection module is instead used to determine whether or not a sample in configuration space is in collision. This greatly improves the planning efficiency [LaValle 2006]. Two important representatives of sampling-based algorithms are Probabilistic Roadmap (PRM) [Kavraki et al. 1996] and Rapidly-exploring Random Trees (RRT) [LaValle and J. 2001]. While PRM invests heavily in preprocessing to provide fast multi-query planning, RRTs are designed to be fast single-query planners. The basic RRT algorithm has probabilistic completeness, i.e. in the limit a path, if it exists, will be obtained with probability one. For a simplified version of PRM, this was proven as well [Kavraki et al. 1998]. Since their initial publication, several extensions were proposed to PRM and RRT motion planning. For our purpose, extensions toward optimal motion planning are the most relevant. Thus, for instance, the asymptotically optimal algorithms RRT* and PRM* were proposed in [Karaman and Frazzoli 2011]. Although sampling-based motion planners have several desirable properties, particularly probabilistic completeness, they frequently suffer from non-smooth trajectories, which require further post-processing. This ultimately increases the planning time. Furthermore, complex objectives and constraints lead to a high computational load. This can be a problem when formulating the objectives for comfort, as discussed in Section 3.2.

As a possible solution to these issues, trajectory optimization was proposed. Although, in general, trajectory optimization returns only locally optimal trajectories, it has been successfully applied to robotic motion planning. Trajectory optimization can be used to refine trajectories obtained from sampling-based planners, but it can also be used as a stand-alone algorithm. In [Ratliff et al. 2009; Zucker et al. 2013], an optimization-based planner called Covariant Hamiltonian Optimization for Motion Planning (CHOMP) was proposed. The objective function consists of two cost terms, an obstacle cost based on Euclidean distance fields and a smoothness cost that takes velocities and accelerations into account along the trajectory.

The trajectory is updated iteratively using covariant gradient descent. The update rule ensures that the trajectory remains smooth while decreasing the cost. The experiments demonstrate the algorithm's successful application to robotic manipulation. One significant drawback, which the authors also mention, is that due to the fixed discretization, only trajectories of a predefined length are considered.

In [Kalakrishnan et al. 2011], a stochastic optimization approach for motion planning called Stochastic Trajectory Optimization for Motion Planning (STOMP) is presented. The authors propose using a series of noisy trajectories that deviate slightly from the current candidate trajectory, and are then simulated to determine their costs. The candidate solution is updated based on these costs. One of the main advantages of this approach is that, because of derivative-free stochastic optimization, it can deal with general constraints for which gradients are not always available. This can be an advantage compared to CHOMP [Ratliff et al. 2009; Zucker et al. 2013] if desirable cost functions are not differentiable.

The method in [Schulman et al. 2014] is similar to CHOMP [Ratliff et al. 2009; Zucker et al. 2013], however, the authors make use of sequential convex optimization. In each iteration, a convex approximation of the nonlinear trajectory optimization problem is constructed. A trust region method is used to ensure that the approximation remains valid. In addition, infeasible constraints are converted to ℓ_1 penalties. A quadratic programming solver is used to solve the convex subproblem. For collision checking, GJK as mentioned in Section 3.1 is used. To ensure continuous-time safety, the collision checking procedure takes into account a swept-out volume, which is a polyhedral approximation of the free configuration space between two time steps. When compared to CHOMP [Ratliff et al. 2009; Zucker et al. 2013] and sampling-based planners implemented in the open motion planning library (OMPL) [Şucan et al. 2012] including RRT [LaValle and J. 2001], the experiments show a significant improvement in terms of speed, the problems that can be solved, and the quality of the resulting trajectories. Furthermore, this framework allows for inclusion of more complex cost functions, such as those related to human comfort.

Recently, a framework for guaranteed sequential trajectory optimization (GuSTO) [Bonalli et al. 2019] using sequential convex programming (SCP) was proposed. In contrast to TrajOpt [Schulman et al. 2014], which makes use of SCP as well, theoretical guarantees for convergence to at least a stationary point are given by the authors. Numerical simulations demonstrate that this approach provides more accurate results in less time compared to other state-of-the-art SCP-based planners.

To capture dynamic environments and real-time planning, several approaches can be found in the literature. Extensions to RRT planning include [Li and Shie

2002; Ferguson et al. 2006] and [Zucker et al. 2007]. In addition, [Svenstrup et al. 2010] use the RRT algorithm in combination with a dynamic potential field. The potential field takes into account the robot's position in the environment, its goal, and the humans moving in its vicinity. To account for changes in the environment the planner is implemented as a model predictive controller (MPC). To that end, only the first few steps of the planned trajectory are executed, while the planner calculates a new trajectory on-line. In [Sun et al. 2015], a similar RRT-based approach for high-frequency replanning was developed. A stochastic motion model of the robot is used. Several independent RRTs are executed in parallel to quickly find an optimal plan. The lowest cost plan is then chosen. While a single RRT will not find an optimal solution, it is proven that running several RRTs in parallel will asymptotically converge to an optimal plan. However, sampling-based planners for dynamic environments have the same drawbacks as their static counterparts.

In [Park et al. 2012], a similar concept using trajectory optimization is proposed. The motion of dynamic obstacles is taken into account by predicting their motion over a short-time horizon and computing a conservative local bound on their location and velocity. Based on this information, a constraint optimization problem is solved to compute a plan. Because dynamic object trajectories are only predicted for a short period of time, prediction uncertainty grows quickly. The planner is executed again in each time step, and only one step of the trajectory is executed before replanning.

The works [Ghazaei Ardakani et al. 2015, 2019] present an MPC approach for real-time point-to-point trajectory generation for a robot manipulator. A linear kinematic robot model is used, given by a double integrator system, where joint positions, velocities and accelerations serving as optimization variables. The final trajectories are generated using linear interpolation with a fixed sampling time. Because of the fixed sampling intervals and the goal constraint on the final step, it is assumed that the trajectory duration is sufficient to reach the goal while taking the robot's kinematic limits into account. The authors successively reduce the sampling period in the experiments, increasing the time resolution of the trajectory as the robot approaches the goal. The fixed sampling period, on the other hand, implies that the robot trajectory is initially quite coarse, which can be problematic in terms of constraint satisfaction, such as collision constraints for safety. Due to the convex formulation of the optimization problem, the authors report fast convergence of their algorithm. The convex formulation, on the other hand, significantly limits the available optimization criteria.

In contrast, in [Krämer et al. 2020] a different approach utilizing a cost-to-go-term was proposed replacing the requirement of a goal constraint. This allows for a fewer discretization points along the trajectory without sacrificing sampling density. This is especially important in terms of safety because high sampling density reduces the likelihood of collision between trajectory samples while being computationally less expensive than continuous collision checking as, for example, done in [Schulman et al. 2014]. The results of [Krämer et al. 2020] show that achieving planning times below 100 ms per MPC iteration for pick-and-place tasks is feasible.

In [Agboh and Dogar 2018], an extension of STOMP [Kalakrishnan et al. 2011] to real-time replanning for grasping in cluttered environments is proposed. Initially, an open-loop trajectory is generated with numerous iterations to obtain a locally optimal solution. Starting with this initial trajectory, replanning is done with fewer iterations and with feedback from the current state. High-quality trajectories can be generated while maintaining fast planning times if the inital trajectory is a good initialization for replanning. The experiments show that the approach works well for grasping in cluttered environments that do not change too quickly. For moving targets or obstacles, the initialization is not a good approach since the trajectory can already be infeasible when the planner has finished.

A local receding horizon trajectory optimization given a global reference path in a difficult terrain is proposed in [Howard et al. 2010]. In [Toit and Burdick 2012], robot motion planning is formulated as a stochastic dynamic programming (SDP) problem. The authors explicitly address uncertainty rooted in the robot's environment. Because of the stochastic context, constraints are formulated as chance constraints [Toit and Burdick 2011], meaning that the constraint has to be fulfilled with a certain confidence. Given the complexity of the SDP problem, it is approximately solved using stochastic receding horizon control in the belief space. In dynamic uncertain environments, the stochastic approach provides more accurate models for planning. However, when compared to deterministic solutions, the additional computational effort is significant.

Recently, an MPC concept for autonomous guided vehicle motion planning was published [Mercy et al. 2018]. The authors use B-Spline trajectory parametrization to guarantee constraint satisfaction in the resulting nonlinear trajectory optimization problem. In contrast to [Toit and Burdick 2012], obstacles are modeled and predicted in a deterministic way facilitating a linear prediction model. The experiments show that dynamic obstacles in the environment can be safely avoided when combined with fast replanning.

MPC can also be used to plan and track a robot's trajectory at the same time. This has the advantage of not requiring the use of a trajectory following controller. Furthermore, the dynamic constraints of the entire system can be systematically considered allowing for more aggressive trajectories. The MPC framework CIAO [Schoels et al. 2020] is based on a novel convex inner approximation of the collision avoidance constraint. This enables the planning of kinodynamically feasible collision-free trajectories in continuous time. A real-world experiment with a differential drive mobile robot demonstrates the unified trajectory optimization and tracking. Planning for multi-body robots, on the other hand, has yet to be demonstrated.

Simultaneous trajectory optimization and tracking was also applied to full dynamic models of robot manipulators, see, e.g., [Tassa et al. 2012] and drones, see, e.g, [Neunert et al. 2016]. Recently, Kleff et al. [Kleff et al. 2021] proposed an MPC approach based on differential dynamic programming (DDP) in real time on a collaborative robot. However, so far, MPC with full dynamics has only been solved for simplified problems, with additional objectives such as obstacle avoidance being neglected. As a result, for the currently available real-time hardware, approaches with separate trajectory planning and trajectory tracking control are typically used.

5 Receding Horizon Trajectory Optimization

In this section, we provide a brief overview of a receding horizon trajectory optimization approach for robot motion planning that takes into account the requirements from Section 3. In comparison to previous works discussed in Section 4, we explicitly take into account the combined requirements from Section 3, namely pre-collision and post-collision safety, legibility and smooth robot motion while enabling fluent interaction. We maintain compatibility with Cartesian impedance control by introducing computationally efficient singularity avoidance based on penalty functions. In addition, a novel via-point approach for receding horizon trajectory optimization is discussed providing a framework for planning legible and human-like motion with low computational overhead. Due to our emphasis on computational efficiency in the aforementioned features, fluent interactions can be ensured.

The planning approach considers robot manipulators under kinematic constraints. Note that robot dynamics are not considered in the planner. It is assumed that the underlying controller, i.e. a Cartesian compliance control scheme [Ott 2008], compensates for the nonlinear dynamics resulting in a remaining linear double integrator system. The motion planning problem is formulated as a trajectory optimization in the form

$$\min_{\mathbf{u}_{0|n},\dots,\mathbf{u}_{N-1|n}} \sum_{k=0}^{N-1} l(\mathbf{x}_{k|n}, \mathbf{u}_{k|n}) \tag{1a}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1|n} = \mathbf{\Phi}\mathbf{x}_{k|n} + \mathbf{\Gamma}\mathbf{u}_{k|n} \tag{1b}$$

$$\mathbf{x}_{0|n} = \mathbf{x}_{1|n-1}, \ \mathbf{u}_{0|n} = \mathbf{u}_{1|n-1} \tag{1c}$$

$$\underline{\mathbf{x}} \leq \mathbf{x}_{k|n} \leq \overline{\mathbf{x}}, \quad k = 0, \dots, N-1 \tag{1d}$$

$$\underline{\mathbf{u}} \leq \mathbf{u}_{k|n} \leq \overline{\mathbf{u}}, \quad k = 0, \dots, N-1 \tag{1e}$$

for the time steps $k = 0, \dots, N-1$, with fixed sampling time T_s. The optimization problem (1) is solved at every sampling instant n for the finite planning horizon NT_s. Only the first step of the optimal control input is applied to the system until the next sampling instant $n+1$. The optimization problem is then solved again, now starting one sampling time T_s ahead and therefore predicting one step further into the future. Hence, the planning horizon is said to be receding. Eq. (1a) describes a general objective function to be minimized for the planning horizon nT_s to $(n+N-1)T_s$ depending on the robot's state $\mathbf{x}_{k|n}$ and the input $\mathbf{u}_{k|n}$ at the time $(n+k)T_s$, $k = 0, \dots, N-1$. The objective function includes a cost term that represents the distance to the goal such that the robot moves toward this goal. Additional cost terms can be added depending on the specific application, which will be discussed in greater detail in the remainder of this section. The resulting linear system of the robot dynamics is an equality constraint defined by Eq. (1b). The planner is initialized from the previously calculated trajectory through Eq. (1c). State and input constraints, specifically addressing joint limits, velocity limits, and higher derivatives, if necessary, are considered in Eq. (1e) where $\underline{\mathbf{x}}$, $\underline{\mathbf{u}}$, and $\overline{\mathbf{x}}$, $\overline{\mathbf{u}}$ denote lower and upper bounds, respectively.

The receding horizon trajectory optimization shares advantages of the trajectory optimization approach over sampling based algorithms as stated in Section 4. This is particularly relevant to the flexibility of objective functions and constraints in modelling desired properties in human-robot interactions and ensuring smooth trajectories. Furthermore, we use a cost-to-go term for reaching the goal in combination with fixed sampling times similar to what is done in [Krämer et al. 2020]. This allows for fast planning while still maintaining tightly sampled trajectories. Fast planning times are essential to reduce the robot's functional delay, enabling fluent interactions. Nonetheless, safety cannot be sacrificed for the sake of fast planning times. As mentioned in Section 3.1, safety violations can in principle happen between the discrete time steps of the trajectory optimization. Due to the computational effort, we do not consider continuous-time collision checking but instead, rely on small sampling times T_s. This property is in contrast to previous approaches in the literature. For example, [Ghazaei Ardakani et al. 2015] and [Mercy et al. 2018] demand that the final point in the planning horizon already reaches the goal. This requires either a fixed duration of the trajectory, i.e. independent of the distance to the goal or the introduction of the duration as an additional optimization variable, increasing the complexity and thus the computational effort of the problem.

Collision checking for receding horizon trajectory optimization can be performed using well-known approaches from the literature. However, the gradient of the objective function and constraints can be provided to improve the optimization algorithm's convergence behavior. To this end, we use a smooth distance approximation as introduced in [Vu et al. 2020]. We extend the formulation from rectangular boxes and points to spheres as basic obstacle shapes to allow for more complex environments. Because collision checking is frequently the bottleneck, limiting to simple shapes results in faster optimization times. In the context of collision checking, the receding horizon framework also enables planning in dynamic environments. New information about objects in the environment can be incorporated due to the constant replanning. Furthermore, by including object states in the dynamic constraints (1b), predictions of object movements in the planning horizon can be taken into account.

In addition to safety considerations in the pre-collision phase, we also address compatibility with Cartesian impedance control [Ott 2008] to enable post-collision safety. The Cartesian impedance controller requires trajectories to be at least two times continuously differentiable and singularity free. In the proposed approach, sufficient smoothness is guaranteed by the equality constraints (1b) and (1c). Note that, in general, the sampling times of the trajectories are significantly lower than those required for the execution of the controller. As a result, for the controller, the trajectories must be interpolated and resampled. The optimization framework provides several ways to make sure that the planned trajectories are free of singularities. As mentioned in Section 3.1, a safety margin around singularities has to be taken into account. A distance to a singular configuration can in general be defined by the so-called manipulability measure [Yoshikawa 1985]. Alternatively, if a robot's singular configurations are known, direct distance measures in the configuration space can be used. The safety margin can be formulated as an inequality constraint ensuring a minimum distance to singular configurations or by demanding a minimum amount of manipulability. In view of the computational costs, the singularity avoidance is realized by a penalty function which is added to the objective function $l(\mathbf{x}_{k|n}, \mathbf{u}_{k|n})$. Note that, in principle, this does not guarantee singularity-free motions due to competing cost terms, however, such a situation can be avoided by selecting a sufficiently large weight for the penalty function.

Besides safety, we explicitly address comfort discussed in Section 3.2 within the receding horizon trajectory optimization framework. First, we consider human-like movement, as for the example investigated in [Flash and Hogan 1985]. Human movement of the hand is regarded as minimizing jerk there. This corresponds to minimizing jerk along an end-effector trajectory in the task space in the robotic applications under consideration. This formulation can be easily incorporated into the trajectory optimization problem, however, it would require planning in the task space. Direct planning in the task space makes the consideration of the joint limit constraints more involved. Because of the nonlinear relationship between the joint and the task space, enforcing smoothness in the joint space is computationally more efficient but does not always result in human-like movement in the task space. Planning in the joint space, but formulating the cost-to-go term in the task space based on the forward kinematics is another option, again at the cost of higher computational effort. Thus, we propose to approximate a cost-to-go term in the task space by placing via-points in the task space along the planned trajectories. The approximation is more or less coarse and computationally expensive depending on the number of via-points. Such intermediate goals are also important for a variety of other robotic tasks. Grasping for example requires the gripper to be aligned with an object in a so-called pre-grasp pose before the grasp point is reached. Furthermore, via-points can also aid the establishment of comfortable interactions by ensuring appropriate approach directions and end-effector orientations. In addition, intermediate goals can help to disambiguate goals resulting in legible robot trajectories. Again, this is a computationally efficient approximation compared to what is done, e.g. in [Dragan et al. 2013] to achieve legible robot motion. In previous works, see, e.g., [Schulman et al. 2014; Ghazaei Ardakani et al. 2015], a common approach for intermediate goals was to constrain points along the trajectory to via-points. This requires predetermined timings for the via-points along the trajectory. Furthermore, for a receding planning horizon, this approach is not feasible because the via-point may not be reachable within the horizon.

In contrast, we propose to formulate the optimization problem in such a way that only the relative timing between via-points, i.e. a sequence of via-points, is considered, rather than the exact timing along the trajectory. To that end, we introduce a parametrized reference path that linearly interpolates from the starting configuration through the via-points to the goal. The path parameter's dynamics are added to the optimization problem to represent the progress along the path. In contrast to classical path-following control, see, e.g., [Böck and Kugi 2014, 2016; Faulwasser et al. 2017], we are not interested in precisely following the path, but only in accurately passing through the via-points. Therefore, the cost weights are adjusted so that progress along the path is favored over precise tracking between via-points. To pass the via-points exactly, a path progress dependent constraint is introduced. Instead of being specified in advance, the optimizer determines the timings and velocities through the via-points in this formulation.

6 Conclusions

In this work, we outlined the requirements for motion planning algorithms in collaborative human-robot tasks. In addition to physical properties of collaborative robots, a brief overview of algorithmic safety measures for planning algorithms, particularly collision checking, was provided. Although safety is the topmost priority, it is not the only requirement for planning in human-robot collaborative tasks. In this context, properties related to comfort including proximity, legibility, fluency, and human-likeness of robot motion were discussed. Furthermore, the state-of-the-art motion planning algorithms were evaluated concerning these requirements. Finally, a motion planning framework based on a receding horizon optimization approach was outlined. This method enables the flexible specification of control objectives and the systematic incorporation of constraints to easily adjust the desired properties for HRC.

Bibliography

Wisdom C. Agboh and Mehmet R. Dogar. 2018. Real-Time Online Re-Planning for Grasping Under Clutter and Uncertainty. *IEEE-RAS International Conference on Humanoid Robots*, 1–8. https://doi.org/10.1109/HUMANOIDS.2018.8625041

R. Alami, A. Albu-Schaeffer, A. Bicchi, R. Bischoff, R. Chatila, A. De Luca, A. De Santis, G. Giralt, J. Guiochet, G. Hirzinger, F. Ingrand, V. Lippiello, R. Mattone, D. Powell, S. Sen, B. Siciliano, G. Tonietti, and L. Villani. 2006. Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1–16. https://doi.org/10.1109/IROS.2006.6936985

T. Arai, R. Kato, and M. Fujita. 2010. Assessment of operator stress induced by robot collaboration in assembly. *CIRP Annals* 59, 1 (2010), 5–8. https://doi.org/10.1016/j.cirp.2010.03.043

Martin Böck and Andreas Kugi. 2014. Real-time Nonlinear Model Predictive Path-Following Control of a Laboratory Tower Crane. *IEEE Transactions on Control Systems Technology* 22, 4 (2014), 1461–1473. https://doi.org/10.1109/TCST.2013.2280464

Martin Böck and Andreas Kugi. 2016. Constrained model predictive manifold stabilization based on transverse normal forms. *Automatica* 74 (2016), 315–326. https://doi.org/10.1016/j.automatica.2016.07.046

Riccardo Bonalli, Abhishek Cauligi, Andrew Bylard, and Marco Pavone. 2019. GuSTO: Guaranteed Sequential Trajectory optimization via Sequential Convex Programming.

International Conference on Robotics and Automation, 6741–6747. https://doi.org/10.1109/ICRA.2019.8794205

Maya Cakmak, Siddhartha S. Srinivasa, Min Kyung Lee, Sara Kiesler, and Jodi Forlizzi. 2011. Using spatial and temporal contrast for fluent robot-human hand-overs. *ACM/IEEE International Conference on Human-Robot Interaction,* 489–496. https://doi.org/10.1145/1957656.1957823

Stephen Cameron. 1997. Enhancing GJK: computing minimum and penetration distances between convex polyhedra. *International Conference on Robotics and Automation*, 3112–3117. https://doi. org/10.1109/ROBOT.1997.606761

Alessandro De Luca, Alin Albu-Schaffer, Sami Haddadin, and Gerd Hirzinger. 2006. Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1623–1630. https://doi.org/10.1109/IROS.2006. 282053

Anca D Dragan, Kenton C T Lee, and Siddhartha S Srinivasa. 2013. Legibility and predictability of robot motion. *ACM/IEEE International Conference on Human-Robot Interaction,* 301–308. https: //doi.org/10.1109/HRI.2013.6483603

Timm Faulwasser, Tobias Weber, Pablo Zometa, and Rolf Findeisen. 2017. Implementation of Nonlinear Model Predictive Path-Following Control for an Industrial Robot. *IEEE Transactions on Control Systems Technology* 25,4(2017),1505–1511. https://doi.org/10.1109/TCST.2016.2601624

Dave Ferguson, Nidhi Kalra, and Stentz Anthony. 2006. Replanning with RRTs. *IEEE International Conference on Robotics and Automation*,1243–1248. https://doi.org/10.1109/ROBOT.2006. 1641879

Tamar Flash and Neville Hogan. 1985. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience* 5, 7 (1985), 1688–1703. https://doi.org/10. 1523/JNEUROSCI.05- 07- 01688.1985

M Mahdi Ghazaei Ardakani, Björn Olofsson, Anders Robertsson, and Rolf Johansson. 2015. Realtime trajectory generation using model predictive control. *IEEE International Conference on Automation Science and Engineering*,942–948. https://doi.org/10.1109/CoASE.2015.7294220

M Mahdi Ghazaei Ardakani, Björn Olofsson, Anders Robertsson, and Rolf Johansson. 2019. Model Predictive Control for Real-Time Point-to-Point Trajectory Generation. *IEEE Transactions on Automation Science and Engineering* 16, 2 (2019), 972–983. https://doi.org/10.1109/TASE.2018.2882764

Elmer G Gilbert, Daniel W Johnson, and S Sathiya Keerthi. 1988. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal on Robotics and Automation* 4, 2 (1988),193–203. https://doi.org/10.1109/56.2083

Sami Haddadin, Alin Albu-Schaffer, Alessandro De Luca, and Gerd Hirzinger. 2008. Collision Detection and Reaction: A Contribution to Safe Physical Human-Robot Interaction. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3356–3363. https://doi.org/10.1109/IROS.2008.4650764

Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. 2017. Robot Collisions: A Survey on Detection, Isolation, and Identification. *IEEE Transactions on Robotics* 33, 6 (2017), 1292–1312. https://doi.org/10.1109/TRO.2017.2723903

Edward T Hall. 1963. A System for the Notation of Proxemic Behavior. *American Anthropologist* 65, 5(1963),1003–1026. https://doi.org/10.1525/aa.1963.65.5.02a00020

Gerhard Hirzinger, Norbert Sporer, Alin Albu-Schaffer, M. Hahnle, Rainer Krenn, A. Pascucci, and Manfred Schedl. 2002. DLR's torque-controlled light weight robot III-are we reaching the technological limits now? *IEEE International Conference on Robotics and Automation*,1710–1716. https://doi.org/10.1109/ROBOT.2002.1014788

Guy Hoffman. 2019. Evaluating Fluency in Human–Robot Collaboration. *IEEE Transactions on Human-Machine Systems* 49, 3 (2019), 209–218. https://doi.org/10.1109/THMS.2019.2904558

Guy Hoffman and Cynthia Breazeal. 2007. Cost-Based Anticipatory Action Selection for Human–Robot Fluency. *IEEE Transactions on Robotics* 23, 5 (2007), 952–961. https://doi.org/10.1109/TRO.2007.907483

Thomas Howard, Colin Green, and Alonzo Kelly. 2010. Receding Horizon Model-Predictive Control for Mobile Robot Navigation of Intricate Paths. *Field and Service Robotics*, 69–78. https://doi.org/10.1007/978-3-642-13408-1_7

Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. 2011. STOMP: Stochastic trajectory optimization for motion planning. *IEEE International Conference on Robotics and Automation*, 4569–4574. https://doi.org/10.1109/ICRA.2011.5980280

Sertac Karaman and Emilio Frazzoli. 2011. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research* 30, 7 (2011), 846–894. https://doi.org/10.1177/0278364911406761

Lydia E Kavraki, Mihail N Kolountzakis, and Jean-Claude Latombe. 1998. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation* 14, 1 (1998), 166–171. https://doi.org/10.1109/70.660866

Lydia E Kavraki, Petr Svestka, Jean-Claude Latombe, and Mark H Overmars. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12, 4 (1996), 566–580. https://doi.org/10.1109/70.508439

Sebastien Kleff, Avadesh Meduri, Rohan Budhiraja, Nicolas Mansard, and Ludovic Righetti. 2021. High-Frequency Nonlinear Model Predictive Control of a Manipulator. *IEEE International Conference on Robotics and Automation*, 7330–7336. https://doi.org/10.1109/ICRA48506.2021.9560990

Kheng L Koay, Kerstin Dautenhahn, Sarah N Woods, and Michael L Walters. 2006. Empirical Results from Using a Comfort Level Device in Human-Robot Interaction Studies. *ACM SIGCHI/SIGART Conference on Human-Robot Interaction*,194–201. https://doi.org/10.1145/1121241.1121276

Kheng L Koay, Emrah A Sisbot, Dag S Syrdal, Mick L Walters, Kerstin Dautenhahn, and Rachid Alami. 2007. Exploratory Study of a Robot Approaching a Person in the Context of Handing Over an Object. *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, 18–24.

Maximilian Krämer, Christoph Rösmann, Frank Hoffmann, and Torsten Bertram. 2020. Model predictive control of a collaborative manipulator considering dynamic obstacles. *Optimal Control Applications and Methods* 41, 4 (2020), 1211–1232. https://doi.org/10.1002/oca.2599

Dana Kulić and Elizabeth Croft. 2007. Physiological and subjective responses to articulated robot motion. *Robotica* 25, 1 (2007), 13–27. https://doi.org/10.1017/S0263574706002955

Przemyslaw A Lasota, Terrence Fong, and Julie A Shah. 2017. A Survey of Methods for Safe Human-Robot Interaction. *Foundations and Trends in Robotics* 5, 4 (2017), 261–349. https://doi.org/10.1561/2300000052

Steven M LaValle. 2006. *Planning Algorithms.* Cambridge University Press, Cambridge, U.K.

Steven M LaValle and James J Kuffner 2001. Rapidly-Exploring Random Trees: Progress and Prospects. In *Algorithmic and Computational Robotics*, Bruce Donald, Kevin Lynch, Daniela Rus (Hrsg.). A K Peters/CRC Press, NewYork. https://doi.org/10.1201/9781439864135

Tsai-Yen Li and Yang-Chuan Shie. 2002. An incremental learning approach to motion planning with roadmap management. *Proceedings 2002 IEEE International Conference on Robotics and Automation,* 3411–3416. https://doi.org/10.1109/ROBOT.2002.1014238

Christina Lichtenthäler, Tamara Lorenzy, and Alexandra Kirsch. 2012. Influence of legibility on perceived safety in a virtual human-robot path crossing task. *International Symposium on Robot and Human Interactive Communication*, 676–681. https://doi.org/10.1109/ROMAN.2012.6343829

Ruikun Luo, Rafi Hayne, and Dmitry Berenson. 2018. Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces. *Autonomous Robots* 42, 3 (2018), 631–648. https://doi.org/10.1007/s10514-017-9655-8

Tim Mercy, Ruben Van Parys, and Goele Pipeleers. 2018. Spline-Based Motion Planning for Autonomous Guided Vehicles in a Dynamic Environment. *IEEE Transactions on Control Systems Technology* 26, 6 (2018), 2182–2189. https://doi.org/10.1109/TCST.2017.2739706

Wolfgang Merkt, Vladimir Ivan, and Sethu Vijayakumar. 2019. Continuous-Time Collision Avoidance for Trajectory Optimization in Dynamic Environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 7248–7255. https://doi.org/10.1109/IROS40897.2019.8967641

Brian Mirtich. 1998. V-Clip: Fast and Robust Polyhedral Collision Detection. *ACM Transactions on Graphics* 17, 3 (1998), 177–208. https://doi.org/10.1145/285857.285860

Michael Neunert, Cédric de Crousaz, Fadri Furrer, Mina Kamel, Farbod Farshidian, Roland Siegwart, and Jonas Buchli. 2016. Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking. *IEEE International Conference on Robotics and Automation*, 1398–1404. https://doi.org/10.1109/ICRA.2016.7487274

Christian Ott. 2008. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots.* Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69255-3

Chonhyon Park, Jia Pan, and Dinesh Manocha. 2012. ITOMP: Incremental Trajectory Optimization for Real-Time Replanning in Dynamic Environments. *International Conference on Automated Planning and Scheduling*, 207–215.

Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. 2009. CHOMP: Gradient optimization techniques for efficient motion planning. *IEEE International Conference on Robotics and Automation,* 489–494. https://doi.org/10.1109/ROBOT.2009.5152817

Tobias Schoels, Luigi Palmieri, Kai O. Arras, and Moritz Diehl. 2020. An NMPC Approach using Convex Inner Approximations for Online Motion Planning with Guaranteed Collision Avoidance. *IEEE International Conference on Robotics and Automation*, 3574–3580. https://doi.org/10.1109/ICRA40945.2020.9197206

John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. 2014. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research* 33, 9 (2014), 1251– 1270. https://doi.org/10.1177/0278364914528132

Emrah A Sisbot and Rachid Alami. 2012. A Human-Aware Manipulation Planner. *IEEE Transactions on Robotics* 28, 5 (2012), 1045–1057. https://doi.org/10.1109/TRO.2012.2196303

Ioan A Şucan, Mark Moll, and Lydia E Kavraki. 2012. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine* 19, 4 (2012), 72–82. https://doi.org/10.1109/MRA.2012.2205651

Wen Sun, Sachin Patil, and Ron Alterovitz. 2015. High-Frequency Replanning Under Uncertainty Using Parallel Sampling-Based Motion Planning. *IEEE Transactions on Robotics* 31, 1 (2015), 104–116. https://doi.org/10.1109/TRO.2014.2380273

Mikael Svenstrup, Thomas Bak, and Hans J Andersen. 2010. Trajectory planning for robots in dynamic human environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 4293–4298. https://doi.org/10.1109/IROS.2010.5651531

Yuval Tassa, Tom Erez, and Emanuel Todorov. 2012. Synthesis and stabilization of complex behaviors through online trajectory optimization. *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 4906–4913. https://doi.org/10.1109/IROS.2012.6386025

Emanuel Todorov and Michael I Jordan. 2002. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience* 5, 11 (2002), 1226–1235. https://doi.org/10.1038/nn963

Noel E Du Toit and Joel W Burdick. 2011. Probabilistic Collision Checking With Chance Constraints. *IEEE Transactions on Robotics* 27, 4 (2011), 809–815. https://doi.org/10.1109/TRO.2011.2116190

Noel E Du Toit and Joel W Burdick. 2012. Robot Motion Planning in Dynamic, Uncertain Environments. *IEEE Transactions on Robotics* 28, 1 (2012), 101–115. https://doi.org/10.1109/TRO.2011.2166435

Minh N Vu, Patrik Zips, Amadeus Lobe, Florian Beck, Wolfgang Kemmetmüller, and Andreas Kugi. 2020. Fast motion planning for a laboratory 3D gantry crane in the presence of obstacles. *IFAC-PapersOnLine* 53, 2 (2020), 9508–9514. https://doi.org/10.1016/j.ifacol.2020.12.2427

Tsuneo Yoshikawa. 1985. Manipulability and redundancy control of robotic mechanisms. *IEEE International Conference on Robotics and Automation*, 1004–1009. https://doi.org/10.1109/ROBOT. 1985.1087283

Matt Zucker, James Kuffner, and Michael Branicky. 2007. Multipartite RRTs for Rapid Replanning in Dynamic Environments. *IEEE International Conference on Robotics and Automation,* 1603–1609. https://doi.org/10.1109/ROBOT.2007.363553

Matthew Zucker, Nathan Ratliff, Anca Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher Dellin, James Andrew Bagnell, and Siddhartha Srinivasa. 2013. CHOMP: Covariant Hamiltonian Optimization for Motion Planning. *International Journal of Robotics Research* 32 (2013), 1164–1193. https://doi.org/10.1177/0278364913488805