Safe and Efficient Human-Robot Collaboration Part II: Optimal Generalized Human-in-the-Loop Real-Time Motion Generation

Roman Weitschat and Harald Aschemann

***Abstract*— The coexistence of humans and robots in fence-less robot cells requires robust safety precautions to prevent humans from being injured. Currently, safety is ensured by limiting the robot velocity, force and power. This results in large cycle times and, hence, very inefficient industrial applications, where no amortization of the robotic system can be expected. In this paper, a novel method for improving the robot performance is presented that still complies with the international safety standards for collaborative robots. The approach of this paper is based on a projection of a human arm motion into the robot's path to estimate a possible collision with the robot. This idea is addressed in an optimization approach by minimizing the time needed by the robot to reach the goal position under human-in-the-loop constraints. The segmented path is optimized by solving a nonlinear programming problem, and the effect of crucial parameters is analyzed. To guarantee a flexible motion of the resulting optimized path, a generalization method using dynamic movement primitives and the compliance of constraints are proposed. Experiments validate this new method that significantly improves the efficiency of human-robot coexistence.**

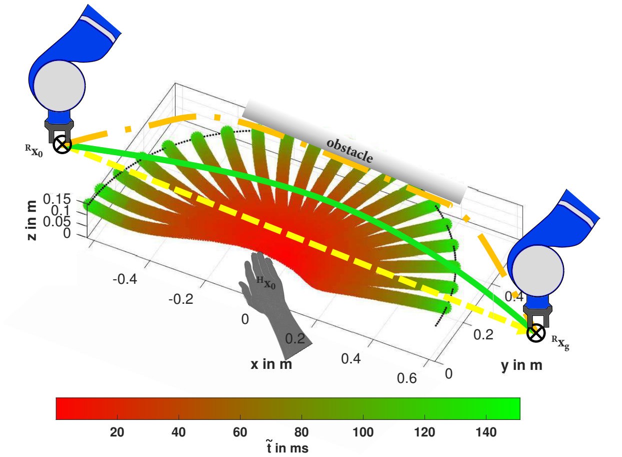


Fig. 1. Path optimization approach with human-in-the-loop constraints for safe and efficient human-robot collaboration.

I. Human arm motion projection approach

In physical human-robot collaboration, safety is the most important topic. Injuries of a human caused by a collision with the robot are not acceptable and have to be avoided. For this, robot motions are usually executed in a very slow manner to prevent from injuring humans. The result is that the applications tend to be rather inefficient, and the installation of such a system is disadvantageous in direct comparison to classical human work. Alternatively, the robots are separated by fences that deteriorate the flexibility of the robotic system. Therefore, a solution is envisaged that enables a flexible but highly efficient collaborative robotic system for industrial applications.

In *Safe and Efficient Human-Robot Collaboration Part I* [1], a new approach towards human arm motion estimation is presented, where human arm motion experiments were executed by a human experimenter and transferred to a simplified dynamic model of the human arm. With this dynamic model, human arm motions are repeatedly projected into the path of a moving robot considering the current velocities of the robot and the human hand, in order to dynamically simulate motions of minimal duration directed to the robot path. The core idea of the human projection approach is to maximize the performance by allowing for collisions between humans and robots, complying with limits given by ISO/TS 15066, so that we can use every possible additional velocity to maximize the efficiency of the application.

As – for safety reasons – the worst case scenario has to be considered, the projection of the human arm motions focuses on possible collisions that may happen in the shortest time possible. Therefore, for every change of the robot or human position or, alternatively, at every time step, it is checked whether a collision may occur and what is the duration of motion to a collision. To entirely ensure safety, in general the whole motion of every body part has to be considered. As a starting point, this paper focuses on the human arm motion with a fixed human base. However, to minimize injuries, ISO standards are used to determine the maximum permissible robot velocity in case of a collision with a human. With this information, the maximum robot velocity can be determined online by

$$^R\dot{x}_{max} = \mathrm{v}_{ISO} +^R \ddot{x}_{max}\tilde{t}, \tag{1}$$

where $\tilde{t}$ is the estimated minimum time a human needs to reach the robot's path, which is obtained by a simulation of the human arm motion. A maximum velocity v_{ISO} is determined by a risk analysis and experiments measuring the maximum force and maximum pressure acting on a human body in case of a collision. The maximum robot deceleration $^R\ddot{x}_{max}$ is defined by the robot dynamic constraints. In Fig. 1, the extension of the human projection approach is illustrated that represents the main contribution of this paper. Robots typically move on the direct path, the yellow dashed line, in the shortest time possible w.r.t. a given maximum velocity. Taking the human and its dynamic behavior into account, however, ideal movements are not

Roman Weitschat is with the Robotics and Mechatronics Center, DLR - German Aerospace Center, Wessling, Germany, roman.weitschat@dlr.de
Prof. Dr. Harald Aschemann is with the Chair of Mechatronics, University of Rostock, Germany harald.aschemann@uni-rotock.de

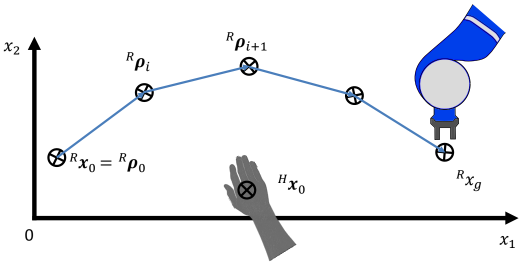
possible because a movement close to the human leads to a velocity reduction, or finally to a still-stand of the robot with $\mathrm{v}_{ISO} = 0\,\mathrm{m/s}$, when no collision is allowed due to safety reasons. Therefore, a path optimization under the consideration of the human-in-the-loop is worthwhile. The green line depicts the time-optimal path through the shared workspace of both human and robot. Here, the distance to the human and the time required to cover the distance are balanced in such a way that a minimum-time path can be found. Additionally, physical constraints like reachability, robot performance, and Cartesian space limitation by objects define the optimal path represented by the dashed orange line. To achieve a flexible and efficient robot motion, our approach includes the following steps:

1) Separating the desired path into multiple segments.
2) Calculating the time a human needs to reach each segment, and consequently the maximum robot velocity of each segment.
3) Solving the numerical nonlinear programming problem to obtain a Cartesian robot path and velocity limits of each segment to reach the goal in minimum time under human-in-the-loop constraints.
4) Generate the necessary trajectory from 3), in particular the desired time-minimal velocity and acceleration.
5) Generalize the optimal motion by generating dynamic movement primitives from 4) to obtain flexible online adaptable motions.
6) Guaranteeing human-in-the-loop constraints during online motion generation.

The order of the steps three to five has to be maintained, to obtain an online adaptable motion generation.

II. Related Work

Human-robot collaboration was enabled by the development of sensitive robotic systems. The DLR-Light-Weight Robot, see [2], is able to measure torques in each joint and to detect collisions in milliseconds. By using this technology in combination with impedance control and a momentum collision observer [3], safety requirements could be specified and improved, see [4] , [5] and [6], and international standards could be defined in the ISO/TS 15066. Regarding safety and efficiency, also trajectory generation has become a more important field in robotics research, in particular with a focus on acceleration and jerk limitations. Usual methods are ramps, cubic splines, polynomials, or B-Splines, which can be found in [7], [8], [9], to generate trajectories through desired via points subject to desired maximum velocity and desired maximum acceleration. In [10], minimum-time trajectories are generated using exponential functions to obtain a minimal duration of movements. According to a desired behavior, application-oriented nonlinear programming methods as well as optimal control methods have been developed. In [11], the solution of a nonlinear programming problem is presented for the purpose of optimal control. As these numerical methods can be only applied offline, online adaptation methods have been developed by using nearest-neighbor and interpolation approaches, see [12] and [13]. Another popular approach to generate task-oriented motions has been developed by the learning-by-demonstration community. Human motions are recorded and transferred into differential equations to generate and adapt robot motions online, cf. [14] and [15], and to realize an online collision avoidance [16].



x_2
${}^{R}\boldsymbol{\rho}_i$
${}^{R}\boldsymbol{\rho}_{i+1}$
${}^{R}\boldsymbol{x}_0 = {}^{R}\boldsymbol{\rho}_0$
${}^{H}\boldsymbol{x}_0$
${}^{R}\boldsymbol{x}_g$
0
x_1

Fig. 2. Robot path consisting of multiple segments with different allowed maximum velocities.

In [17], these state-of-the-art methods were combined to generalize robotic motions with minimal energy for variable-stiffness elastic robots. This method defined a suitable generalization of optimal motions and will be further extended in this paper.

III. Optimal Path Generation

To enable an efficient coexistence of humans and robots, the maximum velocity is not the most relevant parameter to look at. In our approach, the closer the robot moves to the human, the lower is the corresponding velocity limit of the robot, cf. Sec. I. This directly leads to the optimization problem to find the best path regarding the shortest duration of movement subject to the human-in-the-loop constraints. In this section, a segmented path is calculated to determine the shortest distance to the human in each path segment. Thereby, an optimal trajectory to the goal position with the shortest duration possible can be found. The cost function and the constraints are described in detail, and the corresponding results are discussed.

A. Path Segmentation

Given parameters for a robot motion – a point-to-point motion or a straight-line motion, representing the direct path to the goal – are usually the initial as well as the goal position. To define different desired velocities, depending on the distance to the human, the path is separated into multiple segments.

For the path generation, we employ multiple auxiliary points ${}^{R}\rho_i$, or via-points, to define the robot path as shown in Fig. 2. One segment is defined by two auxiliary points ${}^{R}\rho_i$ and ${}^{R}\rho_{i+1}$, where the straight line is given by

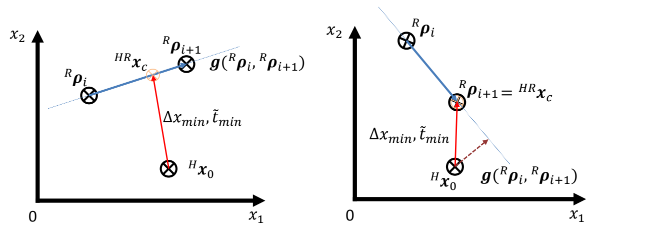
$$\mathbf{g} = \boldsymbol{\rho_i} + \lambda_g \Delta\boldsymbol{\rho_i}, \tag{2}$$

illustrated in Fig. 3. Here, λ_g and $\Delta\boldsymbol{\rho}_i = \boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_i$ define the direction vector. To obtain the shortest distance to the segment at a collision point ${}^{HR}\mathbf{x}_c$, a standard point-to-straight-line calculation can be used. The resulting shortest distance to the segment follows as

$$\Delta\mathbf{x}_{min} = \begin{cases} \|\Delta\mathbf{x}_{min}({}^{H}\mathbf{x}_0, {}^{HR}\mathbf{x}_c)\| & , \ 0 < \lambda_g({}^{HR}\mathbf{x}_c) < 1 \\ \|\Delta\mathbf{x}_{min}({}^{H}\mathbf{x}_0, \boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1})\| & , \ else \end{cases}, \tag{3}$$

with

$$\lambda_g({}^{HR}\mathbf{x}_c) = \frac{{}^{HR}\mathbf{x}_c - \boldsymbol{\rho}_i}{\Delta\boldsymbol{\rho}_i}. \tag{4}$$


x_2
${}^R\boldsymbol{\rho}_i$
${}^{HR}\boldsymbol{x}_c$
${}^R\boldsymbol{\rho}_{i+1}$
$\boldsymbol{g}({}^R\boldsymbol{\rho}_i, {}^R\boldsymbol{\rho}_{i+1})$
$\Delta x_{min}, \tilde{t}_{min}$
${}^H\boldsymbol{x}_0$
0
x_1
x_2
${}^R\boldsymbol{\rho}_i$
${}^R\boldsymbol{\rho}_{i+1} = {}^{HR}\boldsymbol{x}_c$
$\Delta x_{min}, \tilde{t}_{min}$
${}^H\boldsymbol{x}_0$
$\boldsymbol{g}({}^R\boldsymbol{\rho}_i, {}^R\boldsymbol{\rho}_{i+1})$
0
x_1

(a) Segment: shortest distance to a straight line

(b) Segment: shortest distance to an auxiliary point

Fig. 3. Schematic illustration of path segments with minimum distance to the human hand and, finally, shortest time for motions from the current human arm position to the robot path.

Here, $\Delta\mathbf{x}_{min}({}^H\mathbf{x}_0, {}^{HR}\mathbf{x}_c)$ represents the shortest distance to a point on the segment between two auxiliary points, see Fig. 3(a), or the collision point is not in between these two points, then $\Delta\mathbf{x}_{min}({}^H\mathbf{x}_0, \boldsymbol{\rho}_i, \boldsymbol{\rho}_{i+1})$ holds, see Fig. 3(b). The length of a segment $s_i = \|\boldsymbol{\rho}_{i+1} - \boldsymbol{\rho}_i\|$ allows to calculate the time

$$t_{s_i} = \frac{s_i}{{}^R\dot{x}_{max_i}} \tag{5}$$

the robot needs to pass this segment, where ${}^R\dot{x}_{max_i}({}^Hx_0)$ depends on the human position. These segments have to be in the range of the robot and must not lie in between the human body and the human hand. The segmentation strategy is used to gain a fast optimization with low computation time.

B. Nonlinear Programming for Humans in the Loop

The given complexity of the optimization problem leads to numerical optimization. The optimization problem is given by

$$\begin{aligned} & y = \min_{\boldsymbol{\rho}\in\mathbb{R}^n} \Gamma(\boldsymbol{\rho}) && \Gamma : \mathbb{R}^n \mapsto \mathbb{R}^m, && (6)\\ & c_i(\boldsymbol{\rho}) \leq 0 && i = 1, ...m_c, && (7)\\ & h_i(\boldsymbol{\rho}) = 0 && i = 1, ...m_h, && (8) \end{aligned}$$

with the cost function $\Gamma(\boldsymbol{\rho})$ defined by a set of parameters $\boldsymbol{\rho}$, inequality constraints $c_i(\boldsymbol{\rho})$, and equality constraints $h_i(\boldsymbol{\rho})$.

1) Cost Function:

The cost function

$$\Gamma(\boldsymbol{\rho}) = \sum_{i=1}^{n} t_{s_i}({}^R\dot{x}_{max_i}, s_i), \tag{9}$$

corresponds to a minimum-time to reach the goal position with the robot, with ${}^R\dot{x}_{max_i}(\Delta\mathbf{x}_{min}, \tilde{t})$, which can be calculated by Eq. (3) and Eq. (1). Approximately, the shortest time to collision can be calculated assuming a maximum human arm velocity, which minimizes the computation time in an online scheme. In this paper, the maximum velocity of the human arm is given by ${}^H\dot{x}_{max} = 3.3\,\text{m/s}$, which stem from the arm motion experiments discussed in Sec. I. The resulting time for each segment, hence, can be calculated by $\tilde{t} \approx \tilde{t}_{\approx} = \frac{\Delta\mathbf{x}_{min}}{{}^H\dot{x}_{max}}$.

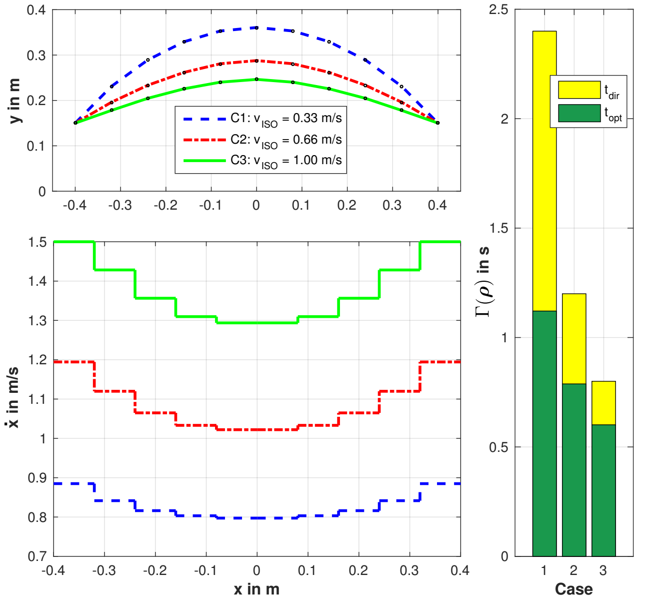


Fig. 4. Optimization of robot motion with human-in-the-loop constraints. The upper left plot depicts the path in the Cartesian space by solving the optimization problem. The different lines correspond to different collision velocities (Case 1-3) determined by ISO standards for human-robot-collaboration. The left bottom plot depicts the maximum robot velocity for each segment under the human-in-the-loop constraints with a constant robot deceleration of 10 m/s^2. The right plot shows the different costs of the straight-line motion (yellow bar) and the optimized motion (green bar). The necessary duration of movement is smaller than 50 % of the direct motion.

2) Constraints:

For the optimization problem, several inequality constraints are defined

$$\begin{aligned} 0 \leq {}^R\dot{x} &\leq {}^R\dot{x}_l, && (10)\\ {}^R\boldsymbol{x}_{min} \leq {}^R\boldsymbol{x} &\leq {}^R\boldsymbol{x}_{max}, && (11)\\ {}^R\boldsymbol{x} &\geq {}^H\boldsymbol{x}_0. && (12) \end{aligned}$$

The maximum robot velocity ${}^R\dot{x}$ shall not exceed the robot velocity limit given by ${}^R\dot{x}_l$, the Cartesian limitation ${}^R\boldsymbol{x}$ is defined by the robot range ${}^R\boldsymbol{x}_{min}$ and ${}^R\boldsymbol{x}_{max}$, which prevents a movement through its own base. Finally, the robot is not allowed to move below the human hand position, as the reachability could let the robot collide with the human body.

C. Results with Parameter Comparison

Given the properties of the robot and the geometric constraints, the optimization may lead to different results. In dependence on the maximum possible velocity of the robot, e.g., the path may change significantly. Regarding a robot without any velocity limits and any environmental limitation, it would be able to move far beyond the reachability of the human in arbitrarily short time. Since the initial position and the goal position may be located inside the reachability of the human, an optimal motion has to be found.

Due to given physical limitations, every robot is subject to a certain velocity limit. In Fig. 4, the optimization of the path from an initial position ${}^R\boldsymbol{x}_0 = [-0.4\,0.15]^T$ to the goal position ${}^R\boldsymbol{x}_g = [0.4\,0.15]^T$, with a total distance of 0.8 m, is illustrated. The human hand position is located at ${}^H\boldsymbol{x}_0 = [0.0\,0.05]^T$, which means that the robot would pass

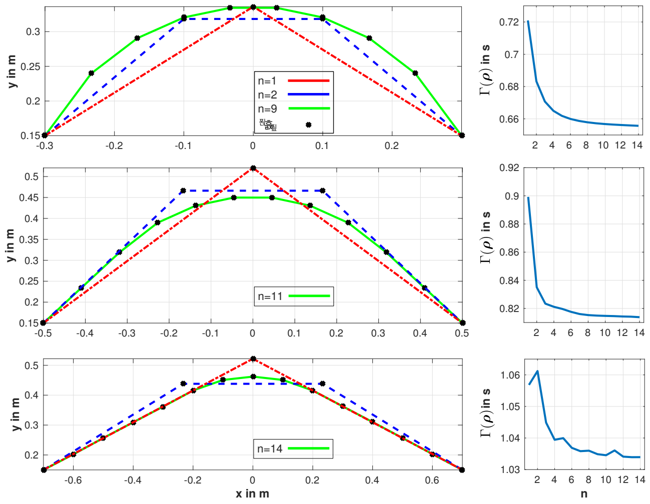


Fig. 5. Solution of the optimization problem with different distances and number of segments n: distances are $0.6\,m$ (upper left plot), $1\,m$ (left middle plot), $1.4\,m$ (left bottom plot), with a different number of segments. The right plots depict the cost changes in dependence on the number of segments.

the human arm on a direct path in a distance of 0.1 m. In the upper left plot of Fig. 4, the resulting Cartesian motion is illustrated for three different cases and a number of $n = 10$ path segments. The first case (C1) shows a motion with a maximum allowed collision velocity of 0.33 m/s determined by ISO/TS 15066, C2 with 0.66 m/s and C3 with 1 m/s. The faster the robot is allowed to move with a both force- and power-limited robot, the closer the movement is to the human, which is the result of a balance between robot velocity and distance to move.

The chosen method using segments also affects the path and the cost depending on the number of segments. In Fig. 5, results are presented for different motion distances and alternative segment numbers. In the upper left plot, three paths are depicted with $n = 1, n = 2$, and $n = 9$ segments defining the path. The motion distance for the robot is 0.6 m in the left plot, $1\,m$ in the middle plot and $1.4\,m$ in the left bottom plot. The right row illustrates the final costs depending on the number of segments. It becomes obvious that for $n > 4$ the number of segments has only a minor influence on the costs, which implies that a highly resolved incrementation of the path is not necessary.

From the optimization we obtain the desired path information and the corresponding velocity constraints for each segment. In a subsequent step, the path has to be converted into a trajectory to provide a suitable input to the robot axis controllers. Therefore, in the following section a minimal-time trajectory generation is presented.

IV. Adapted Trajectory Generation

By analyzing the resulting velocity profile and existing trajectory generating methods based on splines, see [18], [19], [9], commonly used polynomials or B-Splines are found to be incompatible with the requirements of minimum-time motions. The optimization routine only provides the path positions. Therefore, desired velocities and accelerations have to be generated by a trajectory module to enable a smooth robot motion. As the trajectory generation is not in focus of this paper, it is summarized as follows:

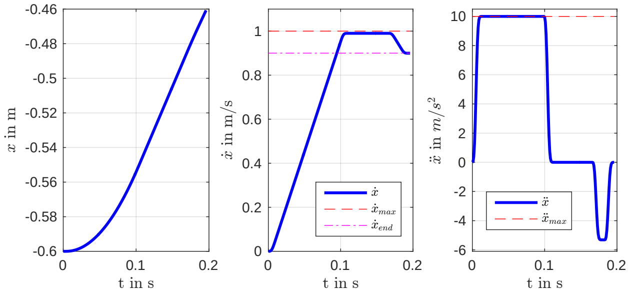


Fig. 6. Trajectory generation to obtain desired velocities and acceleration on the path calculated by solving the nonlinear programming problem.

The trajectory generation regarding minimum-time motions is inspired by [10], where an exponential function is used to define the velocity profile. In our case, an exponential function of third order is chosen to define the acceleration according to

$$ {}^{R}f_{\ddot{x}}(t) = {}^{R}\ddot{x}_{max} - {}^{R}\ddot{x}_{max}e^{-\eta * t^{3}}, \tag{13} $$

with ${}^{R}\ddot{x}_{max}$ as the maximum acceleration of the robot. The parameter η defines the maximum slope of the acceleration, i.e., the jerk. Similar to [10], we divide the path into multiple motion segments, i.e., jerk phases and phases with constant acceleration. In Fig. 6, the trajectory generation is illustrated: the curves depict the position, velocity and acceleration obtained by numerical integration according to

$$ {}^{R}\mathbf{f}_{x}(t) = \int_{0}^{t_f} {}^{R}\mathbf{f}_{\dot{x}}(t)dt = \int_{0}^{t_f}\int_{0}^{t_f} {}^{R}\mathbf{f}_{\ddot{x}}(t)dtdt. \tag{14} $$

Note that no velocity limits are violated. Here, the velocity at the end of the current segment (magenta dashed-dotted line) in the middle plot is identical to the maximum velocity of the following segment. The maximum values for velocity and acceleration are indicated by red dashed lines.

V. Generalization of Motions

Usually, the optimization of the path is executed offline, which would result in an inflexible motion generation scheme due to a high computational load. Since typical collaborative robot applications involve dynamic environments – for example the motion of humans or changing goal or initial positions – the robot motions have to be adapted online accordingly. In Fig. 7, two robot motions are illustrated. The blue solid line depicts the optimized motion of the robot passing the shared workspace. The orange dashed line depicts a new motion which has different initial and terminal conditions.

A more flexible online motion generation can be achieved by using movement primitives. As described in [17], optimized motions serve as input to a learning algorithm of DMPs. In the following section, DMPs are described. Moreover, a path adaptation is presented similar to via-point motion primitives that contribute to a higher path stability and, finally, to the compliance with specified maximum velocities.

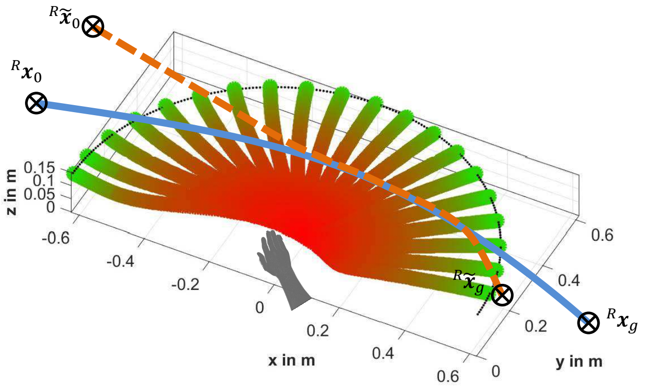


Fig. 7. Generalization of robot motions: the blue solid line depicts the optimal robot path for the given human reachability. The orange dashed line shows the changed path with different initial and goal positions.

A. Dynamic Movement Primitives for Optimal Motions

By inserting the generated desired trajectory into a second-order differential equation as described in [17], a force can be calculated by

$$\mathbf{f}^*(t) = \tau^{2\,R}\mathbf{f}_{\ddot{x}}(t) + D\tau^{R}\mathbf{f}_{\dot{x}}(t) + \kappa(t)({}^{R}\mathbf{f}_{x}(t) - \mathbf{g}(t)), \quad (15)$$

reproducing the motion in each Cartesian direction. Here, τ denotes the duration of motion and $\kappa(t)$ a time-varying stiffness function. Moreover, $\mathbf{g}(t)$ is a time-varying goal function, generated by Gaussian kernels, which are described below, and D characterizes a damping term in the differential equation. In a next step, this force is approximated by Gaussian basis functions

$$\mathbf{f}^*(t) \approx \mathbf{f}_{\approx}(s_c) = \frac{\sum_{i=1}^{N} \mathrm{w}_i \psi_i(s_c)}{\sum_{i=1}^{N} \psi_i(s_c)} s_c. \quad (16)$$

Here, a set of weights $\mathbf{w} \in \mathbb{R}^N$, Gaussian kernels $\psi_i(s_c)$, and a canonical system

$$s_c(t) = e^{-\frac{\alpha}{\tau}t} \quad (17)$$

are introduced, which can be described as a function with a continuous decrease from 1 to 0, parameterized by the slope $\alpha > 0$. The final trajectory can be calculated online by

$$\begin{aligned}{}^{\mathbf{R}}\mathbf{x_d}(t) =& \frac{1}{\tau^2}\int_0^{t_f}\int_{\tau}^{t_f}[\mathbf{f}_{\approx}(\mathbf{s_c}) + \kappa(t)(\mathbf{g}(t) - {}^{\mathbf{R}}\mathbf{x}) \\ &- D\tau{}^{\mathbf{R}}\dot{\mathbf{x}}(t)]\ \mathrm{d}t\ \mathrm{d}t + {}^{\mathbf{R}}\mathbf{x}(0)\end{aligned} \quad (18)$$

to obtain the online generated desired position. In order to comply with all given constraints, we developed novel methods presented in the following sections.

B. Path Adaptation for Via-Point Movement Primitives

By changing the goal position of the robot, the common dynamic movement primitive maintains the principle shape of the path but adapts it as required. This may lead, however, to a violation of a spatial limit, e.g., moving through its own base. In Fig. 8, the upper plot shows the original motion (blue solid line) with the initial position at ${}^R\mathbf{x}_0 = [-0.6\,0.15]$ and ${}^R\mathbf{x}_g = [0.6\,0.15]$. Then, the goal position was changed to ${}^R\mathbf{x}_g = [0.7\,0.25]$, and the corresponding optimization results are depicted as black dashed line. The optimization with the same starting point but different goal position keeps the shape of the DMP motion, which is depicted as red dashed-dotted line. While the shape has not changed, the spatial motion has been extended significantly. Since the robot base is located in ${}^R\mathbf{x}_b = [0.0\,0.6]$ in this example, the robot would have to move through the own base, which is impossible. Therefore, spatial constraints are integrated in such a way that the maximum y-position defined by ${}^R x_y \leq 0.45\,\mathrm{m}$ is not violated. In the bottom plot of Fig. 8, the constrained motions are compared to each other. The blue solid line represents the original motion, the black dashed line the optimized path, and the red dashed-dotted line the resulting DMP. Obviously, the spatial limits are not violated by both paths, the optimal and the DMP path. We achieved this behavior, by developing goal functions with an auxiliary point on the original path, similar to via-points, that can be changed as well. As we directly change the goal, it is necessary to use a function which is more than two times continuously differentiable like exponential functions. A simple ramp, however, would lead to an unwanted behavior and cause jumps in the velocity. The goal functions are calculated according to

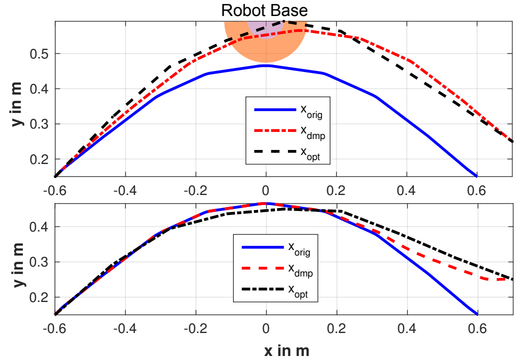


Fig. 8. Via-point path generation using DMPs with spatial limitation. Upper plot depicts the motion with a changed goal position and no spatial constraints. The bottom plot illustrated the constraint motion with the presented method.

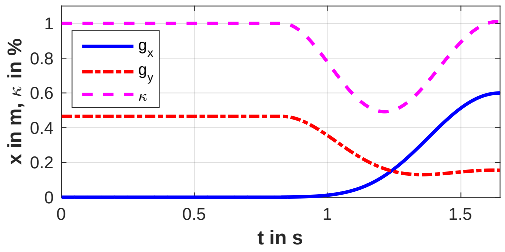


Fig. 9. Via-kernel trajectory for a more robust path generation with different initial conditions. An auxiliary point is placed at turning point of path.

$$\boldsymbol{g}(t) = \sum_{i=1}^{m} \psi_{\varrho_i}(s_c)\boldsymbol{g}_{\varrho} \qquad i = 1, ...m, \quad (19)$$

where $\boldsymbol{g}_{\varrho}$ defines the via-goals of the motion. The Gaussian basis functions are given by

$$\psi_{\varrho_i}(s_c) = e^{-h_{\varrho_i}(s_c(t) - s_{\varrho_i})^2}. \quad (20)$$

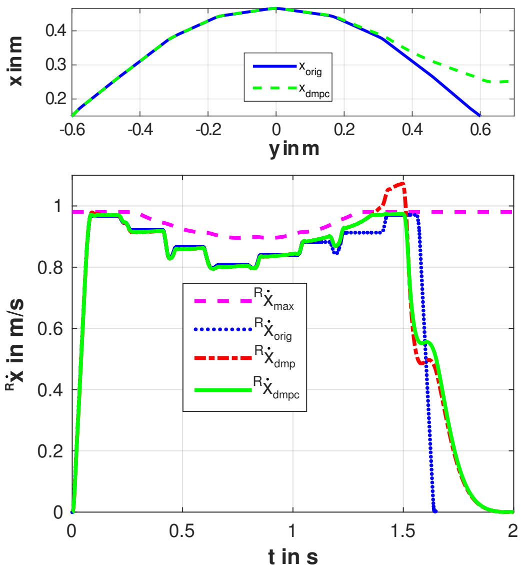


Fig. 10. Generalization of motions complying with the desired velocity limits. Comparison of velocity profiles generated by DMPs with changed goal positions.

The positions of the basis functions can be determined by $s_{\varrho_i} = e^{-\alpha\,\varrho_i}$, where ϱ_i denotes the percentage of the time duration, given in a range of $0 \leq \varrho_i \leq 1$. The width of each kernel is finally calculated by

$$h_{\varrho_i} = \beta_\varrho (e^{-\alpha(\varrho_i - \frac{\Delta t}{\tau})} - e^{-\alpha \varrho_i})^{-1}, \tag{21}$$

with a width parameter β_ϱ. To obtain a smooth behavior between the auxiliary goals and the final goal, the varying-stiffness function $\kappa = \sum_{i=1}^{m} \psi_{\varrho_i}$ is weakened during goals shifts.

In Fig. 9, the results are illustrated. The magenta dashed line depicts the stiffness function $\kappa(t)$, the red dash-dotted line the goal function for the y-coordinate, and the blue solid line stands for the goal function for the x-coordinate. To summarize, using these goal and stiffness functions, spatial constraints for DMPs can be easily met.

C. *Velocity-Constrained Movement Primitives*

Dynamic movement primitives are generated by a second-order differential equation. Consequently, a direct determination of the maximum velocity is not possible. The trajectory is generated by forces and changing the forces by, e.g., potential fields affects the behavior of the trajectory in an unforeseeable way. This includes changes in the velocity profile of the movement. As we limit the velocities to ensure human safety in a close coexistence of robots and humans, the maximum velocity must not be exceeded during the robot motion but should be changeable over time. The same problem arises when changing the goal position $^R x_g$ of the robot. To finally keep the velocity below the maximum, a force vector is added as follows

$$\mathbf{f}_c(^R\dot{\boldsymbol{x}}) = \begin{cases} -K_c(^R\dot{x}_t - (\dot{x}_{max} - \mu))^{\boldsymbol{R}}\dot{\boldsymbol{x}} & , ^R\dot{x}_t > {}^R\dot{x}_{max} - \mu \\ 0 & , ^R\dot{x}_t \leq {}^R\dot{x}_{max} - \mu \end{cases} \tag{22}$$

Here, the current robot velocity is given by $^R\dot{x}_t = \|^R\dot{\boldsymbol{x}}\|$. The parameter μ is a threshold for the velocity potential and K_c the potential value. The velocity can be limited by adapting the DMP input force

$$\tilde{\mathbf{f}}_\approx(s_c) = \mathbf{f}_\approx(s_c) + \mathbf{f}_c(^R\dot{x}) \tag{23}$$

by a regulation force $\mathbf{f}_c(^R\dot{x})$. The two parameters μ and K_c can be determined by fulfilling the inequality constraint $\mathbf{f}_c(^R\dot{x}) \geq \mathbf{f}_\approx(s_c) + \kappa(t)(\mathbf{g}(t) - {}^\mathbf{R}\mathbf{x_d}) - D^R\dot{x}_{max}$.

In Fig. 10, the results are illustrated. The upper plot depicts the Cartesian path of the adapted motion (green dashed line) and the original motion (blue solid line). In the bottom plot, the velocities of the different motions are compared to each other: the maximum allowed velocity (magenta dashed line) and the original motion (blue dotted line). The red dashed-dotted line shows the velocity profile with a not-regulated DMP force input. Obviously, the DMP motion velocity significantly exceeds the allowed maximum velocity. Theoretically, safety is no longer guaranteed and the ability to stop with the deceleration. The green solid line shows the adapted behavior of the DMP with a regulated force vector. Now the velocity limits are not violated anymore, and the safety requirements for this motion can be met.

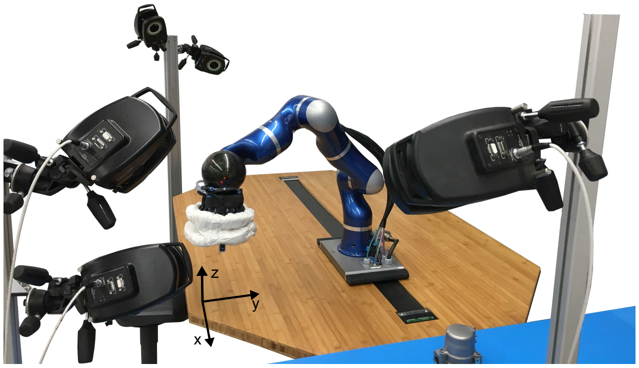


Fig. 11. Experimental setup including a Vicon tracking system and an 8-DoF light weight robotic system. During the experiments, the experimenter is sitting on the left hand side in front of the workbench.

VI. Experiments

In this section, selected experimental results are presented: The setup is described in detail, the experiments are explained and, finally, the corresponding results are discussed.

A. *Experimental Setup*

The octagon collaborative workbench consists of a DLR light-weight robot mounted on a linear axis with a maximum reachability of 2.6 m × 1.6 m as shown in Fig. 11. The experimental setup is surrounded by a Vicon tracking system, which enables position measurements at a frequency of 100 Hz – using special markers that are placed on the back of a human experimenter's hand. For the considered tasks, the human is sitting in front of the workbench with the hand placed on the surface of the table. The position of the hand is changed or moved during the task. The overall 8-Degree of Freedom (DoF) robotic system is controlled by a Cartesian impedance controller with a stiffness of 800 N/m. The online trajectory generation to realize the velocity limited direct motions is realized according to [20].

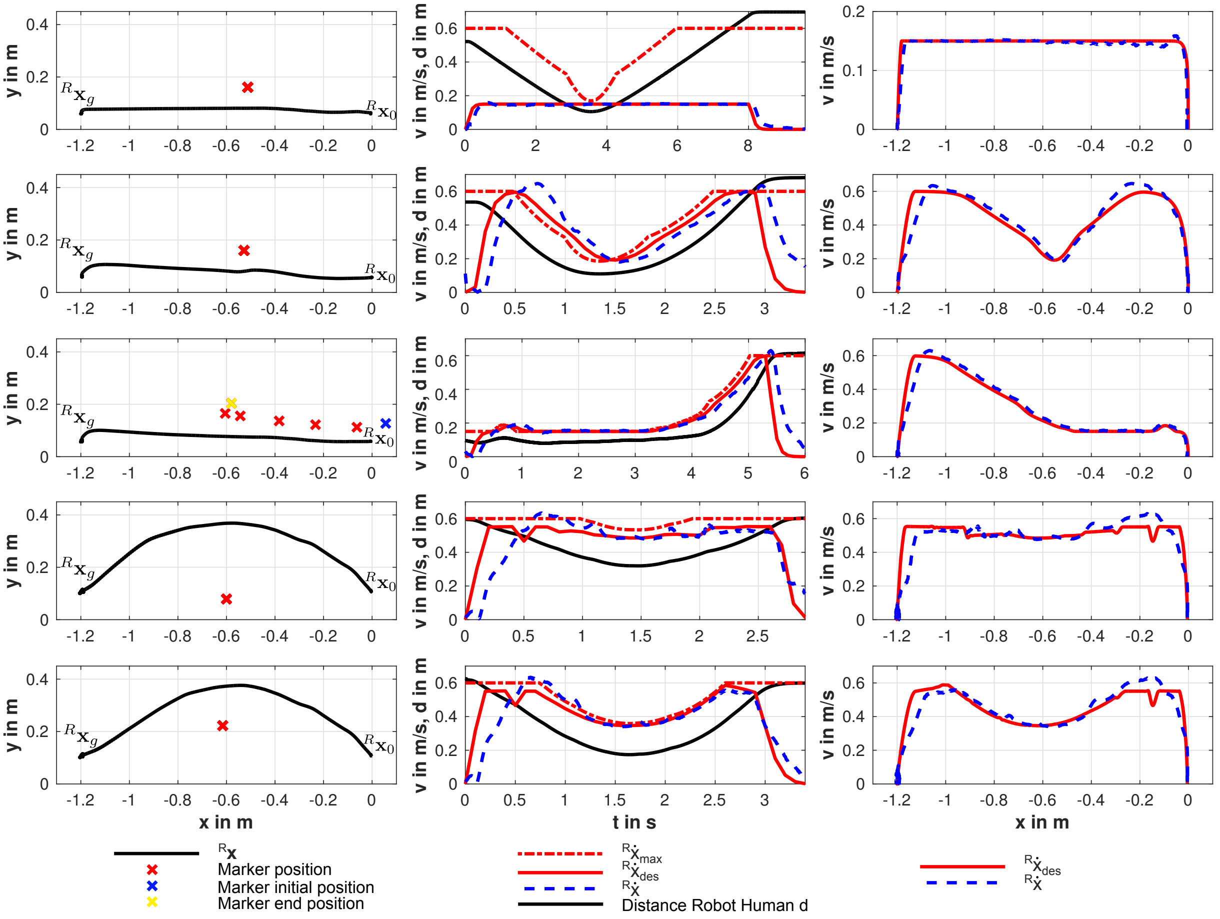


Fig. 12. Experimental results including three different types of motion, with a focus on complying with the velocity constraints.

B. Experimental Results

For the experiments, the robot is allowed to move at $v_{ISO} = 0.15\,\text{m/s}$, which has been determined in collision experiments with the robot end-effector, taking into account the maximum forces and pressure caused by a collision. With this velocity-limit the requirements of ISO/TS 15066 are fulfilled. The final acceleration is $^R\ddot{x}_{max} = 3\,\text{m/s}^2$, which has been determined by simple motion executions. Within the experiments, the robot is using the entire 8 DoF system with the linear axis and is moving from an initial position $x = 0\,\text{m}$ to a goal position $x = -1.2\,\text{m}$. Due to safety reasons, the overall maximum velocity is limited to $0.6\,\text{m/s}$. For the experiments three main scenarios are compared in the sequel.

Typical motion with constant velocity (Fig. 12 row 1): The first scenario follows the usual approach with a fixed maximum velocity of $v_{ISO} = 0.15\,\text{m/s}$. For an intuitive interpretation, the motion was executed only in x-direction, which is depicted in the right column. Nevertheless, the motion can be executed in every direction. In the left column, the Cartesian position is depicted in x-y-plot. The red marker '$\times$' shows the constant hand position at $x \approx -0.5\,\text{m}$. The middle plot depicts four different values: The red solid line stands for the desired velocity given as input to the robot, the blue dashed line depicts the resulting velocity measured at the robot end-effector, the red dashed-dotted line indicates the maximum allowed velocity under human-in-the-loop constraints with the given maximum deceleration ability of the robot, and the black solid line depicts the distance of the robot to the human. The resulting motion requires a duration of movement of more than $8\,\text{s}$ for a motion in just one direction. This justifies the assessment that this application is very inefficient and the potential of robotic system is not exploited.

Adaptable velocity on a direct path (Fig. 12 row 2-3): The upper second row in Fig. 12 illustrates the motion with the proposed calculation of an allowed maximum velocity from Eq. (1). This allowed velocity depends on the time the human needs to reach the robot path and accounts for the maximum deceleration ability. The marker is again placed at the same constant position. Obviously, the measured robot velocity is violating the maximum allowed velocity a little – which is a result of the low-stiffness impedance control of the 8 DoF system. This could be fixed by either a controller with a higher control bandwidth as well as a higher stiffness, or by adding thresholds to the maximum desired velocity. What becomes visible in the second row is that the total time needed for the motion is now only $3.4\,\text{s}$. The performance improvement is 2.47 times the motion limited by ISO – with an identical risk of injury. This also holds in the case of a moving human hand, which is illustrated in the third row. A small distance of the human to the robot leads to the minimum velocity. There, a collision may occur but the risk of an injury caused by a collision is eliminated by the collision experiments.

DMP with human-in-the-loop constraints (Fig. 12 row 4-5): In a third experiment, movement primitives with an optimized motion as presented above are shown in the fourth row of Fig. 12. As can be seen in the left plot, the robot does not move on the direct path anymore but on an optimized path. The marker is placed at $x \approx -0.5\,\mathrm{m}$ and $y \approx 0.2\,\mathrm{m}$. Using DMPs, the goal positions can now be easily changed, which has already been proven in many other publications. The more important questions are the achieved reduction in the duration of movement and whether the constraints are satisfied. Comparing the motion with the first two experiments, the significant efficiency improvement becomes obvious, while safety is still ensured. The duration for the motion is further reduced to $2.8\,\mathrm{s}$, which exploits the potential of the robotic system. In the bottom row, the same motion was executed but with a smaller marker distance to the original path of the robot. This implies lower velocity limits during the motion. In the bottom middle plot, it is shown that the limits decrease and the desired maximum velocity is always below the admissible limits during the motion.

This proves that the presented method – optimized and generalized motions generated with DMPs – is suitable even with human-in-the-loop constraints as well as spatial constraints. To achieve a more adaptable optimal motion generation – especially considering human arm changes – multiple DMP motions can be generated and the weights can be interpolated as presented in [17].

VII. Conclusion

The contribution of this paper is an efficiency improvement in tasks with a coexistence of humans and robots. The approach is based on a projection of the human arm motion to predict possible collisions between the robot and the human. This information is exploited to increase the robot velocity if possible. With this method, however, the direct path from an initial position to a goal position turns out to be inefficient regarding the duration of movement. Therefore, the path is segmented, and the auxiliary points are optimized by nonlinear programming. The resulting optimal path defines the maximum admissible velocity for the robot, that reduces the duration of movement significantly in comparison with the direct motion. To enable a flexible real-time usage, dynamic movement primitives are used for a generalization, which are extended to via-point primitives to guarantee the compliance with Cartesian constraints as well as the maximum velocity. Since only the human hand position is considered, further developments have to include the whole human body. In future research, moreover, the proposed method has to be extended considering restricted areas and additional worst case situations have to be analyzed.

Acknowledgement

This work has been partially funded by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 731540

References

[1] R. Weitschat, J. Ehrensperger, and H. Aschemann, "Safe and efficient human-robot collaboration part I: estimation of human arm motions," in *Accepted International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[2] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger, "The dlr lightweight robot: design and control concepts for robots in human environments," *Industrial Robot: an international journal*, vol. 34, no. 5, pp. 376–385, 2007.

[3] A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger, "Collision detection and safe reaction with the dlr-iii lightweight manipulator arm," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2006, pp. 1623–1630.

[4] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: Measurements, analysis and new insights," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, 2009.

[5] N. Mansfeld, B. Djellab, J. R. Veuthey, F. Beck, C. Ott, and S. Haddadin, "Improving the performance of biomechanically safe velocity control for redundant robots through reflected mass minimization," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5390–5397.

[6] K. Ikuta, H. Ishii, and M. Nokata, "Safety evaluation method of design and control for human-care robots," *The International Journal of Robotics Research*, vol. 22, no. 5, pp. 281–297, 2003.

[7] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[8] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[9] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[10] Z. Rymansaib, P. Iravani, and M. Sahinkaya, "Exponential trajectory generation for point to point motions," in *International Conference on Advanced Intelligent Mechatronics*. IEEE, 2013, pp. 906–911.

[11] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.

[12] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, "Trajectory planning for optimal robot catching in real-time," in *International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 3719–3726.

[13] A. Werner, R. Lampariello, and C. Ott, "Optimization-based generation and experimental validation of optimal walking trajectories for biped robots," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 4373–4379.

[14] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning rhythmic movements by demonstration using nonlinear oscillators," in *International Conference on Intelligent Robots and Systems (IROS)*, no. BIOROB-CONF-2002-003, 2002, pp. 958–963.

[15] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 763–768.

[16] F. Stulp, E. Oztop, P. Pastor, M. Beetz, and S. Schaal, "Compact models of motor primitive variations for predictable reaching and obstacle avoidance," in *International Conference on Humanoid Robots (Humanoids)*. IEEE, 2009, pp. 589–595.

[17] R. Weitschat, S. Haddadin, F. Huber, and A. Albu-Schäffer, "Dynamic optimality in real-time: A learning framework for near-optimal robot motions," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 5636–5643.

[18] M. W. Mueller, M. Hehn, and R. D'Andrea, "A computationally efficient motion primitive for quadrocopter trajectory generation," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1294–1310, 2015.

[19] T. Kröger, "On-line trajectory generation in robotic systems, volume 58 of springer tracts in advanced robotics," 2010.

[20] R. Weitschat, A. Dietrich, and J. Vogel, "Online motion generation for mirroring human arm motion," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4245–4250.