Dynamic Motion Planning for Robots in Partially Unknown Environments

Sami Haddadin, Rico Belder, and Alin Albu-Schäffer

Institute of Robotics and Mechatronics, DLR - German Aerospace Center, Wessling, Germany, Tel: +49(0)8153 28 1047; e-mail: sami.haddadin@dlr.de

Abstract: In both domestic and also industrial settings robotic Co-Workers are expected to become a commodity. Even though the particular application areas may vastly change, a robot always needs to act in a dynamic and partially unknown environment. It shall reactively generate motions and prevent upcoming collisions. If contact is desired or inevitable, it has to handle it robustly and safely. For preventing collisions in a real-time fashion the Circular Fields method is a powerful scheme, which we developed further and evaluated extensively. After an initial analysis in rather complex 2D simulations, we extend the evaluation to 3D as well as 6D, where we introduce a hybrid strategy based on Circular and Potential Fields. Finally, the 6D implementation of a hybrid Circular & Potential Fields approach is used to perform the experimental analysis for static multi-object parcours and to avoid a dynamically moving human in a 6D task motion. Based on the algorithms for collision avoidance we also develop and experimentally verify an algorithm for tactile exploration of complex planar 3D wire elements, whose structure is a-priori unknown.

Fig. 1. Physical cooperation between humans and robots.

1. INTRODUCTION

Future robots are sought to become an integral part of our daily life as multi-purpose service assistants in our homes. Apart from such domestic applications, flexible and versatile robots may also entirely relieve us from monotonous and physically demanding work in industrial settings. In dangerous or even life-threatening surroundings, as e.g. deep-space or underwater, they may even replace humans entirely. However, despite vastly changing particular application areas, a robot always has to be able to act in a dynamic and partially unknown environment. It shall reactively generate motions and prevent upcoming collisions. If contact is desired or inevitable, it needs to robustly and safety handle it. Especially physical Human-Robot Interaction (pHRI) is a field in which such behavior is of immanent importance. As human and robot shall collaborate very closely (see Fig. 1), the problem of generating "human-friendly" motions is of large interest. Even though the close interaction of human and robot in the domestic and industrial sector was always proclaimed to open up entirely new possibilities for service applications and production processes, several problems are still to be tackled before finally achieving this ambitious goal. A particularly important problem indeed is the generation of safe motions in human vicinity, which let the robot safely circumvent dynamic obstacles in real-time. However, up to now reactive motion capabilities were usually developed for mobile robots, Yamamoto and Yun [1995], Ögren et al. [2000], where the complexity of the avoidance problem is limited by nature. For multi-degree of freedom (DoF) articulated manipulators, on the other hand, only few algorithms were developed, Siciliano and Khatib [2008], Khatib [1985], Yamamoto and Yun [1995], Ögren et al. [2000], Brock and Khatib [2002], pf which some were tested experimentally. This is mainly due to the fact that such abilities were unnecessary in real-world applications because of numerous reasons: Articulated robots were almost exclusively used for industrial applications with only static or very predictable environmental constraints. These applications require only precomputed trajectories that usually remain unchanged and humans are segregated with fences from the robot. Recently, however, first articulated robots have gained the mechanical and control capabilities for coping with local uncertainties in their environment and during (physical) interaction with humans. Powerful and highly sensorized arms as e.g. the DLR Lightweight Robot III (LWR-III), Albu-Schäffer et al. [2007] or the Barrett WAM Arm, Townsend and Guertin [1999] were developed over the last decade. Those systems are particularly well designed for applications that incorporate Human-Robot Interaction. Well suited control strategies were developed for this new type of robots for nominal interaction control and sensitive collision detection and reaction, Albu-Schäffer et al. [2007], Haddadin et al. [2008], De Luca et al. [2006], Wang et al. [2007], Ebert and Henrich [2002]. This significant progress finally necessitates the development of appropriate real-time collision avoidance methods that are particularly well suited for the robot's demands and capabilities in pHRI tasks, however, at the same time not being prone to get stuck in local minima.

In this paper we approach the aforementioned problem by extensively evaluating a particularly promising approach: the Circular Fields method. We develop an extension and analyze the scheme in numerous 2D, 3D, and 6D scenarios

⋆ This work has been partially funded by the European Commission's Sixth Framework Programme as part of the project VIACTORS under grant no. 231554.

with respect to its ability to generate reactive motions in real-time with limited local knowledge of a possibly dynamic and complex environment. Furthermore, simultaneous goal convergence, while providing coordinated movement in translation and orientation, is a further primary requirement. For implementing such a motion behavior we significantly extend and combine Circular Fields with Potential Fields, such that they generate smooth, intuitive trajectories and/or virtual disturbance signals that can be fed to low-level controllers. We apply the method to different levels of control and motion generation in order to analyze its respective effectiveness. Furthermore, we provide promising experimental results for rather complex real-world examples, which up to now seemed solvable by global motion planning techniques only[1]. In addition to addressing the reactive motion control for pHRI, we also extend the algorithms such that a powerful scheme for physically exploring unknown wire objects with tactile information only is generated. The algorithm is used to explore complex planar 6D wires in both simulation and experiment. It enables the robot to incrementally build a geometric interaction map of the object and updates it according to the respective sensory input. All methods are experimentally verified on the DLR Lightweight Robot III (LWR-III).

Videos showcasing the performance of our algorithms can be found at http://www.safe-robots.com/ifac2011-cfca.html.

2. ALGORITHM

In this section we introduce the Circular Field method together with some modifcations, which significantly enhance the original scheme. Apart from generating better motion behavior in the sense of intuitive behavior for obstacle avoidance, we also outline how to significantly reduce the calculation load.

2.1 Point mass dynamics

In this paper we associate a virtual attractor dynamics with the robot, which is directly affected by virtual forces $\mathbf{F}_v$ generated by the CF algorithm[2]. This causes the virtual particle n with mass m to avoid collisions, while converging to the final goal. The assumed point mass dynamic system equation is

$$m\ddot{\mathbf{x}} = -k_a(\mathbf{x} - \mathbf{x}_d) + \mathbf{F}_v - k_d\dot{\mathbf{x}}, \tag{1}$$

where the quadratic potential field for goal attraction is

$$U_a = \frac{1}{2}k_a(\mathbf{x} - \mathbf{x}_d)^T(\mathbf{x} - \mathbf{x}_d). \tag{2}$$

and the virtual obstacle force $\mathbf{F}_v = \sum_j {}^j\mathbf{F}_v$ is composed of the subforces generated by the distinct obstacle objects (OOs). $\mathbf{x}, \mathbf{x}_d, k_a, k_d$ are the position, desired goal, (positive) attractive constant, and (positive) damping constant.

2.2 Circular Fields

The Circular Fields approach (CFs), Singh et al. [1996], is based on the generation of artificial electro-magnetic-fields (B-Fields) that are generated by virtual current elements associated to the surface of obstacles[3]. Thus, in contrast to electrostatic charges for Potential Fields (PFs), the analogon of dynamical electric charges are consulted. A modified version of a B-Field ${}^j\mathbf{B}$ generates obstacle forces ${}^j\mathbf{F}_v$ that are vertical to the avoiding object (AO) velocity vector $\dot{\mathbf{x}}$ (see Fig. 2) and are defined as

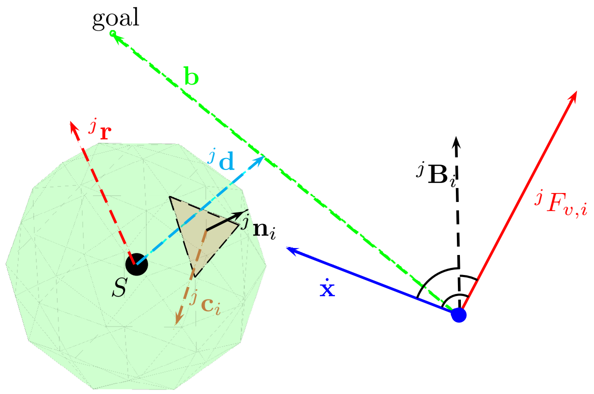


Fig. 2. Principle of the current definition for Circular Fields for obstacle j with surfaces i and robot n (with surface elements k for the 6D case).

$${}^j\mathbf{F}_v = \dot{\mathbf{x}} \times \sum_i {}^j\mathbf{B}_i. \tag{3}$$

${}^j\mathbf{F}_v$ causes a re-orientation of the instantaneous velocity vector and yields a guidance of k around the obstacle object j. This process dissipates no kinetic energy from the system. The local circular field ${}^j\mathbf{B}_i$ of each surface element i (of an obstacle j), acting on the virtual particle is defined as

$${}^j\mathbf{B}_i = I_K \frac{{}^j\mathbf{c}_i \times \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}}{{}^jl_i^2}\, {}^j\mathrm{d}a_i, \tag{4}$$

where I_K is the virtual current factor, ${}^j\mathbf{c}_i$ is the current direction vector of surface element i, ${}^j\mathrm{d}a_i$ is the surface element area, ${}^jl_i = \|\mathbf{x} - \mathbf{x}_{{}^j\mathbf{n}_i}\|$ is the distance of the current position of the point mass $\mathbf{x}$, and $\mathbf{x}_{{}^j\mathbf{n}_i}$ is the position of the obstacle surface element i of obstacle j.

2.3 Adapting the CF Surface Current Rotation Vector

In the original CF approach, Singh et al. [1996], two schemes for current definition where given. They generate independent current elements for every surface and therefore, induce especially for the 3D case unwanted oscillatory behavior. Due to various problems in applying these algorithms already in simulation, we chose an alternative approach to define the current directions. In particular, we select the current of all CF obstacle surfaces to depend on each other. The chosen definition uses the vector from the actual AO position $\mathbf{x}$ to the desired position $\mathbf{x}_d$, the goal vector $\mathbf{b}$ and the center of mass of the respective OO ${}^j\mathbf{m}_{com}$, see Fig. 2. The local circular field ${}^j\mathbf{B}_i$ of each surface element, acting on the virtual particle k is now defined as

$${}^j\mathbf{B}_i := I_K \frac{({}^j\mathbf{n}_i \times {}^j\mathbf{r}) \times \frac{\dot{\mathbf{x}}}{\|\dot{\mathbf{x}}\|}}{{}^jl_i^2}\, {}^j\mathrm{d}a_i, \tag{5}$$

[1] Please note that we do not compare to these in this paper, as this is clearly out o the scope.

[2] Please note that for real experiments we utilize also an attractor that was derived in Haddadin et al. [2010].

[3] This atificial electro-magnetic-field equation is equivalent to eq. (2) in Singh et al. [1996]. For comparison, to the real world, we may e.g. refer to the law of Biot-Savart of the straight conductor, where the magnetic field is obtained as $d\mathbf{B} = \mu_r\mu_0 \frac{I\ d\mathbf{l} \times \mathbf{r}}{4\pi\|\mathbf{r}\|^3}$. $d\mathbf{l}$ represents the length of the conductor part (in our case the size of the surface and direction of current) and $\mathbf{r}$ is the distance vector.

where ${}^j\mathbf{n}_i$ is the normal of surface element i and $\mathbf{x}_{{}^j\mathbf{n}_i}$ is the position of the surface element. The surface current of (4) is defined as ${}^j\mathbf{c}_i :={}^j \mathbf{n}_i \times{}^j \mathbf{r}$.

${}^j\mathbf{r}$ is the field rotation vector of obstacle j, which is defined as

$$
{}^j\mathbf{r} := \frac{{}^j\mathbf{d} \times \mathbf{b}}{||{}^j\mathbf{d} \times \mathbf{b}||}, \tag{6}
$$

with ${}^j\mathbf{d}$ being the shortest distance between the center of mass (com) of j and the goal vector $\mathbf{b}$:

$$
{}^j\mathbf{d} = \mathbf{x} + \mathbf{b}\frac{({}^j\mathbf{m}_{com} - \mathbf{x}) \cdot \mathbf{b}}{||\mathbf{b}||^2} - {}^j\mathbf{m}_{com} \tag{7}
$$

With the given definition the current is continuous around the object and therefore oscillating robot behavior is avoided, as long as the damping and attractor are chosen accordingly.

2.4 Velocity Angle Adaptation

The chosen current definition leads to diverting forces in front of the obstacle and forwarding forces behind it. Even though this behavior will already lead to satisfactory results, it may lead to a "caching" type effect, which may cause the robot to be follow the obstacle surface for some cycles (however no global minimum present). Changing the direction according to the obstacle rotation vector ${}^j\mathbf{r}$ about a constant angle ψ

$$
\dot{\mathbf{x}}^* = \mathbf{R}\left(\frac{{}^j\mathbf{r}}{||{}^j\mathbf{r}||}\psi\right)\dot{\mathbf{x}} \tag{8}
$$

reduces the aforementioned effect significantly, since

- earlier deviation (also if being parallel to an infinite wall) and
- reduction of the CF influence after passing by the obstacle (when not being that deep in the CF anymore) are the result.

2.5 Calculation Load Reduction

The construction of obstacles and AOs based on surfaces results in an exponentially increasing calculation effort for the CF approach. Thus, it is of large interest to simplify the algorithm without loosing performance and to analyze possible parallellizability. For AO n and OO j consisting of surfaces k and i the virtual force equation includes two sum formulations.

$$
{}^{n,j}\mathbf{F}_v = \sum_k\sum_i({}^{n,j}\dot{\mathbf{x}}_{k,i} \times{}^{n,j} \mathbf{B}_{k,i}) \tag{9}
$$

In simulation experiments it turned out to be disadvantageous to include the relative velocity ${}^n\dot{\mathbf{x}}_k$ of surfaces k to the body center point of the object n. Thus, we consider the relative velocity to the body center point approximately to be zero and therefore negligible. The velocity of every avoiding surface element is now the relative velocity ${}^{n,j}\dot{\mathbf{x}}$ of AOs n to OOs j. Because of this simplification it is possible to write (9) as

$$
{}^{n,j}\mathbf{F}_v ={}^{n,j} \dot{\mathbf{x}} \times \sum_k\sum_i {}^{n,j}\mathbf{B}_{k,i}, \tag{10}
$$

In the following, we define $\dot{\mathbf{x}} :={}^{n,j} \dot{\mathbf{x}}$. ${}^{n,j}\mathbf{B}_{k,i}$ may now be written as

$$
{}^{n,j}\mathbf{B}_{k,i} := I_K\frac{{}^j\mathbf{c}_i \times \frac{\dot{\mathbf{x}}}{||\dot{\mathbf{x}}||}}{{}^{n,j}l^2_{k,i}}\,{}^j\mathrm{d}a_i\,{}^n\mathrm{d}a_k, \tag{11}
$$

The current ${}^j\mathbf{c}_i$ is either calculated ahead or alternatively, could be chosen to be constant. If we also regard that the objects are constructed with approximately equally sized surfaces, the calculation effort reduces further and (11) is simplified to

$$
{}^{n,j}\mathbf{B}_{k,i} := I_K\frac{{}^j\mathbf{c}_i \times \frac{\dot{\mathbf{x}}}{||\dot{\mathbf{x}}||}}{{}^{n,j}l^2_{k,i}}\,\mathrm{d}a^2. \tag{12}
$$

Further, reduction is achieved by combining obstacle surface size factor and the current factor to k_B. With this it is possible to change the sum formulation as follows.

$$
\sum_k\sum_i {}^{n,j}\mathbf{B}_{k,i} = I_K\,\mathrm{d}a^2\sum_k\sum_i\frac{{}^j\mathbf{c}_i}{{}^{n,j}l^2_{k,i}} \times \frac{\dot{\mathbf{x}}}{||\dot{\mathbf{x}}||} \tag{13}
$$

$$
= -k_{B_i}\frac{\dot{\mathbf{x}}}{||\dot{\mathbf{x}}||} \times \sum_k\sum_i\frac{{}^j\mathbf{c}_i}{{}^{n,j}l^2_{k,i}} \quad \text{with} \quad k_{B_i} := I_K\,\mathrm{d}a^2 \tag{14}
$$

Thus, the object force may be written as

$$
{}^{n,j}\mathbf{F}_v = -k_{B_i}\left(\dot{\mathbf{x}} \times \left(\frac{\dot{\mathbf{x}}}{||\dot{\mathbf{x}}||} \times \sum_k\sum_i\frac{{}^j\mathbf{c}_i}{{}^{n,j}l^2_{k,i}}\right)\right). \tag{15}
$$

Finally, the calculation effort is significantly smaller and may be even further reduced with a voxel space approximation for ${}^{n,j}l^2_{k,i}$. In addition, the algorithm is now highly parallelizable. The parallelization of sub-parts of the avoiding robot is advantageous for multiple avoiding objects (e.g. several objects attached to the different robot links).

Additional Calculations for PFs Since later on for the 6D case we combine CFs and PFs (CFs for translation and PFs for orientation), we also calculate the corresponding PFs for each obstacle surface acting on every avoiding surface. Therefore, results from the CF force algorithm can be used for the generation of PF forces. If only the potential field forces need to be calculated for every avoiding surface,

$$
{}^{n,j}\mathbf{F}_k = k_r\sum_i\frac{{}^n\mathbf{x}_k - \mathbf{x}_{{}^j\mathbf{n}_i}}{||{}^n\mathbf{x}_k - \mathbf{x}_{{}^j\mathbf{n}_i}||^2} \tag{16}
$$

has to be solved with k_r being a positive constant [4]. This means that the minimal calculation effort $\forall i$ is 8 additions, 3 multiplications, and 3 divisions. This leads to 14 floating point operations (FLOs) in total. Using the fact that ${}^{n,j}\mathbf{dx}_{k,i} ={}^n \mathbf{x}_k - \mathbf{x}_{{}^j\mathbf{n}_i}$ and ${}^{n,j}l^2_{k,i} = \|{}^n\mathbf{x}_k - \mathbf{x}_{\mathbf{n}_i}\|^2$ were already calculated for the CF forces, the remaining calculations are

$$
{}^{n,j}\mathbf{F}_k = k_r\sum_i\frac{{}^{n,j}\mathbf{dx}_{k,i}}{{}^{n,j}l^2_{k,i}}. \tag{17}
$$

In other words, 14 FLOs are reduced to 6 FLOs, or 3 additions and 3 divisions, so the reduction of calculation load is 43 %.

Next, we dicuss various simulation and experimental results obtained with the described method.

3. COLLISION AVOIDANCE: SIMULATIONS & EXPERIMENTS

In this section the experimental performance of the algorithm is shown for various 2D, 3D, and 6D scenarios. In all scenarios we assume all object center of masses to be known, while the instantaneous range of view is limited.

[4] This is of course one possible choice for Potential Fields, however, a very advantageous one by means of calculation load reduction.

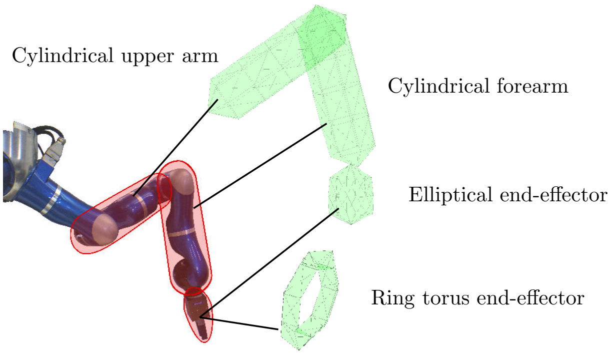


Fig. 3. Implemented geometrical robot hull for the LWR-III. Two different end-effectors are used in our experiments (elliptical and ring torus).

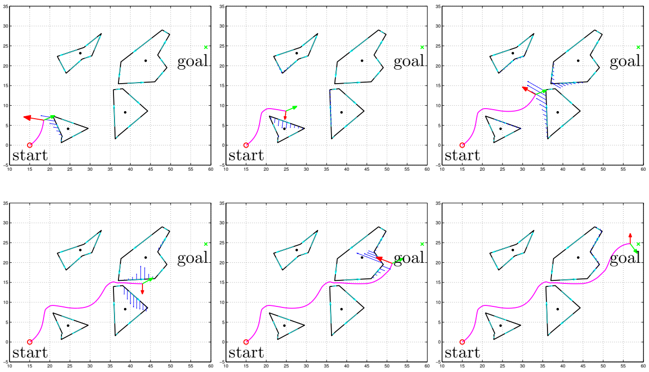


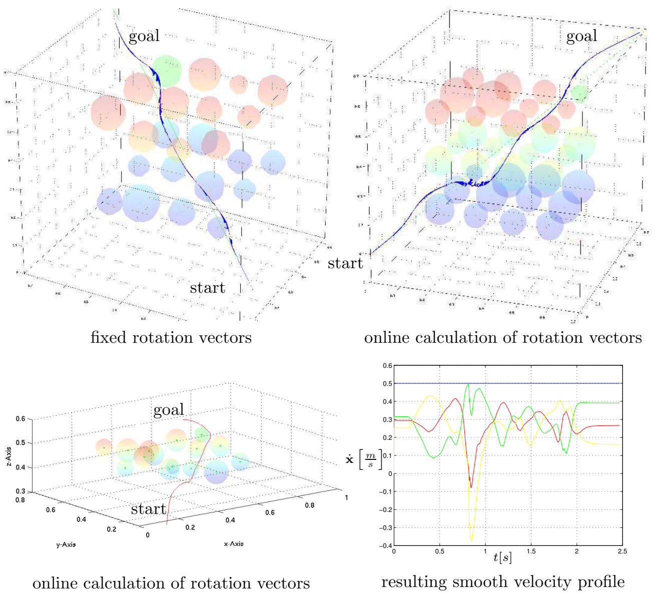
Fig. 4. 2D example our Circular Field method. The robot surpasses a narrow passage in real-time.

3.1 Robot Hull Design

In the 2D and 3D cases we consider the robot to be a point mass, while for the 6D case a volumetric representation is assigned to the moving robot parts. The robot is formally separated into several avoiding objects (AOs), see Fig. 3. Thereby, we can generate virtual forces and moments on each separate robot segment [5].

3.2 Simulation Random 2D

In Figure 4 a 2D example of a narrow passage problem is shown. The polygons are randomly generated [6] and six sample steps from the full simulation were chosen to indicate the performance of the method. The virtual particle has a limited view range, which is indicated by the virtual forces that are attached to the discretized surface elements. The only global knowledge is the geometric coms of the obstacles. The resulting external force (red arrow) acting on the virtual particle and the attractor force (green) are shown. Furthermore, the associated current direction is indicated on the object borders. In contrast to potential fields, which always generate repulsive forces in normal direction to the surface element (and are therefore prone to get stuck in local minima), this approach generates a very intuitive rather deviating force response for the presented problem, smoothly guiding the robot to the goal.



fixed rotation vectors

online calculation of rotation vectors

online calculation of rotation vectors

resulting smooth velocity profile

Fig. 5. 3D avoidance of spherical objects with a point mass.

3.3 Static random 3D spheres

For the 3D case we again chose to evaluate random testbeds, see Fig. 5. The obstacle influence range is between 0.03 and 0.05 units (again, the coms are known). The *upper left* figure depicts the behavior for fixed rotation vectors ${}^{j}\mathbf{r}$. The virtual point mass takes one of the shortest trajectories towards the goal. The images *top right* and *down left* are simulated for calculating the rotation vectors online. The derivation of ${}^{j}\mathbf{r}$ leads to a virtual point mass trajectory with less risk of colliding with obstacles while moving towards the goal. The trajectory is similarly smooth as the previous example and needs approximately the same number of iteration steps for goal convergence, i.e. the generated trajectory is of similar quality as for fixed ${}^{j}\mathbf{r}$. Finally, in the image *down right* an example for a resulting velocity profile is given, showing the smooth behavior during a motion (which is also no granted for many existing algorithms).

3.4 3D Trap Simulation Scenarios

To further analyze the capabilities of this algorithm in 3D, let us consider more complex problems, c.f. Fig. 6 and Fig. 7. Figure 6 shows a 3D trap from different angles. The robot has a limited view and thus enters the trap. However, after it is able to sense the walls of the object, it escapes and converges to the goal. Figure 7 depicts a box with only a single small entry. Even though the robot has only very limited view (indicated by the sparsely visualized grey spheres), it is able to escape and converge very smoothly and converge to the goal. As one can see from the presented simulations, the collision avoidance based on Circular Fields is able to cope with complex obstacles that obviously have local minima and non trivial geometries.

3.5 Dynamic Objects

The behavior in a highly dynamic environment is also simulated in order to judge and adjust the behavior for this especially important situation. For this the algorithm is analyzed for some special cases of obstacle motion, where the velocity of the obstacles is approximately the same

[5] The current hull implementation is only used for evaluation purposes and will be substituted by automatically generated hulls, Frese and Täubig [2009].

[6] We tested 50 random parcours, which were all successfully solved. Several examples can be seen online at http://www.safe-robots.com/ifac2011-cfca.html.

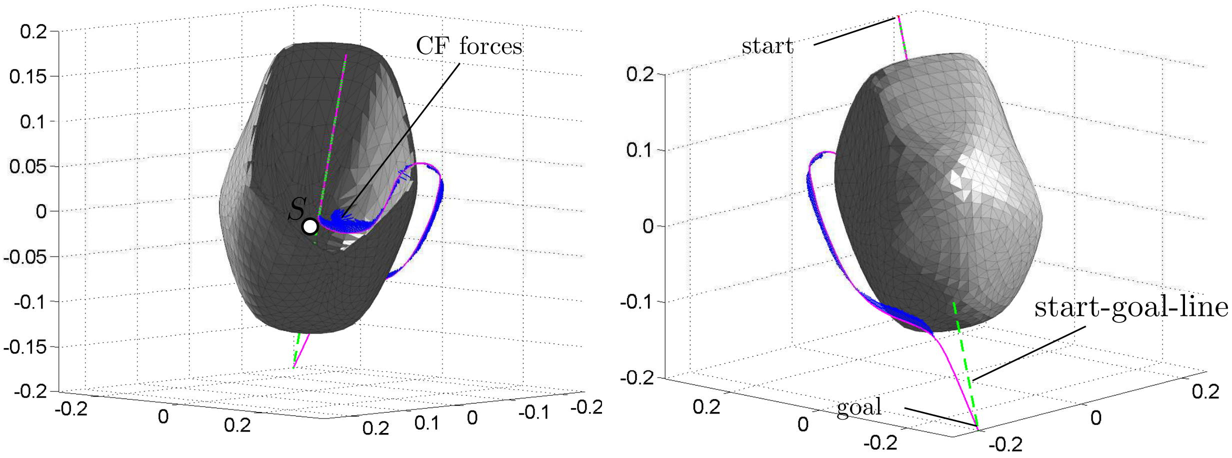


Fig. 6. Path of the robot in a 3D environment, which consists of a local minimum. The range of view is chosen to be very small and the field rotation vector is kept cons without getting stuck in the apparent local minimum.

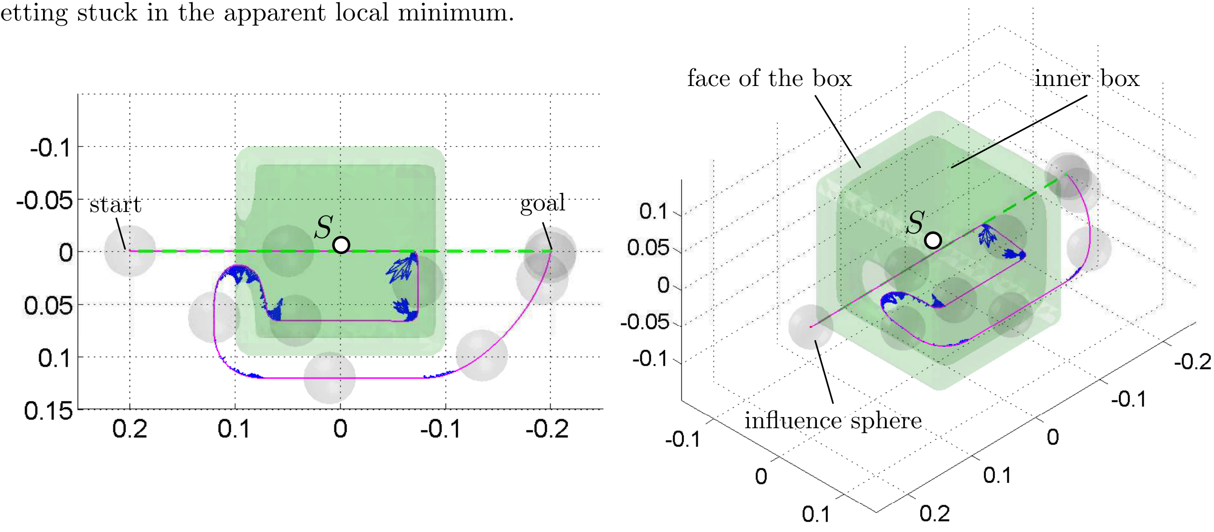


Fig. 7. Using CFs to reactively pass a complex dead-end (top and 3D View).

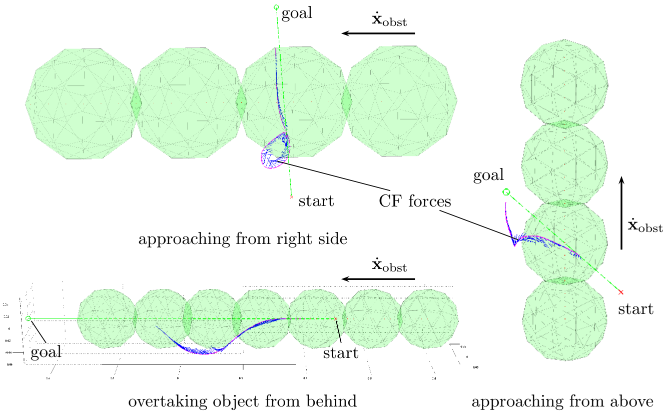

goal
$\dot{\mathbf{x}}_{\text{obst}}$
start
CF forces
approaching from right side
goal
$\dot{\mathbf{x}}_{\text{obst}}$
start
$\dot{\mathbf{x}}_{\text{obst}}$
goal
start
overtaking object from behind
approaching from above

Fig. 8. Point mass avoiding dynamically moving objects.

or larger than the robot velocity, see Fig. 8. In order to avoid the approaching obstacles we chose different strategies according to the current situation. The distinction is made based on whether the object approaches from behind or from the side/above. Avoiding a high velocity obstacle approaching from the front at high velocity can be performed with the approach already described. Therefore, we omit the discussion of this particular case: For the other cases we chose following strategies.

- Relative velocity based
- Overtaking: switch orientation of ${}^{j}\mathbf{r}$ by 180 ° if the relative velocity is a towards motion (AO) or towards the goal (different approaches possible)
- Object from side/below/above: continuous turning of ${}^{j}\mathbf{r}$ depending on the (position to) OO and the relative velocity.

With these strategies reasonable performance can be achieved. An open problem we still encounter is how to consistently combine the used strategies or to find **one** continuous strategy. However, this is left for future work.

Next, we discuss the experimental results and show that even some global motion planning problems can be tackled in real-time.

3.6 6D End-Effector Collision Avoidance Experiment

Figure 9 depicts a classical operational space control loop consisting of

(1) the motion generator for providing the desired reference motion in terms of generalized operational space coordinates $\mathbf{x}_d$ as a function of time,

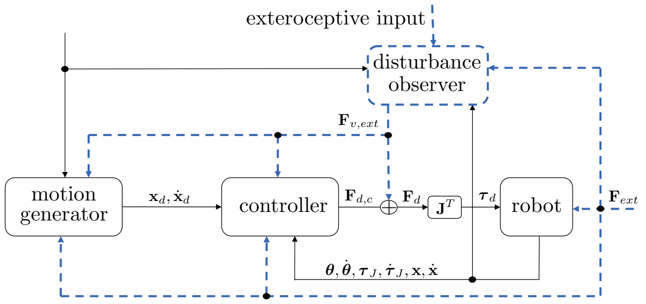


Fig. 9. Using disturbance inputs at different motion control levels for realizing effective disturbance response.

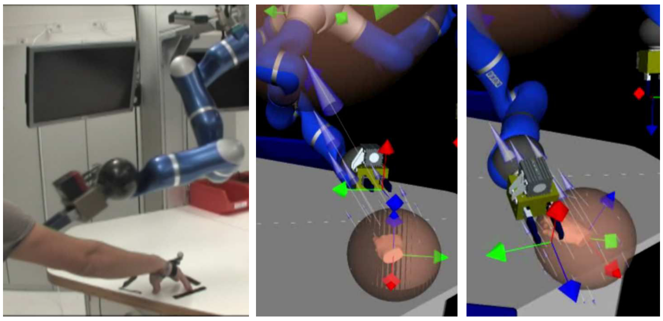


Fig. 10. Avoidance behavior for a Cartesian point to point motion. The depicted arrows denote the occurring forces acting on the ellipsoidal end-effector.

(2) the controller that provides the desired operational space force, which is then transformed via the Jacobian transpose to desired torque commands, and
(3) the physical robot that transforms the desired torque command via a low-level motor torque loop into motor torques that generate the respective robot motion.

A reactive disturbance response[7] can be fourfold:

(1) Physical forces act on the robot and inherently produce a dynamic response of the robot.
(2) The controller can implement a purely passive disturbance response (the measurement of external forces is not directly incorporated) with respect to external forces, or actively react to them (a classical example is inertia shaping). A combination of both is of course possible as well.
(3) Furthermore, generalized virtual forces, e.g. generated by repulsive potentials, can directly act as a motor command input and add a respective behavior to the controller.
(4) The motion generator provides motion commands that directly take into account the presence of physical or generalized virtual forces, leading e.g. to a collision retraction or reactive collision maneuvers.

For the following experiments we use all four of them (impedance control, sensing of external forces in Sec. 4, generation of virtual wrenches acting as on control input, and reactive attractor dynamics for motion reference generation). In the first experiment we evaluate the 6D case for a robot that avoids a dynamically moving human. The resulting forces from the obstacles are depicted for such a run in Fig. 10. As already mentioned we use CFs for translation and PFs for rotation as this generates a quite intuitive behavior of the robot in reality. On the left side the real posture of the OO (passive marker on hand) and the final configuration of the robot arm are illustrated. In the middle the 3D visualization before passing and on the right the one while passing the dynamic obstacle is shown. Additionally, the environment frames, their respective changes, and the progression of the obstacle forces are illustrated. The reorientation of the end-effector, which also has large impact on the motion of the forearm and on the upper arm of the robot can be observed. A time evolution of the process (end-effector hull and obstacle) is depicted in Fig. 11. Please note that in general, if the obstacle approaches the end-effector too closely, PF forces generate a translational movement that prevents a possible collision.

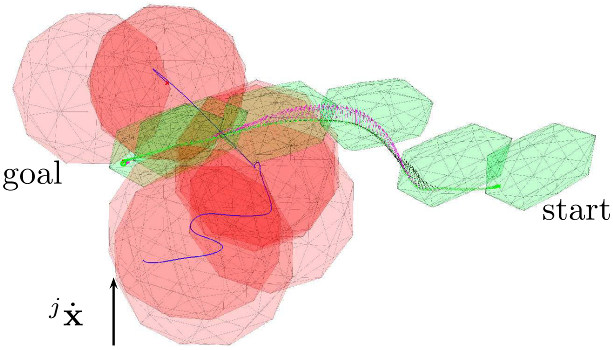


Fig. 11. Visualization of the human behavior (red spheres) and the robot end-effector hull (green object), which smoothly circumvents the dynamically moving human.

Figure 12 shows the behavior of our algorithm for a global motion planning problem with several local minima. However, the robot is able to reach the goal frames, while calculating the avoidance motion in real-time (0.5 kHz). In the video one can also see the direct comparison to the pure PF approach, which easily gets stuck in local minima and when activating the proposed algorithm the robot finds it way again in real-time.

Next, we outline how to use our method such that tactile exploration of an unknown object becomes possible.

4. TACTILE EXPLORATION: SIMULATIONS & EXPERIMENTS

In this section we describe how we use the Hybrid CF-PF approach together with a finite reactive planner to successfully solve a complex tactile exploration task, the "Hot Wire" problem. For this we use the information about interaction forces measured from the LWR-III[8] to generate a virtual obstacle map. Then, this map is interpreted as an obstacle the robot wants to follow without colliding with it. The overall scheme enables us to explore complex 6D objects and, after the exploration phase is over, to perform motions that do not collide with the obstacles anymore. Figure 13 gives a perspective view on the problem and showcases multiple scenarios that were experimentally verified.

The "Hot Wire" problem is intended to show that it is possible to explore a complex geometric structure in real-time

[7] Please note that we do refer to real-time reaction and not adaptation of via points or motion fragments that are e.g. provided by a global motion planner.

[8] We use a nonlinear disturbance observer based on the generalized momentum of the manipulator together with the integrated joint torque sensors to obtain a good estimation of the external torques acting on the robot Haddadin et al. [2008]. These are then transformed into Cartesian wrenches via the Jacobian transpose pseudoinverse.

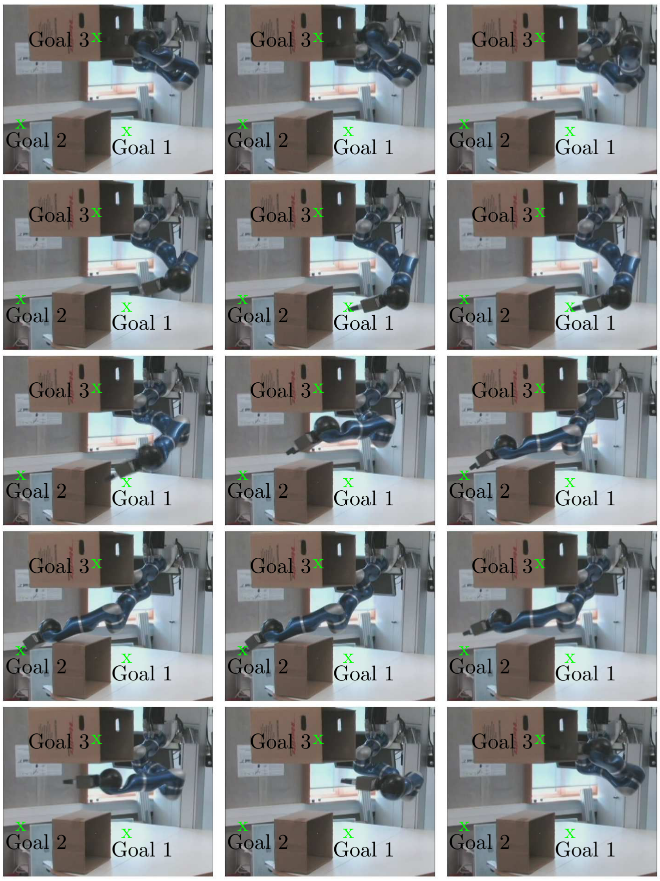


Fig. 12. The task is to reach all three goals in real-time. The robot is able to perform this task without getting stuck in a local minimum.

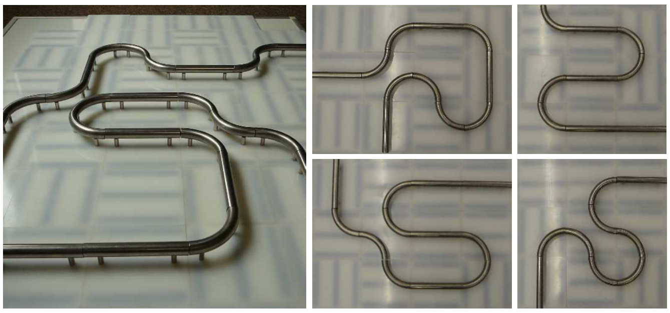


Fig. 13. The hot wire assembly.

and by tactile contact information only[9]. Furthermore, we use contact information to generate a tactile map of the object that subsequently can then be used for future motion generation.

4.1 Simulation

Approach The task to explore an object by interoceptive perception capabilities of a robot only is still a major challenge in robotics. The combination of tactile exploration and local motion generation algorithms is a novel way to approach the problem. The underlying idea is to associate the robot with a virtual 6D end-effector object, which is guiding the motion in a virtual environment. Based on interaction forces/torques sensed during motion this environment is incrementally built. In our implementation we use a torus as the virtual representation of our gripper to explore a wire labyrinth that is not known a-priori, see Fig. 13. This prevents the robot from leaving the scene via the obvious free space above the parcour. We generate translational and angular velocities to control the robot end-effector directly in Cartesian impedance control based on the sensed wrenches and avoidance map iteration.

The basic approach is to explore the physical object (wire) by using force and position measurements of the internal sensors of the robot and to construct an "avoidance map" based on a simple assumption on the unknown object geometry: Initially, the wire is believed to be an infinite straight object with certain radius. Furthermore, we expect the robot initial pose to be aligned with the beginning of the wire-type object to be explored. Based on the sensed tactile information as e.g. force, moment, position, and orientation we then continuously re-orientate this wire element during explorational pase.

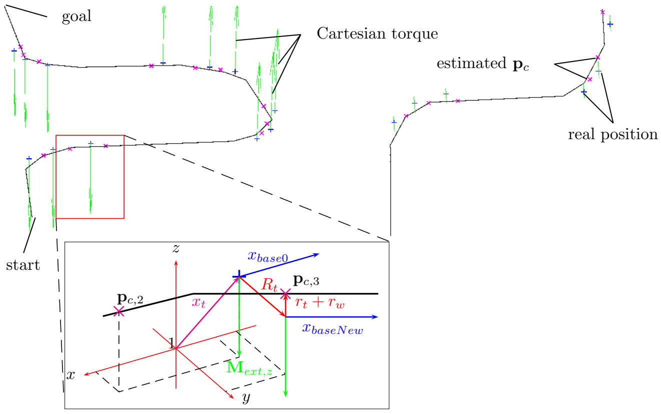


Fig. 14. Hot wire generation sketch.

First, the a-priori assumption on the orientation $\mathbf{x}_{\text{base0}}$ of the wire element is used to rotate the basic wire element object into the respective start pose. Then, two approaches can be used for calculating the rotation vector based on force and position sensor input. First, we may assume that sensed forces are directly aligned with the geometric normal of the object. This leads to aligning new elements orthogonally to this vector. Secondly, the rotation can be performed according to the measured moments (Fig. 14 green arrows). According to some initial testing, the torque based approach appeared to be the more robust due to non negligible friction effects during contact. The update rule for the new orientation is

$$\mathbf{x}_{\text{baseNew}} = \mathbf{R}(f(M_{\text{ext},z}))\mathbf{x}_{\text{base0}}, \tag{18}$$

where f is a suitable function to limit the calculated rotation for high torques $M_{\text{ext},z}$ (the z-axis is the relevant rotation for the problem. $\mathbf{x}_{\text{baseNew}}$ is the reorientated wire direction. The re-orientation is then performed at the estimated wire center point $\mathbf{p}_c$ of the contact (Fig. 14 magenta marks). This is obtained by the actual position of the torus $\mathbf{x}_{\text{t}}$ (blue marks) and the external torque $M_{\text{ext},z}$.

$$\mathbf{p}_c = R_{\text{t}} \frac{M_{\text{ext},z} \times \mathbf{x}_{\text{base0}}}{\|M_{\text{ext},z} \times \mathbf{x}_{\text{base0}}\|} + \mathbf{x}_{\text{t}} - (r_{\text{t}} + r_{\text{w}}) \frac{M_{\text{ext},z} \times \mathbf{x}_{\text{baseNew}}}{\|M_{\text{ext},z} \times \mathbf{x}_{\text{baseNew}}\|} \tag{19}$$

Hereby, r_{w} is the radius of the wire, R_{t} is the outer and r_{t} the inner radius of the estimated ring torus end-effector. In the following, $\mathbf{p}_c$ and the new orientation $\mathbf{x}_{\text{baseNew}}$ are used to calculate the crossing point with the already existing wire element for redefining the internal model of the wire (this is a polygon of line elements, see Fig. 14, black line). The motion generation approach for the problem uses only CFs (no damper and attractor), thus a free floating mass for translation and PFs with damping and no attractor for rotation (rotation energy is always decreasing: $D \propto E_{\text{rot}}$).

[9] With visual feedback this would be a rather trivial problem.

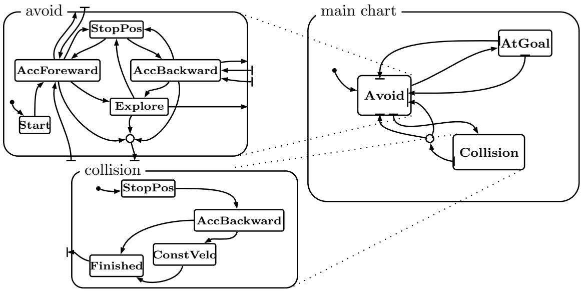


Fig. 15. "Hot Wire" hybrid planner.

There is no "long-term" rotational local minimum due to map building.

Reactive Planner To control the overall robot behavior during the task we designed a finite state reactive planner, which main chart is shown in Fig. 15. It consists of three main states:

(1) the "avoid" state,
(2) the "in collision" state, and
(3) the "reached goal" state.

In the last one the end-effector is stopped and waits for new commands. The "avoid" state consists of 5 sub-states that are self-explanatory: the "wait for start", "stop at this position", "acceleration forward", "acceleration backward", and the "explore" state. The latter tracks the current internal wire model and is left if external forces become too large (leads to a change into the sub-state "in collision"). "in collision" determines the course of action during a collision: The end-effector is stopped and the time evolution of contact forces is observed for creating a new internal model. Then, by calculating the crossing point of the new line element, the new destination for retraction is obtained. In order to avoid discontinuous virtual forces, the avoidance map is not updated while moving back. During the "acceleration backward" and "constant velocity" sub-states the robot then moves back. If the task is completed (finished), the motion is stopped. Next the avoidance map is updated and the state changes to "avoid". With the described behavior it is possible to fully explore the wire and afterwards navigate with the robot through the wire without causing any collisions anymore.

Results In order to judge the quality of the algorithm in simulation, we tested various randomly generated wires, of which two trials are depicted in Fig. 16. As already described the task is to explore an unknown generated wire (green) with a ring torus (green) by continuously deforming the initial iteration (red). The orientation of the ring torus is also marked with red and green marks. For both wire examples the resulting torques on the torus are calculated and this residual is integrated until a certain threshold is surpassed. If this incident occurs, the instantaneous difference of moments is associated with a contact moment and used to re-orientate the wire (red). As shown in Fig. 16 (right) the entire object can be explored such that its full geometry is reconstructed.

Next, we discuss the experimental implementation and results.

4.2 Experiment

The "Hot Wire" experiment is basically performed with the same approach as for simulation. The collision avoidance is used to command the reference translational and angular velocities open-loop for a Cartesian impedance controller, i.e. act as a trajectory generator. However, in case a contact between gripper and wire, the real state measurements are taken into account in order to generate the avoidance map as accurate as possible. The contact is coped with by the reactive planner with help from the local behavior of the impedance controller and the estimation of external forces. In Fig. 17 the result of such a Hot Wire run is depicted. As one can see the virtual wire element is re-oriented at every contact point according to the true wire. At the time instant contact is detected, the desired state is illustrated by a yellow torus. The real robot position is marked with a blue cross mark. The difference between real and desired state of the robot can be observed by the position difference. Magenta crosses mark the points of the estimated position of the wire center during contact. For the contact incidents also the contact torques (green arrows) are depicted (visible for the two left turns of the wire). This experiment was performed with multiple wire configurations, see Fig 13, with the wire structure being changed also online.

5. CONCLUSION

In this paper we presented and analyzed a real-time collision avoidance scheme and discussed its applicability to articulated manipulators in dynamically changing environments. It is suitable for complex environments as it provides good goal convergence properties, as well as the necessary features to work in applications of physical Human-Robot Interaction. We extended the original definition of CFs and finally developed a hybrid CF-PF approach for 6D reactive Operational space real-time motion. Static multi-object parcours and avoidance of dynamically moving humans can be achieved at the same time. Furthermore, the developed algorithm for tactile exploration of complex planar 3D wire elements, whose structure is a-priori unknown, is a novel approach to look at the problem.

Please find videos that show numerous results of the paper at http://www.safe-robots.com/ifac2011-cfca.html

ACKNOWLEDGEMENTS

We would like to thank Sven Parusel for his support during the experimental phase. Furthermore, we would like express our gratitude to Tim Rokahr and Andrea Schwier for building the Hot Wire parcour.

REFERENCES

A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger. The DLR lightweight robot lightweight design and soft robotics control concepts for robots in human environments. *Industrial Robot*, 34(5):376–385, 2007.

A. Albu-Schäffer, C. Ott, and G. Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *Int. J. Robotics Research*, 26(1):23–39, 2007.

O. Brock and O. Khatib. Elastic Strips: A Framework for Motion Generation in Human Environments. *Int. J. Robotics Research*, 21(12):1031–1052, 2002.

A. De Luca, A. Albu-Schäffer, S. Haddadin, and G. Hirzinger. Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1623–1630, 2006.

D. Ebert and D. Henrich. Safe human-robot-cooperation: Image-based collision detection for industrial robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1826–1831, 2002.

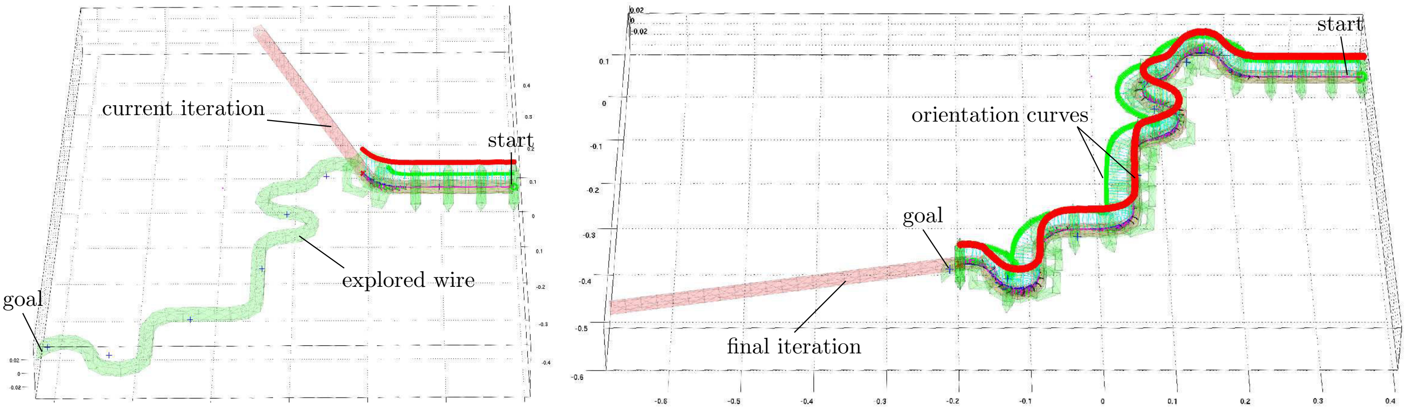


Fig. 16. Simulation of the hot wire exploration: started (left). Simulation hot wire exploration: finished (right).

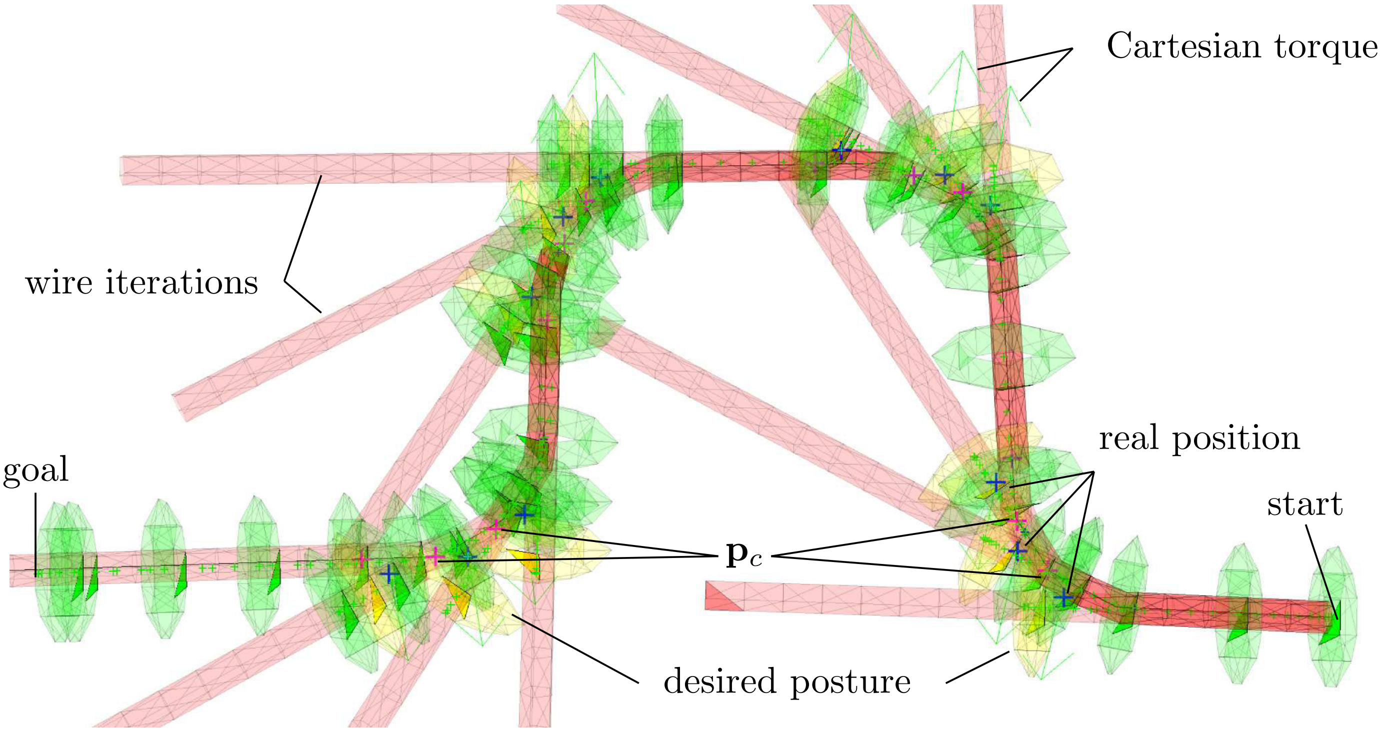


Fig. 17. Perspective view on the explored avoidance map and intermediate steps.

U. Frese and H. Täubig. Modelling and calibration technique of laser triangulation sensors for integration in robot arms and articulated arm coordinate measuring machines. *Research Report RR-09-01*, 2009.

S. Haddadin, A. Albu-Schäffer, A. De Luca, and G. Hirzinger. Collision detection & reaction: A contribution to safe physical human-robot interaction. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3356–3363, 2008.

S. Haddadin, H. Urbanek, S. Parusel, D. Burschka, J. Roßmann, A. Albu-Schäffer, and Gerd Hirzinger. Realtime reactive motion generation based on variable attractor dynamics and shaped velocities. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 4735–4742, 2010.

O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 500–505, 1985.

P. Ögren, N. Egerstedt, and X. Hu. Reactive mobile manipulation using dynamic trajectory tracking. In *IEEE Int. Conf. on Robotics and Automation*, pages 3473–3478, 2000.

B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer, 2008.

L. Singh, H. Stephanou, and J. Wen. Real-time robot motion control with circulatory fields. In *IEEE Int. Conf. on Robotics and Automation*, pages 2737–2742, 1996.

W. Townsend and J.A. Guertin. Teleoperator slave - WAM design methodology. *Int. J. Robotics Research*, 26(3): 167–177, 1999.

S. Wang, J. Bao, and Y. Fu. Real-time motion planning for robot manipulators in unknown environments using infrared sensors. *Robotica*, 25(2):201–211, 2007.

Y. Yamamoto and X. Yun. Coordinated obstacle avoidance of a mobile manipulator. In *IEEE Int. Conf. on Robotics and Automation*, pages 2255–2260, 1995.